Die approbierte Originalversion dieser Dissertation ist in der Hauptbibliothek der Technischen Universität Wien aufgestellt und zugänglich.
http://www.ub.tuwien.ac.at

The approved original version of this thesis is available at the main library of the Vienna University of Technology.
http://www.ub.tuwien.ac.at/eng

FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Inconsistency Management under Preferences for Multi-Context Systems and Extensions

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktor der technischen Wissenschaften

by

Antonius Weinzierl
Registration Number 0926018

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: O. Univ.-Prof. Dr. Thomas Eiter
Dr. Michael Fink

The dissertation has been reviewed by:

(O. Univ.-Prof. Dr. Thomas Eiter) (Prof. Dr. Gerhard Brewka)

Wien, 28.10.2014

(Antonius Weinzierl)

Erklärung zur Verfassung der Arbeit

Antonius Weinzierl
Schaumburgergasse 5, 1040 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum)	(Unterschrift Verfasser)

Danksagung

Ich bedanke mich bei Thomas Eiter für die gute Betreuung und die Freiheit, auch Fragen abseits der vorliegenden Dissertation verfolgen zu können. Michael Fink bin ich dankbar dafür, dass er mich nach Wien geholt hat in das vom Wiener Wissenschafts-, Forschungs- und Technologiefonds (WWTF) geförderte Forschungsprojekt *„Inconsistency Management for Knowledge-Integration Systems“ (ICT 08-020)*. Thomas und Michael haben mich sehr viel gelehrt, was das Schreiben wissenschaftlicher Publikationen betrifft. Ein großer Dank geht auch an Peter Schüller, einem Freund und Kollegen in obigem Forschungsprojekt. Die wertvollen, gemeinsamen Diskussionen über Multi-Context Systeme waren nicht nur sehr unterhaltsam, sondern schufen auch grundlegende Begriffe, auf die diese Arbeit aufbaut. Ich danke auch Gerd Brewka für die gute und spannende Zusammenarbeit bei der Erweiterung der Multi-Context Systeme, welche im letzten Teil dieser Arbeit behandelt wird.

Ich möchte mich auch bei den Kolleginnen und Kollegen sowie den Sekretärinnen am Institut für Informationssysteme bedanken, die mir mit ihrem Rat oft eine große Hilfe waren. Großer Dank gebührt auch Norbert Eisinger, der mit unglaublicher Präzision einen Entwurf dieser Arbeit gelesen und damit dutzende kleine und größere Fehler verhindert hat. Alle verbliebenen Fehler sind die meinigen.

Nicht zuletzt bedanke ich mich bei meinen Eltern Barbara und Anton, bei meiner Schwester Bernadett und bei meiner Freundin Elisabeth für all die Unterstützung, Hilfe, Rat und Liebe ohne die das hier nicht möglich gewesen wäre.

Abstract

Multi-Context Systems (MCS) are a versatile and powerful framework for heterogeneous, non-monotonic knowledge-integration. MCS allow information exchange between legacy information systems, i.e., knowledge bases. Inconsistency occurs easily in such scenarios since it is impossible to foresee all effects and consequences of the information exchange. Inconsistency makes an MCS useless, just as in other formal systems; thus, inconsistency management is a major issue.

Resolving inconsistency by purely technical means is guaranteed to yield a consistent system, but it can easily result in a system where the remaining information exchange leads to unwanted or even dangerous conclusions. Consider, for example, an MCS that is employed in a hospital and the billing subsystem became inconsistent because of a patient with insufficient insurance. Automatically resolving the inconsistency by declaring the patient to be healthy and sending her home instead of administering proper treatment surely is a solution, but it hardly is a valid one. On the other hand, manually resolving all inconsistencies is not feasible since there usually exist too many possible resolutions to consider each of them individually.

This thesis therefore addresses the issues of inconsistency management in MCS with a focus on using preferences to identify and automatically select preferred and valid resolutions of inconsistency, to aid a human operator by significantly reducing the number of possible resolutions to consider.

The novelty of this approach is on the one hand a technique to enable meta-reasoning about inconsistency resolutions within the MCS framework, i.e., preferences expressed in any knowledge formalism can be incorporated to identify preferred resolutions and filter unwanted ones. On the other hand, an extension of the MCS framework is introduced to enable the use of legacy inconsistency management methods directly at each information system.

This thesis consists of three main parts. The first investigates basic notions to identify and explain inconsistency in MCS. These notions are called diagnosis and explanation of inconsistency. Refined notions are investigated and shown to be reducible to the basic notions, and splitting-set based conditions are analyzed which allow to modularly obtain diagnoses and explanations from parts of a given MCS. Finally, a logic program is given that computes all explanations of an MCS.

The second part of this thesis is dedicated to the identification and selection of most-preferred diagnoses and the filtering of unwanted diagnoses. Several transformation-based approaches are introduced which allow a transformed MCS to reason about the diagnoses of the original MCS, i.e., these approaches enable meta-reasoning on diagnoses in MCS. The necessary extended notions of diagnosis are shown to be of the same complexity as the basic notion, except for one, which is of higher complexity but still shown to be worst-case optimal. Therefore, the new meta-reasoning approach incurs no unnecessary complexity-wise cost.

In the third part, the MCS framework is extended to incorporate existing, formalism-specific methods of inconsistency management (e.g., belief revision for classical logics, or updates for logic programs). In such an MCS where each knowledge base comes with local inconsistency management, the overall system can only become inconsistent due to cyclic information exchange. Furthermore, the extended framework is reducible to ordinary MCS and checking consistency in the extended framework is of the same computational complexity as in ordinary MCS.

Kurzfassung

Multi-Context Systeme (MCS) sind ein formales und ausdruckstarkes Rahmenwerk (engl. framework) um Wissen zwischen verschiedenen, nicht homogenen, Informationssystemen auszutauschen. Die Heterogenität von MCS ermöglicht den Informationsaustausch auch zwischen Altsystemen und damit die Nutzung von bestehenden Wissensbasen in neuen Zusammenhängen. Da es allerdings unmöglich ist, sämtliche Konsequenzen des Informationsaustausches im Voraus zu bestimmen, entsteht dabei sehr leicht sogenannte Inkonsistenz, d.h. der Informationsaustausch führt zu Widersprüchen. Ein formales System, das inkonsistent ist, bringt keinen Nutzen, da das System entweder keinerlei Schlussfolgerungen erlaubt, oder auch falsche Schlussfolgerungen als richtig deklariert. Das Behandeln und Beheben von Inkonsistenz ist daher ein wichtiges Thema für Wissenssysteme.

Theoretisch kann man Inkonsistenz als rein technisches Problem betrachten und jede mögliche Behebung der Inkonsistenz daher gleichberechtig auffassen. In der Praxis kann dies jedoch gefährliche Konsequenzen haben. Als Beispiel denke man an ein Krankenhaus, in welchem ein Teilsystem zur automatisierten Erstellung von Rechnungen, ausgelöst durch eine Patientin mit unzureichender Krankenversicherung, inkonsistent wird. Die Inkonsistenz ließe sich dadurch beheben, dass die Patientin für gesund erklärt wird und diese dann ohne die medizinisch notwendige Behandlung nach Hause geschickt wird. Die Inkonsistenz ist damit zwar behoben, aber sicher nicht auf eine zufriedenstellende Art und Weise. Eine manuelle Behebung der Inkonsistenz könnte diesen Fall vermeiden, ist aber mit dem Problem verbunden, dass es oftmals eine große Zahl an Möglichkeiten gibt, Inkonsistenz zu beseitigen. Oft ist es für einen Menschen daher schlicht unmöglich, jede einzelne davon gesondert zu betrachten.

Diese Arbeit untersucht daher die Probleme und Lösungsmöglichkeiten von Inkonsistenz (Konsistenzmanagement) in MCS, wobei ein besonderes Augenmerk auf die Verwendung von Präferenzen gerichtet wird, um die bestmögliche Art der Auflösung der Inkonsistenz zu identifizieren. Die automatisierte Auswahl solch präferierter und valider Lösungen unterstützt letztlich diejenige Person, welche ein MCS bedient und betreibt, da sie nur noch jene Lösungen der Inkonsistenz in Betracht ziehen muss, welche auch relevant sind.

Das Novum dieser Dissertation ist einerseits eine Technik für sogenanntes Meta-Reasoning, welche es erlaubt innerhalb des MCS Frameworks über präferierte Lösungen zu schließen, d.h. die Präferenzen müssen nicht in einem bestimmten Formalismus gegeben werden, sondern jeder Formalismus, der in einem MCS eingesetzt werden kann, kann auch benutzt werden um die beste Behebung der Inkonsistenz zu identifizieren. Andererseits wird im Folgenden auch eine Erweiterung des MCS Frameworks präsentiert, welche es erlaubt bestehende Methoden des

Konsistenzmanagements zu nutzen, um lokal die Konsistenz einzelner Informationssysteme zu garantieren.

Der Hauptteil der vorliegenden Arbeit ist in drei Teile gegliedert. Im ersten Teil werden grundlegende Konzepte untersucht, um Inkonsistenz in MCS zu beseitigen und um die Ursachen für Inkonsistenz zu identifizieren. Diese Konzepte werden „Diagnosis“ (Diagnose) und „Explanation“ (Erklärung) von Inkonsistenz genannt. Es werden mögliche Verfeinerungen untersucht, wobei sich zeigt, dass die Verfeinerungen auf die grundlegenden Konzepte zurückgeführt werden können. Weiters werden, basierend auf sogenannten Splitting-Sets, Eigenschaften identifiziert, die es erlauben, obige Konzepte eines gegebenen MCS dadurch zu erhalten, dass man die Konzepte nur auf Teilen des Systems berechnet und diese dann entsprechend kombiniert.

Der zweite Teil dieser Arbeit untersucht verschiedene Möglichkeiten, wie die am meisten präferierten Diagnosen eines inkonsistenten MCS identifiziert und selektiert werden können. Dabei werden mehrere Transformations-basierte Ansätze entwickelt, welche es erlauben innerhalb eines transformierten MCS über die Diagnosen des originalen MCS zu schließen, d.h. diese Transformationen ermöglichen Meta-Reasoning über Diagnosen innerhalb des MCS Frameworks. Es wird gezeigt, dass die dafür notwendigen erweiterten Begriffe von Diagnose mit einer Ausnahme die gleiche Berechenbarkeitskomplexität haben wie die grundlegende Diagnose. Die Ausnahme hat zwar höhere Komplexität, ist aber dennoch optimal im Bezug auf den schlechtesten Fall (engl. worst-case optimal). Daraus folgt, dass dieser neue Ansatz für Meta-Reasoning keine unnötigen Kosten im komplexitätstheoretischen Sinn verursacht.

Im dritten Teil wird das MCS Framework um lokales Konsistenzmanagement erweitert, damit bereits existierende Methoden des Konsistenzmanagements für bestehende Formalismen (z.B. Belief Revision für klassische Logiken oder Updates für Logikprogramme) lokal genutzt werden können, sobald ein Informationssystem eines MCS auf dem entsprechenden Formalismus basiert. Es zeigt sich, dass Inkonsistenz nur noch durch zyklischen Informationsfluss entstehen kann, wenn jedes Informationssystem eines MCS mit lokalem Konsistenzmanagement ausgestattet ist. Ferner ist das erweiterte Framework rückführbar auf das originale und es wird gezeigt, dass der Test, ob Inkonsistenz im erweiterten Framework vorliegt, die gleiche Berechenbarkeitskomplexität hat, wie im originalen Framework.

Contents

CHAPTER 1

Introduction

As human beings we are used to combine knowledge from various sources of information. When learning something new, we read a book or two, we discuss the matter with our friends and colleagues, and we maybe also read some magazine dedicated to the matter. Combining all the information gives us new insights and in most cases we profit from the multitude of information sources. Every day we gather information from different sources to make up our mind, e.g. information from the local newspapers, from books written in other languages, and from colleagues. For computers, however, this process of considering information from very different sources to gain new knowledge and insights, is still a difficult and largely unsolved problem.

One of the reasons for that is the fact that computers use rigorous and syntactically restricted languages (i.e., formal languages) and they cannot cope with expressions that are written in another formal language. Computerised sources of knowledge often “speak” different formal languages, since for many areas there are specifically-crafted formal languages that make it easy to express knowledge in this area, while expressing knowledge outside this area is often cumbersome or outright impossible. This focus on small areas of knowledge resulted in the development of a plethora of formal languages, each having convincing reasons to exist, and there does not seem to be a unifying language on the horizon to express all kinds of knowledge. Like Esperanto, a universal language intended to be spoken by all humankind, there have been attempts to establish such a language for computers. But they largely failed like Esperanto. Integrating knowledge from different sources therefore must be able to build bridges between formal languages, without trying to abolish any of them.

The problem is not new and especially since the rise of the world wide web there have been numerous approaches to enable computers to integrate knowledge from different sources. Most of these approaches, however, require that the knowledge sources provide the information in a formalism from a given homogeneous class, e.g. all sources must be databases of the same kind as in the area of database fusion and information integration; or all sources must be ontologies as in the area of modular ontologies; or all sources must be formal (usually propositional) logics as in the area of modal logics and many-world interpretations. A restriction which applies to many proposed solutions, is the exclusion of cyclic information exchange, i.e., it is not possible that two such knowledge systems mutually acquire knowledge from each other. Another restriction is the exclusion of nonmonotonic reasoning, although it is very natural to humans since it captures reasoning like "if the timetable at the train station does not show that a train departs at 13:15, then there is no train departing at that time".

One approach to addresses the problem of integrating knowledge from various sources of information is the framework of Multi-Context Systems (MCS). The MCS framework is distinguished in that it allows knowledge specified in different (heterogeneous) formalisms to be exchanged in cyclic and possibly nonmonotonic ways. In an MCS, an ontology can derive new information from the absence of information in a database, while the database can simultaneously use the knowledge of the ontology to derive new database entries.

The knowledge exchange between knowledge systems (called "contexts") is specified in MCS using so-called bridge rules. A bridge rule consists of a head and a body, where the head specifies what information is derived at which context, and the body specifies all conditions that must be met to actually derive the information in the head. For example, the following bridge rule states: if context 1 does not know about an allergy against strong antibiotics ($allergy\_strong\_ab$), then context 4 receives the information that strong antibiotics are allowed ($allow\_strong\_ab$).

$$(4: allow\_strong\_ab) \leftarrow \mathbf{not}\ (1: allergy\_strong\_ab).$$

The head is on the left side of the $\leftarrow$ while the body of this bridge rule is everything on the right. Observe that this bridge rule is nonmonotonic, since it derives new information ($allow\_strong\_ab$) from the absence of some other information ($allergy\_strong\_ab$).

Since the information in the head and all information referred to in the body of a bridge rule may originate from different formal languages, bridge rules are a simple yet powerful way to express information exchange between different knowledge sources. Note that bridge rules are assume to be crafted by a human designer who decides which information in one context is related to what information in another context. The MCS framework therefore does not solve the problem of finding corresponding pieces of knowledge in different formalisms, but it solves the problem of actually combining the knowledge once it is established how the knowledge should interact. Nonmonotonic bridge rules and heterogeneous contexts make MCS a powerful and versatile framework and it thus forms the basis of this thesis.

1.1 Motivation

Consider a hospital where a patient database, a medical ontology, and an expert system suggesting possible treatments are connected via bridge rules. Assume that the patient database states that the patient *Joe* has a certain type of pneumonia and a rare allergy to certain antibiotics. The medical ontology derives from the type of pneumonia that the only available medication is a certain antibiotic. Now the expert system combines these pieces of information and concludes that *Joe* necessarily needs the antibiotics he is allergic to. Because the only option is forbidden by the allergy and *Joe* requires treatment, the expert system becomes inconsistent. If this combination of rare events was not anticipated, the whole system becomes inconsistent and it is not able to conclude anything useful.

In this example, every context by itself is consistent, but their interaction leads to an inconsistency, so a local inconsistency management clearly is not enough. Modifying the content inside a context also is not always possible: consider for example two companies that agree to share some of their business information; if the resulting MCS is inconsistent, neither company is willing to drop its own business data only to satisfy some constraints raised by the other. We therefore consider inconsistency in terms of flawed information exchange, i.e., we identify bridge rules as reasons of inconsistency. The first important issue then is to identify those bridge rules that cause inconsistency, i.e., we want to *explain* the reasons of inconsistency and separate reasons if there are multiple inconsistencies in one MCS. The next question is how the inconsistency can be removed from the MCS, i.e., we want to *repair* the inconsistency. As it turns out, explanations and repairs (called *diagnoses*) of inconsistency in an MCS are in some sense dual concepts that identify the same set of bridge rules as culprits.

A diagnosis is a possible way to resolve the inconsistency. All diagnoses identify all ways to do so, but there usually exists a large number of diagnoses of the inconsistencies in an MCS. In the hospital example above, the inconsistency can be removed by modifying the information flow so that the patient is considered healthy (not requiring treatment), or the information about allergy is disregarded and the medication against pneumonia is given, which then possibly causes an allergic reaction. It is clear that such a case cannot be resolved automatically, since additional expert knowledge (i.e., an experienced medical doctor) is required. So, not all inconsistencies should be treated automatically, but the person(s) responsible for removal shall be supported as much as possible.

Assume, for example, that the cause of the inconsistency is not the presence of an allergy, but the fact that the automated billing system of the hospital does not allow a required medication, because the medication is not covered by the health insurance of the patient. Now there are two possibilities to remove the inconsistency: either ignore the illness of the patient, or ignore the issue with the health insurance. Treating the patient and ignoring the billing issue then surely is the preferred solution. To support those who are accountable for the removal of the inconsistency, the latter diagnosis could be singled out while all less preferred diagnoses are not shown. Selecting from the set of possible diagnoses those diagnoses that are the most preferred therefore is the second major issue regarding inconsistency management in MCS.

Inconsistency, on the other hand, has been studied for many knowledge formalisms and mechanisms to cope with and resolve inconsistency have been developed for these formalisms. For example, in the area of classical logic many methods to integrate new and possibly contradicting information have been investigated in the area of belief revision. There are many reasons for domain-specific solutions dealing with inconsistency. Therefore we do not want to ignore these solutions, but applying them globally to an MCS does not work either, since these methods are specifically tailored to certain logical formalisms while in an MCS many different formalisms may be used. The third big issue therefore is to devise ways such that existing solutions for inconsistency management may be applied to formalisms used in the contexts of MCS whenever these solutions fit.

1.2 Background and State-of-the-Art

The MCS framework originates from so-called MultiLanguage Systems (cf. [73, 74]), where hierarchically ordered logics interact via a specific kind of bridge rule resulting in a system that is similar to modal logics. In [114] Multi-Context Systems are introduced where every context is a propositional logic, but information exchange between them may be nonmonotonic. In [35] Multi-Context Systems are advanced by allowing contexts of an MCS to employ default theories, hence possibly nonmonotonic contexts exchange information via possibly nonmonotonic bridge rules. Finally, in [29], the MCS framework is extended to allow heterogeneous contexts, i.e., each context can use another logical formalism (including nonmonotonic ones) and bridge rules also may be nonmonotonic. This thesis is based on the Multi-Context Systems framework as introduced there.

At the beginning of this work, inconsistency in MCS only has been addressed in [16], later presented in [17, 19], where several algorithmic resolution strategies are proposed. The resolution of inconsistencies is local, trust-, and possibly provenance-based. Due to its local algorithmic description, the overall result is not easily described in formal terms. Furthermore, every context is required to use a strict total-order on contexts (representing trust) to enable the resolution of all inconsistencies. If the trust order is partial, not all inconsistencies can be resolved.

To globally identify the reasons of inconsistency, we use Reiter's diagnosis approach in [113] as a starting point. The most important difference to MCS is that Reiter's approach is given for monotonic logics, while MCS are nonmonotonic reasoning systems. A result of this difference is that our diagnosis notion is a pair of sets (of bridge rules), while Reiter's diagnosis is simply a set (of formulas). Confidence in this choice is strengthened by the fact that other approaches at inconsistency explanations for nonmonotonic formalisms also introduced pairs of sets (cf. [84]).

A large part of this thesis is dedicated to the selection of most preferred diagnoses among all possible diagnoses. While preference on diagnoses is easy to capture by the mathematical notion of a preference order (i.e., a transitive relation), finding a formalism to specify preferences in a natural way (i.e., a formalism for preference elicitation) seems to be much harder. The approach taken by CP-nets [25, 46] for specifying preferences is promising, hence we use it together with the more general notion of preference orders.

Regarding the treatment of inconsistency locally within one formalism, we find many approaches based on belief revision in classical logic (cf. the survey [105]) whose postulates of rational belief revision (AGM postulates) have been applied also to other formalisms. Another interesting application of inconsistency removal is the area of updates in logic programming [2, 26], since bridge rules of MCS are similar to rules in nonmonotonic logic programming.

More details on related work are given in Chapter 6.

1.3 Contributions and Methodology

This thesis addresses the three main issues of inconsistency management in Multi-Context Systems described above. Before that, however, it must be clarified what constitutes inconsistency with respect to MCS. There are two basic possibilities: an MCS is considered inconsistent, if there are two contexts whose beliefs contradict each other, or an MCS is inconsistent if the semantics yields no useful result. The former is closer to the classical understanding of inconsistency, but it does not seem to be suitable for MCS. Consider for example an MCS where one context believes $is\_raining$ and another context believes the opposite, i.e., $\neg is\_raining$. If the two contexts represent weather stations at different locations, then these beliefs are perfectly valid. Here the same symbol denotes two different real world entities (the weather at two different locations). In general, the meaning or semantics of a statement may vary from one context to another. Furthermore, an important application of MCS is the realisation of multiple communicating agents where the possibility to model conflicts of opinion is paramount. In this case, it does not even matter whether the weather stations are at different locations, since these express beliefs at different contexts/agents and opposite views are bearable. The latter notion of inconsistency therefore is more suitable for MCS, hence an MCS is considered to be inconsistent, if the semantics of MCS yields no result (a formal description follows in Chapter 2).

Our contributions to the main issues of inconsistency management in MCS are as follows.

- We develop basic notions to (resolve) and explain inconsistency in MCS: the diagnosis notion allows to identify all possible ways to repair an MCS by removing bridge rules or making them condition-free (such that they always add information); the notion of explanation identifies the reasons of an inconsistency and is able to separate multiple sources of inconsistency. We further investigate refined notions that apply more fine-grained modifications to bridge rules. We prove that these refinements can be obtained from the regular notions on a transformed version of the original MCS. Diagnoses or explanations of an MCS that are subset-minimal with respect to all diagnoses or explanations, respectively, point out bridge rules that are relevant for inconsistency.

  We further prove that under certain conditions, the (subset-minimal) diagnoses or explanations of an MCS may be obtained by combining the (subset-minimal) diagnoses or explanations, respectively, of parts of the MCS. This may aid in the computation of these notions since it allows for a significant reduction in search space. Some of these results are also used as stepping stones to prove the correctness of our approaches to preference. We also investigate an approach to decompose a context of an MCS; again, this may result in a smaller search space, but it also allows to de-centralise a given MCS.

  Finally, to complement the computation of diagnoses given in [117], we develop a program that computes all explanations of an MCS.

- To select the best diagnoses of an MCS, we propose two basic methods, one allows to filter out those diagnoses that fail some required properties, the other is to compare diagnoses with each other and select the most preferred one. For the former, we use filters, for the latter we use preference orders. Both are general concepts that can capture many concrete instances to express unwanted or preferred diagnoses. Since the MCS framework itself is flexible and open to integrate information from many different logical formalisms, we do not restrict the formalism in which filters and preferences are specified, i.e., the formalism is open to the user's choice.

  To realise the selection of diagnoses in such an open way, we develop several transformations to enable meta-reasoning about diagnoses in MCS, i.e., given an MCS and a filter or preference order, a transformed MCS is constructed such that the diagnoses of the original MCS also occur in the transformed MCS, but an additional context is able to observe these diagnoses and apply some custom reasoning. Since the observer context is not restricted to any particular formalism, this allows that filters and preferences are expressed in any formalism that can be employed as a context of an MCS. Using CP-nets (cf. [25]) to specify preference among diagnoses is also possible using the general approach for preference. The transformations are shown to allow correct selection of diagnoses that pass the filter or that are most preferred, respectively.

In the course of this, two extensions of the notion of diagnosis are introduced. The computational complexity of these notions is analysed and by the use of reductions, the first notion is shown to be of the same complexity as checking whether a pair of sets of bridge rules constitutes a diagnosis that is subset-minimal among all possible diagnoses. The second notion is shown to be of higher complexity using a genuine algorithm. Nevertheless the notion is still worst-case optimal, because it is shown that the basic problem of selecting most-preferred diagnoses also has presumably higher complexity.

- The notion of a diagnosis in MCS helps to restore global consistency, yet it may be the case that much better and more fine-grained resolutions of inconsistency are possible at the local level of each employed context. Our contribution to that is a significant generalisation of the MCS framework, which allows that each context is equipped with a method to modify its knowledge base given the information from other contexts. Our approach, which we call *managed Multi-Context Systems* (mMCS) allows the application of arbitrary operations on knowledge bases, but most importantly, it also allows the use of local, consistency-restoring methods (e.g., belief revision). Furthermore, it allows that each context is enhanced by those consistency-restoring methods that fit best to the formalism employed by the context, e.g., a belief revision operator for classical logics or an update mechanism for logic programs.

  We prove for any mMCS where every context is equipped with a manager guaranteeing local consistency of the context, that the mMCS is consistent if it contains no cyclic information flow. Furthermore, for an mMCS with such context managers and cyclic information flow, it holds that every inconsistency explanation includes a cycle. In other words, in an mMCS where each context manager guarantees consistency of its local context, the source of inconsistency always is some cyclic information exchange.

  Finally, we show by a reduction to the ordinary MCS framework, that deciding whether an mMCS is consistent has the same computational complexity as deciding whether an MCS is consistent, assuming that the complexity of the MCS includes the complexity of managing. This also shows that mMCS are not more expressive than MCS, but since we also show that the mMCS framework captures the MCS framework, it establishes that mMCS allow a more detailed study of inconsistency management in MCS without additional cost.

1.4 Structure

This thesis is structured as follows.

- Chapter 2 introduces preliminary notions of Multi-Context Systems and accompanying notions of abstract logics, bridge rules, and equilibrium semantics.
- In Chapter 3 the basic notions for explaining and removing inconsistency in MCS are introduced and investigated; the chapter contains refinement and modularity results for these notions, it states conversion and computational complexity results and it shows an encoding in logic programming to compute explanations.
- In Chapter 4 open and general ways to discriminate between possible ways of removing inconsistency in MCS are investigated. Two basic ways, filters and preferences, are given as well as multiple ways to realise these in an MCS using meta-reasoning transformations in such a way that the user may choose a concrete formalism for specifying a filter or preference. Correctness of these transformations is proven and the computational complexity of the transformations and additional notions is shown.
- In Chapter 5 managed Multi-Context Systems are introduced, where each context is accompanied by a manager that is tailored to the specific formalism used in the context. While in MCS the information flow only adds information, in an mMCS the information from other contexts may trigger the manager to perform an arbitrary action, like applying a belief revision operator on a context using classical logic, or applying logic program updates on a context employing an answer-set program. It is investigated how consistency-ensuring managers influence the consistency of the overall system and the computational complexity of mMCS and its expressiveness are shown to be the same as for ordinary MCS.
- Chapter 6 relates the work presented here with other approaches and solutions to inconsistency in knowledge-exchange systems.
- Finally, Chapter 7 concludes this work and gives an outlook on open problems and future work.

1.5 Publications

This thesis originates from a research project on MCS[1], and is in part based on material that has been published in preliminary form. Chapter 2 and Chapter 3 are based on and use material from [52], [53], and [54]. The basic notions of diagnosis and explanation in Chapter 3 were jointly developed with Peter Schüller (cf. [117]) and some of the conversion properties were also analysed together (on the other hand, refinements of the basic notions, modularity properties, and the logic programming encoding for explanation computation are the work of the author). Related work in Chapter 6 also uses text from [53] and [54].

Chapter 4 on preferences is based on [55] and [126]; it uses material from these publications but significantly extends these by a more rigid investigation of possible realisations of meta-reasoning in MCS, and it corrects several issues with the original approach.

Chapter 5 on local inconsistency management is based on [32] and uses material from this publication. It extends it by missing proofs and some minor corrections.

[1] Vienna Science and Technology Fund (WWTF) project ICT 08-020

CHAPTER 2

The Multi-Context Systems Framework

In this chapter we recall the framework of Multi-Context Systems (MCS) as introduced in [29], which constitutes the basis of this thesis. MCS are a powerful framework for heterogeneous nonmonotonic knowledge-integration, meaning that information specified in a large variety of knowledge-representation formalisms can be exchanged. This exchange may be nonmonotonic, i.e., not only the presence, but also the absence of information may be used to infer new information. The MCS framework is based on three basic concepts: abstract logics to capture any knowledge-representation formalism, bridge rules to specify the information exchange, and contexts which represent concrete instances of knowledge bases; an MCS then simply is a collection of such contexts and their respective bridge rules. Finally, the semantics of an MCS is given in terms of equilibria. In the following these concepts are presented in detail.

2.1 Abstract Logics

To cover all kinds of knowledge-representation formalisms, MCS capture each formalism using a very general concept called an abstract logic (or just logic).

Definition 2.1. *An* abstract "logic" *L, is a triple $L = (\mathbf{KB}, \mathbf{BS}, \mathbf{ACC})$ where:*

- $\mathbf{KB}$ *is the set of knowledge bases of L. We assume each element of $\mathbf{KB}$ is a set of elements ("formulas").*
- $\mathbf{BS}$ *is the set of possible belief sets, where the elements of a belief set are statements that possibly hold or, beliefs adopted by an agent.*
- $\mathbf{ACC} : \mathbf{KB} \rightarrow 2^{\mathbf{BS}}$ *is a function describing the semantics of the logic by assigning to each knowledge base a set of acceptable belief sets.*

Intuitively, each knowledge base $kb \in \mathbf{KB}$ is a set of "well-formed formulas" while each belief set $bs \in \mathbf{BS}$ is a set of "beliefs" (statements) that a reasoner may jointly hold. The acceptability function $\mathbf{ACC}(kb)$ singles out, given a knowledge base $kb \in \mathbf{KB}$, those sets of beliefs that are acceptable according to some reasoning method given kb. $\mathbf{ACC}$ is a multi-valued function to capture nonmonotonic formalisms, where a knowledge base may have multiple acceptable belief sets (as e.g. in Answer-Set Programming [57, 71, 72], reasoning with Default Theories [112], or in Abstract Argumentation [47]).

Depending on the concrete situation, e.g. given an existing legacy system or a theorem prover for a specific logic, different formalisations for some logic might be used. Therefore there is no fixed mapping between a given logic and an abstract logic representing the given logic.

An advantage of this very general approach is that various formalisms for knowledge representation can be captured, e.g. relational databases, logic programs, description logics, and propositional logic.

The following examples show how some typical knowledge-representation formalisms can be captured using abstract logics. These logics are used throughout this thesis in various examples.

Example 2.1 (Classical Propositional Logic)**.** *To capture classical (propositional) logic over a set Σ of propositional atoms, we may define:*

- $\mathbf{KB}^c = 2^{\Sigma^{wff}}$ *is the set of all subsets of* Σ^{wff}*, where* Σ^{wff} *is the set of well-formed formulas over* Σ *built using the connectives* $\wedge, \vee, \neg, \rightarrow$*;*
- $\mathbf{BS}^c = 2^{\Sigma^{wff}}$*, i.e., each set of formulas is a possible belief set; and*
- $\mathbf{ACC}^c$ *returns for each set* $kb \in \mathbf{KB}^c$ *of well-formed formulas a singleton set that contains the set of formulas entailed by* kb*; if* $\models_c$ *denotes classical entailment, then* $\mathbf{ACC}^c(kb) = \{\{F \in \Sigma^{wff} \mid kb \models_c F\}\}$.

The resulting logic $L^c_\Sigma = (\mathbf{KB}^c, \mathbf{BS}^c, \mathbf{ACC}^c)$ *captures entailment in classical logics. Following common terminology, we call* Σ *the signature of the logic.*

Observe that any tautological formula is entailed by any knowledge base, hence any $bs \in \mathbf{ACC}^c(kb)$ *for some* $kb \in \mathbf{KB}^c$ *is infinite (given that* Σ *is non-empty). In practice, therefore the formulas in knowledge bases and belief sets might be restricted to particular forms, e.g., to literals; we denote the logic where belief sets are restricted to literals by* $L^{pl}_\Sigma = (\mathbf{KB}, \mathbf{BS}, \mathbf{ACC})$*, where* $\mathbf{BS} = \{bs \in \mathbf{BS}^c \mid bs \subseteq \{A, \neg A \mid A \in \Sigma\}\}$*,* $\mathbf{KB} = \mathbf{KB}^c$*, and*

$$\mathbf{ACC}(kb) = \{\{A \in \Sigma \mid kb \models_c A\} \cup \{\neg A \mid A \in \Sigma \text{ and } kb \models_c \neg A\}\}\,.$$

Example 2.2. *Consider a propositional logic based on the abstract logic of Example 2.1 to reason about the "weather" conditions in a front lawn. We want to express whether the grass is wet, whether it is raining, whether a rainbow can be seen and whether the lawn sprinkler is turned on; so* $\Sigma = \{grass\_is\_wet, rainbow\_visible, is\_raining, sprinkler\_on\}$ *and the respective abstract logic is* $L^{pl}_\Sigma = (\mathbf{KB}, \mathbf{BS}, \mathbf{ACC})$.

The following set T of formulas is used to express that the grass is wet if the sprinkler is on or if it is raining, that the sprinkler is on or it is raining, and that no rainbow is visible.

$$\begin{aligned} T = \{ & \mathit{sprinkler\_on} \vee \mathit{is\_raining}, \\ & \mathit{sprinkler\_on} \rightarrow \mathit{grass\_is\_wet}, \\ & \mathit{is\_raining} \rightarrow \mathit{grass\_is\_wet}, \\ & \neg \mathit{rainbow\_visible} \} \end{aligned}$$

T is a knowledge base of L^{pl}_{Σ}, i.e., it holds that $T \in \mathbf{KB}$. The set of acceptable belief sets of T is $\mathbf{ACC}(T) = \{S\}$ where $S = \{\mathit{grass\_is\_wet}, \neg \mathit{rainbow\_visible}\}$. Since L^{pl}_{Σ} is constructed such that only entailed literals occur in the acceptable belief sets, S is the only such set.

Note that $\mathbf{ACC}(T)$ above is a set containing the belief set S, since the formalism of abstract logics allows multiple acceptable belief sets. This is especially useful for capturing Answer-Set Programs (cf. Example 2.5), but it might also be useful for classical logics if one wants to do model-based reasoning, i.e., one can design an abstract logic where every belief set corresponds to a model; naturally, a set of formulas can have multiple models, hence multiple accepted belief sets are also useful in such a case.

Example 2.3 (Description Logic)**.** *For ontologies with syntax and semantics of the description logic $\mathcal{ALC}$ (see [4]), we use the abstract logic $L^{\mathcal{ALC}}_{\Sigma}$ defined as follows.*

Over a signature Σ of atomic concepts $\mathbb{C}$, roles $\mathbb{R}$, and individuals I, T-Box axioms and A-Box axioms are defined based on concepts. The latter are inductively defined as follows: every atomic concept is a concept, the universal concept $\top$ and the bottom concept $\bot$ are concepts, and if C, D are concepts and $R \in \mathbb{R}$ is a role, then $C \sqcap D$, $C \sqcup D$, $\neg C$, $\forall R.C$, and $\exists R.C$ are concepts. Given concepts C, D, a role $R \in \mathbb{R}$, and individuals $a, b \in I$, a T-Box axiom (terminological axiom) is a formula of the form $C \sqsubseteq D$, and an A-Box axiom (assertional axiom) is either of the form $a : C$, or of the form $(a, b) : R$. Finally, $\mathcal{ALC}$ axioms are either T-Box axioms or A-Box axioms. Then, $L^{\mathcal{ALC}}_{\Sigma}$ is composed of

- $\mathbf{KB}$*, being the collection of sets of finite $\mathcal{ALC}$ axioms,*
- $\mathbf{BS}$*, being the set of possibly believed assertions, i.e.,* $\mathbf{BS}$ *is the powerset of the set of atomic A-Box axioms, and*
- $\mathbf{ACC}$*, being a mapping from knowledge bases to the set of assertions entailed by the knowledge base. For our purpose, $\mathbf{ACC}(kb) = \{S\}$ where S is the set of atomic A-Box axioms entailed by kb (see [4] for details).*

An $L^{\mathcal{ALC}}_{\Sigma}$-knowledge base contains both A-Box and T-Box axioms. An accepted belief set of such a knowledge base is the set of atomic assertions that follow from the knowledge base.

Example 2.4. *Consider a description logic $L^{\mathcal{ALC}}_{\Sigma} = (\mathbf{KB}, \mathbf{BS}, \mathbf{ACC})$ modelling knowledge about cars, where $\mathbb{C} = \{\mathit{Car}, \mathit{Vehicle}\}$, $\mathbb{R} = \emptyset$, and $I = \{a, d\}$. To state that every car is a* *vehicle, and that d is a car, the knowledge base (i.e., the union of the respective T-Box and A-Box) is as follows:*

$$kb = \{Car \sqsubseteq Vehicle,\ d : Car\}$$

The set of assertions entailed by kb is $S = \{d : Car,\ d : Vehicle\}$ and hence $\mathbf{ACC} = \{S\}$.

Example 2.5 (Disjunctive Answer Set Programming)**.** *For normal disjunctive logic programs under answer set semantics over a non-ground signature Σ (cf. [111] and [64]), we use the abstract logic $L_\Sigma^{asp} = (\mathbf{KB}, \mathbf{BS}, \mathbf{ACC})$, which is defined as follows:*

- $\mathbf{KB}$ *is the set of normal disjunctive logic programs over Σ, i.e., each $kb \in \mathbf{KB}$ is a set of rules of the form*

  $$a_1 \vee \ldots \vee a_n \leftarrow b_1, \ldots, b_i, not\, b_{i+1}, \ldots, not\, b_m,$$

  where all a_i, b_j, are atoms over a first-order language Σ, and $n + m > 0$. Let r be a rule of the aforementioned form, then $H(r) = \{a_1, \ldots, a_n\}$, $B^+(r) = \{b_1, \ldots, b_i\}$, $B^-(r) = \{b_{i+1}, \ldots, b_m\}$, and $B(r) = B^+(r) \cup B^-(r)$. Each rule $r \in kb$ must be safe, i.e., $vars(H(r)) \cup vars(B^-(r)) \subseteq vars(B^+(r))$, where for a set of atoms A, $vars(A) = \{vars(a) \mid a \in A\}$ and $vars(a)$ is the set of variables occurring in the atom a,

- $\mathbf{BS}$ *is the set of Herbrand interpretations over Σ, i.e, each $bs \in \mathbf{BS}$ is a set of ground atoms from Σ, and*

- $\mathbf{ACC}(kb)$ *returns the set of kb's answer sets: for $P \in \mathbf{KB}$ and $T \in \mathbf{BS}$, let $P^T = \{r \in grnd(P) \mid T \models B(r)\}$ be the FLP-reduct (cf. [64]) of P wrt. T, where $grnd(P)$ returns the ground instances of all rules in P. Then $bs \in \mathbf{BS}$ is an answer set, i.e., $bs \in \mathbf{ACC}(kb)$, iff bs is a $\subseteq$-minimal model of $kb^{bs} = \{r \in grnd(kb) \mid bs \models B(r)\}$.*

It is well-known for ASP that constraints (i.e., rules whose head is $\perp$) are expressible using rules as in $L_\Sigma^{asp} = (\mathbf{KB}, \mathbf{BS}, \mathbf{ACC})$. We thus allow constraints in knowledge bases and note that for any $bs \in \mathbf{BS}$ and $kb \in \mathbf{KB}$ it holds that: if there exists some $r \in kb^{bs}$ with $H(r) = \{\perp\}$ then this bs cannot be an answer set, i.e., $bs \notin \mathbf{ACC}(kb)$.

Example 2.6. *Consider a nonmonotonic ASP program which states that the sun is shining whenever it is not cloudy and vice versa. We employ L_Σ^{asp} with $\Sigma = \{sunshine, cloudy\}$ and the following knowledge base:*

$$kb = \{sunshine \leftarrow not\ cloudy. \quad cloudy \leftarrow not\ sunshine.\}$$

Observe that kb has exactly two answer sets, namely $S = \{sunshine\}$ and $S' = \{cloudy\}$, hence $\mathbf{ACC}(kb) = \{S, S'\}$.

Example 2.7 (Relational Database)**.** *We capture relational databases using the abstract logic $L_\Sigma^{DB} = (\mathbf{KB}, \mathbf{BS}, \mathbf{ACC})$ over a (finite) signature Σ composed of a set C of constant symbols and a set R of predicate symbols and corresponding arity. L_Σ^{DB} is defined as follows:*

- $\mathbf{KB}$ *is the powerset of facts over* Σ*, i.e.,* $\mathbf{KB} = 2^A$ *where*

$$A = \{r(c_1, \ldots, c_k) \mid r \in R \textit{ with arity } k, c_1, \ldots, c_k \in C\},$$

- $\mathbf{BS} = \mathbf{KB}$*, and*
- $\mathbf{ACC}(kb) = \{kb\}$*, for any* $kb \in \mathbf{KB}$.

Note that a much more involved logic is necessary to fully capture relational databases and important concepts like the closed-world assumption (cf. [1]). For illustration purposes, however, this very simple view on databases is sufficient.

In the remainder of this thesis we sometimes omit the explicit definition of the signature Σ for an abstract logic L^{pl}_{Σ}, $L^{\mathcal{ALC}}_{\Sigma}$, L^{asp}_{Σ}, or L^{DB}_{Σ} if Σ is clear from the context.

2.2 Bridge Rules

To specify information exchange between contexts, so-called bridge rules are used. Bridge rules are similar in form and behaviour to rules in logic programming. They differ from each other by the fact that bridge rules are based on beliefs from (possibly) different abstract logics and corresponding contexts. Based on the presence (or absence) of beliefs at other contexts, a bridge rule can add information to a context.

Definition 2.2. *Given a sequence* $L = (L_1, \ldots, L_n)$ *of abstract logics where for* $1 \leq j \leq n$, $L_j = (\mathbf{KB}_j, \mathbf{BS}_j, \mathbf{ACC}_j)$*. An* L^k-bridge rule over L *with* $1 \leq k \leq n$ *is of the following form:*

$$(k : s) \leftarrow (c_1 : p_1), \ldots, (c_i : p_i), \mathbf{not}\ (c_{i+1} : p_{i+1}), \ldots, \mathbf{not}\ (c_m : p_m). \tag{2.1}$$

where for each $1 \leq i \leq m$ *we have that* $c_i \in \{1, \ldots, n\}$*,* p_i *is an element of some belief set of the abstract logic* L_{c_i} *(i.e.,* $p_i \in \bigcup \mathbf{BS}_{c_i}$*), and* s *is a knowledge base formula of* L_k *(i.e.,* $s \in \bigcup \mathbf{KB}_k$*).*

Each bridge rule in an MCS is associated to a certain context in such a way that all L^k bridge rules belong to the context with identifier k.

We denote by $\varphi(r)$ the formula s in the head of r and by $C_h(r)$ the context k where r belongs to. The full head of r is denoted by $head(r) = (k : s)$, thus $head(r) = (C_h(r) : \varphi(r))$. The literals in the body of r are referred to by $body^{\pm}(r)$, $body^{+}(r)$, $body^{-}(r)$, $body(r)$, which denotes the set $\{(c_1 : p_1), \ldots, (c_m : p_m)\}$, $\{(c_1 : p_1), \ldots, (c_j : p_j)\}$, $\{(c_{j+1} : p_{j+1}), \ldots, (c_m : p_m)\}$, $\{(c_1 : p_1), \ldots, (c_j : p_j), \mathbf{not}\ (c_{j+1} : p_{j+1}), \ldots, \mathbf{not}\ (c_m : p_m)\}$, respectively.

Furthermore, $C_b(r)$ denotes the set of contexts referenced in r's body, i.e., $C_b(r) = \{c_i \mid (c_i : p_i) \in body^{\pm}(r)\}$. Note that different from [29], the head of r contains not only the knowledge-base formula s but also the context identifier k. This choice merely is syntactic sugar and allows easier identification of the context where r belongs to. For technical use later, we denote by $cf(r)$ the *condition-free* bridge rule resulting from r by removing all elements in its body, i.e., $cf(r)$ is $(k : s) \leftarrow .$ and for any set of bridge rules R, we let $cf(R) = \bigcup_{r \in R} cf(r)$.

Example 2.8. *Consider the sequence* $L = (L^{asp}_{\Sigma'}, L^{pl}_{\Sigma''}) = (L_1, L_2)$ *of abstract logics where* $L^{asp}_{\Sigma'}$ *and* $L^{pl}_{\Sigma''}$ *are defined as in Examples 2.6 and 2.2, respectively, except that their signatures are* $\Sigma' = \{sunshine, cloudy\}$ *and* $\Sigma'' = \{cloudy, grass\_is\_wet, rainbow\_visible, is\_raining, sprinkler\_on\}$*, respectively.*

We now present two L^2 *bridge rules over L, which we denote by* r_1 *and* r_2*. Intuitively, the bridge rule* r_1 *derives no rain (*$\neg is\_raining$*), if at* 1 *it is believed that the sun is shining (*$sunshine$*). The bridge rule* r_2 *derives that it is cloudy (*$cloudy$*), if at* 2 *it is believed that it is raining (*$is\_raining$*) and at* 1 *it is not believed that the sun is shining (*$sunshine$*).*

$$
\begin{aligned}
r_1 :& \quad (2 : \neg is\_raining) \leftarrow (1 : sunshine). \\
r_2 :& \quad (2 : cloudy) \leftarrow (2 : is\_raining), \mathbf{not}\ (1 : sunshine).
\end{aligned}
$$

Since r_2 *derives information for* 2 *and it also refers to beliefs at* 2*,* r_2 *is an example of cyclic information flow. It also demonstrates non-monotonicity, since it refers to the absence of the belief in sunshine.*

Using the introduced notation about bridge rules, we observe that

$$
\begin{aligned}
\varphi(r_1) &= \neg is\_raining & \varphi(r_2) &= cloudy \\
body(r_1) &= \{(1 : sunshine)\} & body(r_2) &= \{(2 : is\_raining), \mathbf{not}\ (1 : sunshine)\} \\
body^{+}(r_1) &= \{(1 : sunshine)\} & body^{+}(r_2) &= \{(2 : is\_raining\} \\
body^{\pm}(r_1) &= \{(1 : sunshine)\} & body^{\pm}(r_2) &= \{(2 : is\_raining), (1 : sunshine)\} \\
body^{-}(r_1) &= \emptyset & body^{-}(r_2) &= \{(1 : sunshine)\} \\
C_b(r_1) &= \{1\} & C_b(r_2) &= \{1, 2\} \\
C_h(r_1) &= 2 & C_h(r_2) &= 2
\end{aligned}
$$

Also note that the condition-free versions of r_1 *and* r_2 *are as follows:*

$$
\begin{aligned}
cf(r_1) :& \quad (2 : \neg is\_raining) \leftarrow . \\
cf(r_2) :& \quad (2 : cloudy) \leftarrow .
\end{aligned}
$$

Observe that bridge rules only deal with elements of knowledge bases and elements of belief sets, both of which are considered to be atomic expressions from the perspective of MCS. Incorporating variables into bridge rules is possible but requires restrictions on context logics or additional machinery for variable substitution (cf. [8, 65, 118] for details).

2.3 Multi-Context Systems

An abstract logic together with one of its knowledge bases and a set of bridge rules is called a *context*; formally, a context C is a triple $C = (L, kb, br)$ such that $L = (\mathbf{KB}, \mathbf{BS}, \mathbf{ACC})$ is an abstract logic, $kb \in \mathbf{KB}$ is a knowledge base, and br is a set of bridge rules with respect to a suitable set of logics (cf. details below). Each context captures a concrete instance of a (legacy) knowledge system and extends it with bridge rules. A Multi-Context System then simply is a sequence of contexts, where the bridge rules of each context are defined with respect to the sequence of logics employed in the contexts of the MCS. Formally:

Definition 2.3. *A* Multi-Context System $M = (C_1, \ldots, C_n)$ *is a collection of contexts* $C_i = (L_i, kb_i, br_i)$*,* $1 \leq i \leq n$*, where*

- $L_i = (\mathbf{KB}_i, \mathbf{BS}_i, \mathbf{ACC}_i)$ *is an abstract logic,*
- $kb_i \in \mathbf{KB}_i$ *is a knowledge base, and*
- br_i *is a set of* L^i*-bridge rules over* $L = (L_1, \ldots, L_n)$*.*

Furthermore, for each $H \subseteq \{\varphi(r) \mid r \in br_i\}$ *it holds that* $(kb_i \cup H) \in \mathbf{KB}_i$*, i.e., adding bridge rule heads to a knowledge base again yields a knowledge base.*

Note that br_i consists of L^i-bridge rules, which means that all of their heads have i as their context identifier. Formally: if $r \in br_i$ then $C_h(r) = i$.

If two MCS are considered at the same time, which occurs often in the remainder of this thesis, easy access to the bridge rules of a certain context of an MCS comes in handy. We therefore introduce the following notation: $br(M) = \bigcup_{i=1}^{n} br_i$ denotes the set of all bridge rules of M, $C(M) = \{1, \ldots, n\}$ denotes the set of all context identifiers of M, and $br_i(M)$ denotes the set of bridge rules of context i of M, i.e., $br_i(M) = \{r \in br(M) \mid C_h(r) = i\}$.

Example 2.9. *Consider an MCS* M *representing a health care decision support system*[1] *that contains the following contexts: a patient history database (*C_1*), a blood and X-Ray analysis database (*C_2*), a disease ontology (*C_3*), and a decision support system (*C_4*) which suggests suitable treatments, formally:* $M = (C_1, C_2, C_3, C_4)$*. Note that for illustration purposes, the MCS only covers a single patient.*

The contexts C_1 *and* C_2 *use logics* $L^{pl}_{\Sigma_1}$ *and* $L^{pl}_{\Sigma_2}$ *as introduced in Example 2.1 over signatures* $\Sigma_1 = \{allergy\_strong\_ab\}$ *and* $\Sigma_2 = \{blood\_marker, xray\_pneumonia\}$*, respectively. Their knowledge bases are as follows:*

$$kb_1 = \{allergy\_strong\_ab\},$$
$$kb_2 = \{\neg blood\_marker, xray\_pneumonia\}.$$

Those knowledge bases provide information that the patient is allergic to strong antibiotics (kb_1*) and that a certain blood marker is not present while pneumonia was detected in an X-ray examination (*kb_2*).*

The corresponding semantics is given by $\mathbf{ACC}(kb_1) = \big\{\{allergy\_strong\_ab\}\big\}$ *for* C_1*, and* $\mathbf{ACC}(kb_2) = \big\{\{\neg blood\_marker, xray\_pneumonia\}\big\}$ *for* C_2*.*

We use an ontology about diseases, given by context C_3 *using* $L^{\mathcal{ALC}}_{\Sigma_3}$ *from Example 2.3 with the signature* Σ_3 *containing concepts* $Pneumonia, BacterialDisease, AtypPneumonia$*, individuals d and m1, and the role* has_marker*. Its knowledge base,* kb_3*, consists of two axioms, where the first states that pneumonia is a bacterial disease and the second that pneumonia together*

[1]Throughout this thesis several examples are inspired by the medical domain. Note that medical information conveyed by these examples may be incorrect or wrong.

with an associated blood-marker indicates atypical pneumonia (a severe form of pneumonia). The corresponding knowledge base is:

$$\begin{aligned} kb_3 = \{&Pneumonia \sqsubseteq BacterialDisease, \\ &Pneumonia \sqcap \exists has\_marker.\top \sqsubseteq AtypPneumonia\}. \end{aligned}$$

As kb_3 is satisfiable and contains only terminological knowledge, no assertions follow from this knowledge base, thus $\mathbf{ACC}(kb_3) = \{\emptyset\}$. Adding the assertions that d is pneumonia and that the role has_marker holds between d and a marker $m1$ results in the conclusion that d is also a bacterial disease and atypical pneumonia, i.e.,

$$\begin{aligned} &\mathbf{ACC}\big(kb_3 \cup \{d : Pneumonia, (d, m1) : has\_marker\}\big) = \\ &\big\{\{d : Pneumonia, d : BacterialDisease, d : AtypPneumonia, (d, m1) : has\_marker\}\big\}. \end{aligned}$$

For the context C_4 that is suggesting proper treatments, we employ $L^{asp}_{\Sigma_4}$ from Example 2.5 where $\Sigma_4 = \{give\_strong, give\_weak, need\_ab, allow\_strong\_ab, give\_nothing\}$. The knowledge base for C_4 is:

$$\begin{aligned} kb_4 = \{&give\_strong \vee give\_weak \leftarrow need\_ab. \\ &give\_strong \leftarrow need\_strong. \\ &\bot \leftarrow give\_strong, not\ allow\_strong\_ab. \\ &give\_nothing \leftarrow not\ need\_ab, not\ need\_strong.\}. \end{aligned}$$

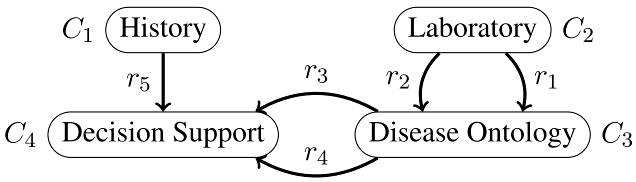
Depending on whether antibiotics or strong antibiotics is needed, C_4 suggests a treatment which is either a strong antibiotics, a weak antibiotics, or no medication at all. Without further information, kb_4 thus concludes that nothing is required, i.e., $\mathbf{ACC}(kb_4) = \big\{\{give\_nothing\}\big\}$. If $need\_ab$ and $allow\_strong\_ab$ are added, however, kb_4 results in two answer sets, i.e., $\mathbf{ACC}\big(kb_4 \cup \{need\_ab., allow\_strong\_ab.\}\big) = \{A_1, A_2\}$ where $A_1 = \{give\_strong, need\_ab, allow\_strong\_ab\}$ and $A_2 = \{give\_weak, need\_ab, allow\_strong\_ab\}$. Note that by the constraint in kb_4 the strong antibiotics is only given if allowed, hence $\mathbf{ACC}\big(kb_4 \cup \{need\_ab.\}\big) = \big\{\{give\_weak, need\_ab\}\big\}$.

The bridge rules of the MCS are given as follows:

$$\begin{aligned} &r_1: && (3 : d : Pneumonia) \leftarrow (2 : xray\_pneumonia). \\ &r_2: && (3 : (d, m1) : has\_marker) \leftarrow (2 : blood\_marker). \\ &r_3: && (4 : need\_ab.) \leftarrow (3 : d : BacterialDisease). \\ &r_4: && (4 : need\_strong.) \leftarrow (3 : d : AtypPneumonia). \\ &r_5: && (4 : allow\_strong\_ab.) \leftarrow \mathbf{not}\ (1 : allergy\_strong\_ab). \end{aligned}$$

Rules r_1 and r_2 (belonging to context C_3) provide input for disease classification to the ontology; they assert facts about the disease 'd' and the blood marker '$m1$'. Rules r_3 and r_4 (belonging to context C_4) link disease information with medication requirements, while r_5 (also belonging to context C_4) relates acceptance of strong antibiotics with an allergy check on the patient database. The sets of bridge rules of each context are as follows: $br_1 = br_2 = \emptyset$, $br_3 = \{r_1, r_2\}$, and $br_4 = \{r_3, r_4, r_5\}$.

An overview of the knowledge bases and bridge rules of M is given in Figure 2.1, where each bridge rule $r \in \{r_1, \ldots, r_5\}$ with $C_h(r) = j$ and $i \in C_b(r)$ is depicted as an arrow from C_i to C_j. The figure also shows the set of equilibria $\text{EQ}(M)$*, the semantics of M which is explained in detail below.*



r_1: $(3 : d : \mathit{Pneumonia}) \leftarrow (2 : \mathit{xray\_pneumonia})$.
r_2: $(3 : (d, \mathit{m1}) : \mathit{has\_marker}) \leftarrow (2 : \mathit{blood\_marker})$.
r_3: $(4 : \mathit{need\_ab}.) \leftarrow (3 : d : \mathit{BacterialDisease})$.
r_4: $(4 : \mathit{need\_strong}.) \leftarrow (3 : d : \mathit{AtypPneumonia})$.
r_5: $(4 : \mathit{allow\_strong\_ab}.) \leftarrow \textbf{not}\ (1 : \mathit{allergy\_strong\_ab})$.

$$kb_1 = \{\mathit{allergy\_strong\_ab}\}$$
$$kb_2 = \{\neg \mathit{blood\_marker}, \mathit{xray\_pneumonia}\}$$
$$kb_3 = \{\mathit{Pneumonia} \sqsubseteq \mathit{BacterialDisease}, \\ \mathit{Pneumonia} \sqcap \exists \mathit{has\_marker}.\top \sqsubseteq \mathit{AtypPneumonia}\}$$
$$kb_4 = \{\mathit{give\_strong} \vee \mathit{give\_weak} \leftarrow \mathit{need\_ab}. \\ \mathit{give\_strong} \leftarrow \mathit{need\_strong}. \\ \bot \leftarrow \mathit{give\_strong}, \mathit{not\ allow\_strong\_ab}. \\ \mathit{give\_nothing} \leftarrow \mathit{not\ need\_ab}, \mathit{not\ need\_strong}.\}$$

$$br_1 = \emptyset \qquad br_2 = \emptyset$$
$$br_3 = \{r_1, r_2\} \qquad br_4 = \{r_3, r_4, r_5\}$$
$$C_1 = (L^{pl}_{\Sigma_1}, kb_1, br_1) \qquad C_2 = (L^{pl}_{\Sigma_2}, kb_2, br_2)$$
$$C_3 = (L^{\mathcal{ALC}}_{\Sigma_3}, kb_3, br_3) \qquad C_4 = (L^{asp}_{\Sigma_4}, kb_4, br_4)$$

$$\text{EQ}(M) = \{(\{\mathit{allergy\_strong\_ab}\}, \{\neg \mathit{blood\_marker}, \mathit{xray\_pneumonia}\}, \\ \{d : \mathit{Pneumonia}, d : \mathit{BacterialDisease}\}, \{\mathit{need\_ab}, \mathit{give\_weak}\})\}$$

Figure 2.1: The MCS $M = (C_1, C_2, C_3, C_4)$ from Example 2.9 with contexts C_i, bridge rules br_i, and knowledge bases kb_i, $1 \leq i \leq 4$.

2.4 Equilibrium Semantics

Recall that an abstract logic $L = (\mathbf{KB}, \mathbf{BS}, \mathbf{ACC})$ captures the semantics of a formalism via its acceptability function $\mathbf{ACC} : \mathbf{KB} \to 2^{\mathbf{BS}}$ which designates some belief sets as acceptable under a given knowledge base. Belief sets therefore form the basis of the semantics of MCS. More precisely, the semantics of MCS is based on sequences of belief sets, i.e., one belief set per context. A sequence of belief sets of an MCS is called a belief state. Formally, given an MCS $M = (C_1, \ldots, C_n)$ with $C_i = (L_i, kb_i, br_i)$ and $L_i = (\mathbf{KB}_i, \mathbf{BS}_i, \mathbf{ACC}_i)$, a *belief state* of M is a sequence $S = (S_1, \ldots, S_n)$ of belief sets $S_i \in \mathbf{BS}_i$, $1 \leq i \leq n$.

Given a belief state S of M, one can evaluate for all bridge rules $r \in br(M)$ whether the body of r is satisfied in S, i.e., whether r is applicable in S. Formally, a bridge rule r of form (2.1) is *applicable* in a belief state S, denoted by $S \models\!\!\leadsto r$, iff for all $(j : p) \in body^+(r)$ it holds that $p \in S_j$, and for all $(j : p) \in body^-(r)$ it holds that $p \notin S_j$. For a set R of bridge rules and a belief state S, $app(R, S)$ denotes the set of bridge rules of R that are applicable in S, i.e., $app(R, S) = \{r \in R \mid S \models\!\!\leadsto r\}$.

Equilibrium semantics designates some belief states as acceptable. Intuitively, it selects a belief state $S = (S_1, \ldots, S_n)$ of an MCS M as acceptable, if each context C_i takes the heads of all its bridge rules that are applicable in S into account to enrich its knowledge base with, and accepts its designated belief set S_i under this enlarged knowledge base.

Definition 2.4. *A belief state $S = (S_1, \ldots, S_n)$ of M is an* equilibrium *iff for every belief set S_i, $1 \leq i \leq n$, it holds that*

$$S_i \in \mathbf{ACC}_i\big(kb_i \cup \{\varphi(r) \mid r \in app(br_i, S)\}\big).$$

The set of all equilibria of an MCS M is denoted by $\text{EQ}(M)$.

An alternative definition of equilibrium is based on $app_i(S, M)$, the knowledge-base formulas of bridge rules of context C_i that are applicable in S, i.e., $app_i(S, M) = \{\varphi(r) \mid r \in br_i(M), S \models\!\!\leadsto r\}$. In these terms a belief state $S = (S_1, \ldots, S_n)$ of an MCS M is an equilibrium iff for all $1 \leq i \leq n$ it holds that $S_i \in \mathbf{ACC}_i\big(kb_i \cup app_i(S, M)\big)$. Since $app_i(S, M) = \{\varphi(r) \mid r \in app(br_i(M), S)\}$, the two definitions of an equilibrium are equal.

Example 2.10. *The MCS M of Example 2.9 (cf. Figure 2.1) has a single equilibrium S with $S = (S_1, S_2, S_3, S_4)$ where*

$$\begin{aligned} S_1 =& \{allergy\_strong\_ab\}, \\ S_2 =& \{\neg blood\_marker, xray\_pneumonia\}, \\ S_3 =& \{d : Pneumonia, d : BacterialDisease\}, \text{ and} \\ S_4 =& \{need\_ab, give\_weak\}. \end{aligned}$$

The only rules applicable in S are r_1 and r_3, as $app(br_1(M), S) = app(br_2(M), S) = \emptyset$, $app(br_3(M), S) = \{r_1\}$, and $app(br_4(M), S) = \{r_3\}$.

Note that if we replace S_4 with $\{need\_ab, give\_strong, allow\_strong\_ab\}$, then the resulting belief state is not an equilibrium: C_4 uses answer set semantics, therefore $allow\_strong\_ab$ cannot be part of S_4 unless it is added by a bridge rule. The only bridge rule with this head is r_5 but it is not applicable because of the presence of $allergy\_strong\_ab$ in kb_1 and in S_1.

To create bridge rules that are always applicable or never applicable, we also allow bridge rules to contain the symbols $\top$ or $\bot$ as the single element in their body. Both $\top$ and $\bot$ are syntactic sugar for an empty body and a body containing $(\ell : p)$, $\mathbf{not}\ (\ell : p)$ where p is some belief of some context C_ℓ, respectively. For a bridge rule r of form $(k : s) \leftarrow \top.$ it therefore holds for all belief states S that $S \models\!\!\leadsto r$, while for a bridge rule r' of form $(k : s) \leftarrow \bot.$ it holds for all belief states S that $S \not\models\!\!\leadsto r'$. For such bridge rules, we let $\mathit{body}(r) = \{\top\}$, $\mathit{body}(r') = \{\bot\}$, and we consider their bodies to refer to no other contexts, i.e., $C_b\,(r) = C_b\,(r') = \emptyset$, as well as we regard those bridge rules to contain no literals in their bodies, i.e., $\mathit{body}^-(r) = \mathit{body}^+(r) = \mathit{body}^\pm(r) = \emptyset$ and $\mathit{body}^-(r') = \mathit{body}^+(r') = \mathit{body}^\pm(r') = \emptyset$. The remaining notions on bridge rules, $\varphi\,(r)$ and $C_h\,(r)$, are the same as for other bridge rules.

Given an MCS $M = (C_1, \ldots, C_n)$ over abstract logics $L = (L_1, \ldots, L_n)$ and a set R of bridge rules, we call R *compatible* with M, if there exists a partitioning $R_1, \ldots, R_n$ of R such that for every $1 \leq k \leq n$, $r \in R_k$ implies that r is an L^k-bridge rule over L. In the following chapters, we often consider modifications to the bridge rules of an MCS. We use the following notation to denote an MCS where bridge rules have been exchanged: given an MCS M and a set R of bridge rules such that R is compatible with M, we denote by $M[R]$ the MCS obtained from M by replacing its set of bridge rules $br(M)$ with R. For example, $M[br(M)] = M$ and $M[\emptyset]$ is M with no bridge rules at all.

Regarding equilibria, we write $M \models \bot$ to denote that M has no equilibrium, i.e., $\textsc{Eq}(M) = \emptyset$. Conversely, by $M \not\models \bot$ we denote the opposite, i.e., $\textsc{Eq}(M) \neq \emptyset$. As mentioned in Chapter 1, we consider an MCS to be inconsistent if it has no equilibrium, i.e., $\textsc{Eq}(M) = \emptyset$ and $M \models \bot$ are equivalent to saying that M is inconsistent.

CHAPTER 3

Basic Notions of Inconsistency

3.1 Introduction

Inconsistency in MCS arises easily due to unforeseen effects of the information exchange. Consider again the MCS of Example 2.9, where a hospital employs four already existing knowledge bases such that their combined knowledge provides decision support for patient medications. In Example 2.9 the patient is allergic to the strong antibiotic, while blood tests show that a certain marker is absent and X-Ray indicates pneumonia. Now consider the case that the marker is present:

Example 3.1. *Changing the MCS of Example 2.9 such that the blood serum analysis shows the presence of the blood marker, i.e.* $kb_2 = \{\mathit{blood\_marker}, \mathit{xray\_pneumonia}\}$*, yields the MCS* M *depicted in Figure 3.1.*

The only accepted belief sets of C_1 *and* C_2 *are*

$S_1 = \{\mathit{allergy\_strong\_ab}\}$ *and*
$S_2 = \{\mathit{blood\_marker}, \mathit{xray\_pneumonia}\}$*, respectively.*

Therefore bridge rules r_1 *and* r_2 *are applicable in any belief state which is acceptable at* C_1 *and* C_2*. Applicability of* r_1 *and* r_2 *in turn yields that the only accepted belief set of* C_3 *then is*

$$S_3 = \{d : \mathit{Pneumonia}, d : \mathit{BacterialDisease}, d : \mathit{AtypPneumonia}, (d, \mathit{m1}) : \mathit{has\_marker}\}.$$

Hence, r_3 *and* r_4 *are applicable in any belief state which is acceptable at* C_1*,* C_2*, and* C_3*.*
At C_4 *it is then concluded that strong antibiotics are required, while the constraint*

$$\bot \leftarrow \mathit{give\_strong}, \mathit{not}\ \mathit{allow\_strong\_ab}.$$

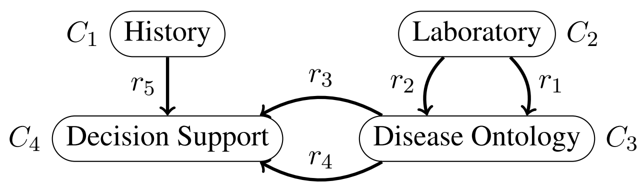
states that giving the strong antibiotics must be allowed explicitly. Since the patient's allergy does not allow giving the strong antibiotics, the constraint is violated and hence C_4 has no answer-set. This means that there exists no belief set S_4 which is accepted at C_4. In consequence, no belief state of M is an equilibrium.

Note that applicability of r_5 would resolve this inconsistency by activating $allow\_strong\_ab$. The applicability of r_5, however, is prevented by the presence of $allergy\_strong\_ab$ in S_1 together with the body literal '$\mathbf{not}\ (1 : allergy\_strong\_ab)$'.

Since the MCS has no equilibrium, one cannot draw any conclusion from the inconsistent MCS. Maybe the MCS encountered a real-world scenario which is not properly modelled by the MCS (arguably, this is the case for Example 3.1). In practice, however, there will be multiple patients and one forgotten use case for one patient should not make the whole system unusable. In such circumstances, it is paramount to somehow get rid of the inconsistency and the first step towards this is to identify the reasons of the inconsistency. In this chapter we therefore develop and investigate basic notions to explain and analyse the reasons of inconsistency. The aim of that is to understand where and how inconsistencies occur and how they can be removed. Since MCS model loosely integrated knowledge with autonomous sources (e.g. different companies linking parts of their business logic), we do not consider changing or modifying the data of contexts (no company will allow another to meddle with crucial business data just to repair some flaw in a common knowledge-exchange system). Therefore, we consider the data inside contexts as impossible to modify and the information flow as source of inconsistency. Since information exchange and interlinking in MCS is specified using bridge rules, we therefore focus on bridge rules as possible reasons of inconsistency. In this chapter, we assume that each context by itself is consistent, while in Chapter 5 we also investigate the case of inconsistent contexts and modifications to the internal data of contexts.

Observe that our approach is very different from data integration scenarios (cf. related work in Chapter 6) where integration is considered from a central perspective and data is modified. Speaking in data-integration terms, we consider changing the mapping rather than modifying the basic data. Another difference is that MCS allow cyclic information flow. Diagnosing faults in a monotonic system and identifying the reasons of inconsistency by minimal sets of inconsistent formulas is well-established (cf. Reiter's seminal work [113]). These notions, however, are not directly applicable to MCS, since MCS allow nonmonotonic reasoning, hence the concepts of a diagnosis and the reasons for an inconsistency have to be addressed differently than in monotonic reasoning systems. Non-monotonicity and cyclic information flow make the task of identifying and explaining inconsistency a non-trivial task.

This chapter introduces basic notions to explain the reasons of inconsistency and to identify diagnoses which allow to remove the inconsistency. This is followed by an investigation of refined notions of explanations and diagnoses where it is shown that those refinements can be simulated by the basic notions. A simplified notion, which only considers deleting bridge rules, is also considered. Several properties regarding diagnoses and explanations are established. These properties help to understand diagnoses and explanations in more detail, as well as they aid in computing diagnoses and explanations by pointing at possibilities for a significant reduction of



r_1: $(3 : d : \mathit{Pneumonia}) \leftarrow (2 : \mathit{xray\_pneumonia})$.
r_2: $(3 : (d, \mathit{m1}) : \mathit{has\_marker}) \leftarrow (2 : \mathit{blood\_marker})$.
r_3: $(4 : \mathit{need\_ab}.) \leftarrow (3 : d : \mathit{BacterialDisease})$.
r_4: $(4 : \mathit{need\_strong}.) \leftarrow (3 : d : \mathit{AtypPneumonia})$.
r_5: $(4 : \mathit{allow\_strong\_ab}.) \leftarrow \mathbf{not}\ (1 : \mathit{allergy\_strong\_ab})$.

$$
\begin{aligned}
kb_1 &= \{\mathit{allergy\_strong\_ab}\}\\
kb_2 &= \{\mathit{blood\_marker}, \mathit{xray\_pneumonia}\}\\
kb_3 &= \{\mathit{Pneumonia} \sqsubseteq \mathit{BacterialDisease},\\
&\qquad \mathit{Pneumonia} \sqcap \exists \mathit{has\_marker}.\top \sqsubseteq \mathit{AtypPneumonia}\}\\
kb_4 &= \{\mathit{give\_strong} \vee \mathit{give\_weak} \leftarrow \mathit{need\_ab}.\\
&\qquad \mathit{give\_strong} \leftarrow \mathit{need\_strong}.\\
&\qquad \bot \leftarrow \mathit{give\_strong}, \mathit{not\ allow\_strong\_ab}.\\
&\qquad \mathit{give\_nothing} \leftarrow \mathit{not\ need\_ab}, \mathit{not\ need\_strong}.\}
\end{aligned}
$$

$$
\begin{aligned}
br_1 &= \emptyset & br_2 &= \emptyset\\
br_3 &= \{r_1, r_2\} & br_4 &= \{r_3, r_4, r_5\}
\end{aligned}
$$

$$
\begin{aligned}
C_1 &= (L^{pl}_{\Sigma_1}, kb_1, br_1) & C_2 &= (L^{pl}_{\Sigma_2}, kb_2, br_2)\\
C_3 &= (L^{\mathcal{ALC}}_{\Sigma_3}, kb_3, br_3) & C_4 &= (L^{asp}_{\Sigma_4}, kb_4, br_4)
\end{aligned}
$$

Minimal diagnoses and explanations as defined in Section 3.2.

$$
\begin{aligned}
D^{\pm}_m(M) &= \{(\{r_1\}, \emptyset), (\{r_2\}, \emptyset), (\{r_4\}, \emptyset), (\emptyset, \{r_5\})\}\\
E^{\pm}_m(M) &= (\{r_1, r_2, r_4\}, \{r_5\})
\end{aligned}
$$

Figure 3.1: Running example MCS with contexts C_i, bridge rules br_i, and knowledge bases kb_i.

the search-space.Finally, a method is proposed to compute all explanations using so-called HEX-programs employing the technique of saturation from answer-set programming. The approach also has been implemented.

The remainder of this chapter is structured as follows. In Section 3.2 we introduce basic notions explaining and analysing inconsistency and we investigate possible refinements. Section 3.3 investigates properties of the basic notions, among these are conversion results, modularity properties and decomposition results. Section 3.4 gives a brief overview of computational complexities of decision problems related to inconsistency. Section 3.5 gives a short introduction to HEX-programs and presents a saturation-based approach to compute the basic notion of inconsistency explanation. Section 3.6 concludes this chapter and points at possible future work.

3.2 Diagnoses and Explanations for Inconsistency

As the combination and interaction of heterogeneous, possibly autonomous, systems can easily have unforeseen and intricate effects, inconsistency is a major problem in MCS. To provide support for restoring consistency, we seek to understand and give reasons for inconsistency.

Recall that by inconsistency we understand here that an MCS has no equilibrium. This differs from the classical understanding, where inconsistency usually occurs in the presence of contradictive knowledge.

Example 3.2. *Consider an MCS $M = (C_1, C_2)$ where the two contexts disagree on the truth of $is\_raining$, i.e., $\mathbf{ACC}_1(kb_1) = \{\{is\_raining\}\}$, $\mathbf{ACC}_2(kb_2) = \{\{\neg is\_raining\}\}$, $br(M) = \emptyset$, and kb_1 and kb_2 are the respective knowledge bases of C_1 and C_2.*

Then, M is not inconsistent, since $S = (\{is\_raining\}, \{\neg is\_raining\})$ is an equilibrium. Assuming that C_1 and C_2 are weather stations in different locations, our understanding of inconsistency is appropriate. Even if C_1 and C_2 are at the same locations, their sensor readings might differ withouth being considered inconsistent, e.g. due to known different sensitivity of C_1 and C_2 the above belief sets might just indicate a few raindrops.

Note that in Chapter 5 more strict forms of inconsistency are discussed. Recall that $M \models \bot$ denotes that M has no equilibrium, i.e., M is inconsistent, and $M \not\models \bot$ denotes the opposite.

In the following, we consider two possibilities for identifying the reasons of inconsistency in MCS: first, a consistency-based formulation, which identifies a part of the bridge rules which need to be changed to restore consistency. Second, an entailment-based formulation, which identifies a part of the bridge rules which is required to make the MCS inconsistent. Following common terminology, we call the first formulation a *diagnosis* (cf. [113]) and the second an *inconsistency explanation*.

Diagnosis

As well-known, adding knowledge in nonmonotonic reasoning can both cause and prevent inconsistency; the same is true for removing knowledge.

For our consistency-based explanation of inconsistency, we therefore consider pairs of sets of bridge rules, such that if we deactivate the rules in the first set, and add the rules in the second set in unconditional form, the MCS becomes consistent (i.e., admits an equilibrium). Adding rules unconditionally is the most severe form of modification of a rule's body, but as we see later, this notion also allows to capture more fine-grained forms of modification.

Definition 3.1. *Given an MCS M, a* diagnosis *of M is a pair (D_1, D_2), $D_1, D_2 \subseteq br(M)$, such that $M[br(M) \setminus D_1 \cup cf(D_2)] \not\models \bot$. We denote by $D^{\pm}(M)$ the set of all diagnoses.*

An alternative reading of this notion is that a diagnosis indicates which bridge rules are assumed to require modification in order to obtain a consistent MCS. By Occam's razor, a hypothesis is preferable to another one if it requires fewer assumptions, hence a minimal modification is preferable, because it assumes a minimal set of bridge rules to require modification. We thus prefer subset-minimal diagnoses to obtain a more relevant set of diagnoses.

For the remainder of this thesis, we extend the subset relation from sets to pairs of sets: given pairs $A = (A_1, A_2)$ and $B = (B_1, B_2)$ of sets, the pointwise subset relation $A \subseteq B$ holds iff $A_1 \subseteq B_1$ and $A_2 \subseteq B_2$; additionally, $A \subset B$ holds iff $A \subseteq B$ and $A \neq B$ both hold, where $A = B$ holds iff $A_1 = B_1$ and $A_2 = B_2$.

Definition 3.2. *Given an MCS M, $D^{\pm}_m(M)$ is the set of all pointwise subset-minimal diagnoses of an MCS M, i.e.,*

$$D^{\pm}_m(M) = \{D \in D^{\pm}(M) \mid \forall D' \in D^{\pm}(M) : D' \subseteq D \Rightarrow D \subseteq D'\}.$$

We call a pair $D = (D_1, D_2) \subseteq (br(M), br(M))$ a *diagnosis candidate*, regardless of whether or not $D \in D^{\pm}(M)$ holds.

Example 3.3. *In our running example, we obtain*

$$D^{\pm}_m(M) = \{(\{r_1\}, \emptyset), (\{r_2\}, \emptyset), (\{r_4\}, \emptyset), (\emptyset, \{r_5\})\}.$$

Accordingly, deactivating r_1, or r_2, or r_4, or adding r_5 unconditionally, will result in a consistent MCS. Note that $(\{r_3\}, \emptyset)$ is no diagnosis since in this case the presence of r_4 ensures that $give\_strong$ is inferred. Since administering the strong antibiotics is not allowed for $kb_4 \cup \{need\_strong\}$, there is no acceptable belief set for C_4 for this diagnosis candidate.

In more detail, we find: diagnosis $(\{r_1\}, \emptyset)$ removes bridge rule r_1. This way we ignore the X-Ray finding and obtain the following equilibrium:

$$\begin{aligned} EQ_1 = (&\{allergy\_strong\_ab\}, \{blood\_marker, xray\_pneumonia\}, \\ &\{(d, m1) : has\_marker\}, \{give\_nothing\}). \end{aligned}$$

It represents that we do not treat the patient since both the disease ontology and, consequently, the expert system have no information about a disease being present.

Diagnosis $(\{r_2\}, \emptyset)$ removes the bridge rule r_2. This ignores the result about the presence of the blood marker and the following equilibrium is obtained:

$$\begin{aligned} EQ_2 = (&\{allergy\_strong\_ab\}, \{blood\_marker, xray\_pneumonia\}, \\ &\{d : Pneumonia, d : BacterialDisease\}, \{need\_ab, give\_weak\}). \end{aligned}$$

It represents that the patient is given a wrong medication.

Diagnosis $(\{r_4\}, \emptyset)$ *removes bridge rule* r_4*. This ignores the information that treating the illness requires the strong antibiotics. We obtain the following equilibrium:*

$$\begin{aligned} EQ_3 = (&\{\mathit{allergy\_strong\_ab}\}, \{\mathit{blood\_marker}, \mathit{xray\_pneumonia}\},\\ &\{(d, \mathit{m1}){:}\mathit{has\_marker}, d{:}\mathit{Pneumonia}, d{:}\mathit{BacterialDisease},\\ &\ d{:}\mathit{AtypPneumonia}\}, \{\mathit{need\_ab}, \mathit{give\_weak}\}). \end{aligned}$$

Similarly to the previous diagnosis, it represents that the patient is treated wrongly.

Diagnosis $(\emptyset, \{r_5\})$ *adds an unconditional copy of bridge rule* r_5*, which forces strong antibiotics to be allowed as a treatment. The modified system has the following equilibrium:*

$$\begin{aligned} EQ_4 = (&\{\mathit{allergy\_strong\_ab}\}, \{\mathit{blood\_marker}, \mathit{xray\_pneumonia}\},\\ &\{(d, \mathit{m1}){:}\mathit{has\_marker}, d{:}\mathit{Pneumonia}, d{:}\mathit{BacterialDisease},\\ &\ d{:}\mathit{AtypPneumonia}\}, \{\mathit{need\_ab}, \mathit{need\_strong}, \mathit{allow\_strong\_ab},\\ &\ \mathit{give\_strong}\}). \end{aligned}$$

Any or none of the above possibilities might be the right choice: such decisions ought to be taken by a domain specialist (e.g., a doctor) and cannot be done automatically.

Preference on diagnoses can be defined in general, relying on some notion of plausibility (see e.g., for abduction [36]). Adding preferences to select most preferred diagnoses is investigated in detail in Chapter 4.

For talking about the MCS resulting from the application of a diagnosis (or diagnosis candidate) $(D_1, D_2) \subseteq (br(M), br(M))$, we also write $M[D_1, D_2]$ to denote the MCS $M[br(M) \setminus D_1 \cup cf(D_2)]$.

Explanations

Knowing all possibilities to remove inconsistency from an MCS is important, identifying the reasons of inconsistency to help a domain specialist understand the causes of inconsistency is equally important. In the spirit of abductive reasoning, we also propose an entailment-based notion of explaining inconsistency. An *inconsistency explanation* (in short, an *explanation*) is a pair of sets of bridge rules, whose presence or absence entails a relevant inconsistency in the given MCS.

Definition 3.3. *Given an MCS* M*, an* inconsistency explanation *of* M *is a pair* (E_1, E_2) *of sets* $E_1, E_2 \subseteq br(M)$ *of bridge rules, such that for all* (R_1, R_2) *where* $E_1 \subseteq R_1 \subseteq br(M)$ *and* $R_2 \subseteq br(M) \setminus E_2$*, it holds that* $M[R_1 \cup cf(R_2)] \models \bot$*. By* $E^{\pm}(M)$ *we denote the set of all inconsistency explanations of* M*, and by* $E^{\pm}_m(M)$ *the set of all pointwise subset-minimal ones.*

The intuition about E_1 is as follows: bridge rules in E_1 are crucial to create an inconsistency in M (i.e., $M[E_1] \models \bot$), and this inconsistency is relevant for M in the sense that adding any other bridge rules from $br(M)$ to $M[E_1]$ never yields a consistent system.

This condition of relevancy is necessary for nonmonotonic reasoning systems; for example the logic program $P = \{a \leftarrow \mathit{not}\, a.\}$ is inconsistent under the answer-set semantics, but its superset $P' = \{a \leftarrow \mathit{not}\, a.\ \ a.\}$ is consistent. The inconsistency of P does not matter for P'. In terms

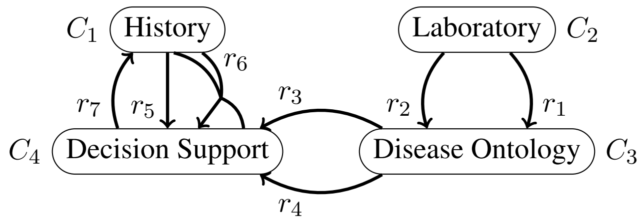


Figure 3.2: Modified medical MCS with inconsistent loop (cf. Example 3.5).

of MCS, a subset of all bridge rules may create an inconsistency in M, but this inconsistency is irrelevant, if it does not occur when more or all bridge rules are present.

The intuition about E_2 regards inconsistency wrt. the application of bridge rules: $M[E_1]$ cannot be made consistent unless at least one bridge rule from E_2 fires.

In summary, bridge rules E_1 create a relevant inconsistency, and at least one bridge rule in E_2 must be applied in unconditional form to repair that inconsistency.

Example 3.4. *In our running example, we have one minimal inconsistency explanation, namely* $(\{r_1, r_2, r_4\}, \{r_5\})$*. To trigger the only possible inconsistency, which is in* C_4*, we need to import* $need\_strong$ *(using* r_4*) and we must not import* $allow\_strong\_ab$ *(using* r_5*). Furthermore,* r_4 *can only fire if* C_3 *accepts* $d : AtypPneumonia$*, which is only possible if* r_1 *and* r_2 *fire. Therefore,* r_1*,* r_2*, and* r_4 must *be present to get inconsistency, and the head of* r_5 must not *be present.*

From Definition 3.3 the following property follows immediately.

Proposition 3.1. *Given an explanation* E *of an MCS* M*, each* E' *such that* $E \subseteq E' \subseteq (br(M), br(M))$ *is also an explanation.*

The following examples each illustrate some aspect of minimal explanations.

Example 3.5. *Consider a modification of our running medical example, where further bridge rules are added for the administration of anti-allergenics. Bridge rule* r_6 *encodes that an allergy blocking (anti-allergenics) medication is given if the strong antibiotics is needed, the patient is allergic to it, and nothing was done to block the allergic reaction;* r_7 *encodes that the patient database is informed if an anti-allergenics is applied:*

$$r_6\colon (4 : give\_antiallergenic.) \leftarrow (4 : need\_strong), (1 : allergy\_strong\_ab.), \mathbf{not}\ (1 : allergy\_blocked).$$
$$r_7\colon (1 : allergy\_blocked.) \leftarrow (4 : give\_antiallergenic).$$

The resulting system is depicted in Figure 3.2; it has two minimal inconsistency explanations: the previous explanation $(\{r_1, r_2, r_4\}, \{r_5\})$*, and the new* $(\{r_1, r_2, r_4, r_6, r_7\}, \{r_6, r_7\})$*. The latter shows the typical effect of a cycle with an odd number of negations through* r_6 *and* r_7*: both rules of the cycle* r_6 *and* r_7 *are present in both components of the minimal explanation. Intuitively, all rules of the cycle are necessary to cause the inconsistency while founding the cycle anywhere prevents the inconsistency. Minimal diagnoses of this MCS are* $(\{r_1\}, \emptyset)$*,* $(\{r_2\}, \emptyset)$*,* $(\{r_4\}, \emptyset)$*,* $(\{r_6\}, \{r_5\})$*,* $(\{r_7\}, \{r_5\})$*,* $(\emptyset, \{r_5, r_6\})$*, and* $(\emptyset, \{r_5, r_7\})$*.*

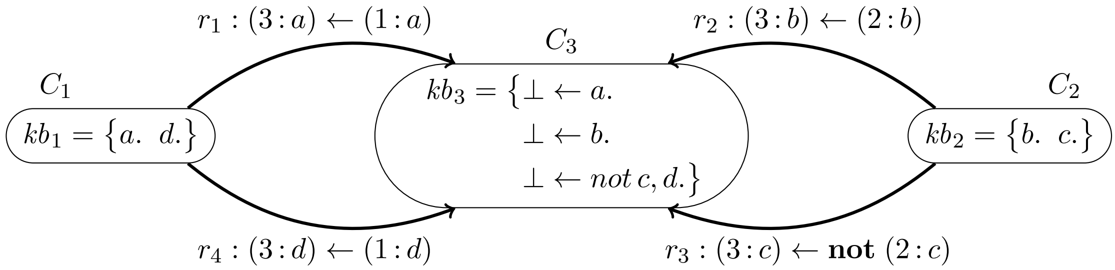


Figure 3.3: An MCS with three reasons for inconsistency.

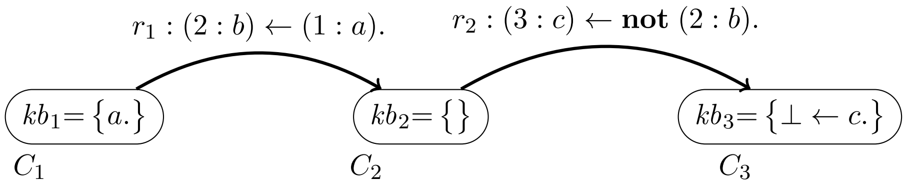


Figure 3.4: An MCS with an irrelevant inconsistency, ignored by the notion of explanation.

The following examples all use the logic L^{asp}_{Σ} for answer-set programs where knowledge bases have the common signature $\Sigma = \{a, b, c, d\}$.

Example 3.6. *Explanations separate independent reasons for inconsistency, as the MCS $M = (C_1, C_2, C_3)$ depicted in Figure 3.3 shows. Intuitively, there are three possibilities for M to become inconsistent: by r_1 which triggers the constraint $\bot \leftarrow a.$ in C_3; by r_2 which triggers $\bot \leftarrow b.$; and by r_3 not being applicable, which together with r_4 triggers $\bot \leftarrow not\, c, d$. Therefore, there are three minimal explanations, namely $(\{r_1\}, \emptyset)$, $(\{r_2\}, \emptyset)$, and $(\{r_4\}, \{r_3\})$. The latter indicates that r_3 must become applicable to remove the inconsistency.*

The use of subset-minimality to single out preferred solutions also suggest that cardinality-minimality could be an option. Example 3.6, however, shows that cardinality-minimal explanations cannot identify all sources of inconsistency, since there are three $\subseteq$-minimal explanations, but only two cardinality-minimal ones. Additionally, the set of cardinality-minimal explanations does not point out all bridge rules that must be modified to obtain a consistent system.

Example 3.7. *Consider the MCS $M = (C_1, C_2, C_3)$ given in Figure 3.4. The MCS $M' = M[\{r_2\}]$ which only contains bridge rule r_2 is inconsistent, since r_2 is applicable in M' and thus the constraint in kb_3 is violated. The MCS $M[\{r_1, r_2\}]$, however is not inconsistent due to r_1 causing r_2 to be not applicable. Hence, this inconsistency yields no explanation and $E^{\pm}_{m}(M) = \emptyset$; this agrees with the fact that M is consistent since $S = (\{a\}, \{b\}, \emptyset)$ is an equilibrium.*

Example 3.8. *The MCS $M = (C_1, C_2)$ depicted in Figure 3.5 is inconsistent, since there is a cycle with an odd number of negations through r_1 and r_2. The single explanation for M is*

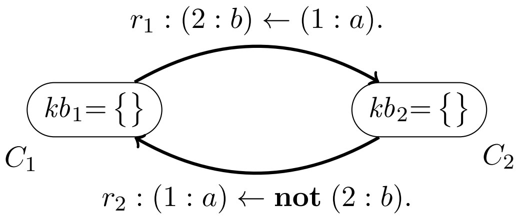


Figure 3.5: An MCS with inconsistency caused by a cycle.

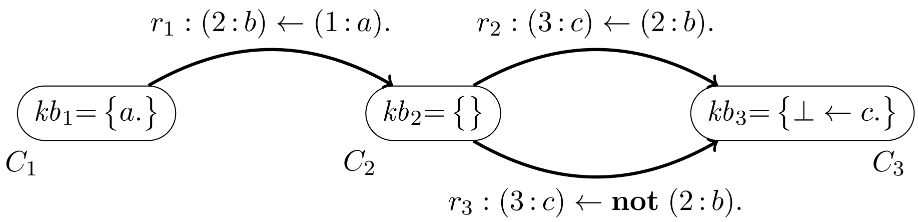


Figure 3.6: An MCS where mutually exclusive bridge rules r_2 and r_3 are in the same explanation.

$(\{r_1, r_2\}, \{r_1, r_2\})$, *intuitively because the cycle can either be broken by removing one of its rules, or it can be founded by making one rule unconditional.*

Example 3.9. *The MCS $M = (C_1, C_2, C_3)$ depicted in Figure 3.6 shows that mutually exclusive bridge rules, here r_2 and r_3, may be part of the same explanation $(\{r_2, r_3\}, \emptyset) \in E^{\pm}_m(M)$. Intuitively, the presence of r_3 causes the violation of the constraint in C_3 for $M[\{r_2, r_3\}]$ while r_2 causes the violation of the same constraint for $M[br(M)]$. Note that $(\{r_3\}, \emptyset) \notin E^{\pm}_m(M)$ since $M[\{r_3, r_1\}]$ is consistent, hence the inconsistency caused by r_3 alone is irrelevant. Besides that, there is another minimal explanation for M, namely $(\{r_1, r_2\}, \emptyset)$.*

Deletion-Diagnosis / Deletion-Explanation

For domains where removal of bridge rules is preferred to unconditional addition of rules, we specialise $D^{\pm}$ to obtain diagnoses of the form $(D_1, \emptyset)$. As for $D^{\pm}$, subset-minimal diagnoses are preferred.

Definition 3.4. *Given an MCS M, a deletion-diagnosis of M is a set $D \subseteq br(M)$ such that $M[br(M) \setminus D] \not\models \bot$. The set of all deletion-diagnoses is denoted by $D^-(M)$, the set of $\subseteq$-minimal deletion-diagnoses is denoted by $D^-_m(M)$.*

Example 3.10. *In Example 3.1, $D^-_m(M) = \{\{r_1\}, \{r_2\}, \{r_4\}\}$.*

Specialising inconsistency explanations to the first component, i.e., disregarding that rules may be added unconditionally, all explanations are of the form $(E_1, br(M))$.

Definition 3.5. *Given an MCS M, a deletion-explanation of M is a set $E \subseteq br(M)$ such that each R, where $E \subseteq R \subseteq br(M)$, satisfies $M[R] \models \bot$. The set of all such ($\subseteq$-minimal) explanations is denoted by $E^+(M)$, and the set of $\subseteq$-minimal ones by $E_m^+(M)$.*

Example 3.11. *The only, and thus minimal, deletion-explanation in Example 3.1 is $\{r_1, r_2, r_4\}$.*

Refined Notions of Diagnosis and Explanation

Refined Diagnosis

One can generalise Definition 3.1 to refined changes of bridge rules, such that bridge rules in the second component of a diagnosis become applicable by only removing some body literals instead of all. Hence, those bridge rules whose head formula restores consistency are not made completely condition-free, but only a minimal set of the conditions in the body are removed. Note that in the remainder of this section, we identify the body of a bridge rule with the set of its literals. This means that a bridge rule where a literal occurs more than once is identified with the bridge rule where each literal occurs exactly once. This is no real restriction since the two bridge rules behave exactly the same anyway.

Let $br_{\mathit{ref}}(M)$ denote the set of bridge rules of M where some body literals have been removed, i.e., $br_{\mathit{ref}}(M) = \{\mathit{head}(r) \leftarrow B. \mid B \subseteq \mathit{body}(r)\}$. A function $\mathit{fg} : br(M) \rightarrow br_{\mathit{ref}}(M)$ is called a *body-reduction function*; it maps bridge rules to rules where some or no body literals are removed. In the following, we identify $\mathit{fg} : br(M) \rightarrow br_{\mathit{ref}}(M)$ with the corresponding function $\mathit{fg} : 2^{br(M)} \rightarrow 2^{br_{\mathit{ref}}(M)}$ on sets of bridge rules, i.e., for a set $R \subseteq br(M)$ we have $\mathit{fg}(R) = \{\mathit{fg}(r) \mid r \in R\}$.

Definition 3.6. *A* refined diagnosis *is a triple (D_1, D_2, fg) consisting of sets of bridge rules $D_1, D_2 \subseteq br(M)$ and a body-reduction function $\mathit{fg} : br(M) \rightarrow br_{\mathit{ref}}(M)$, such that the resulting MCS is consistent, i.e., $M[br(M) \setminus D_1 \cup \mathit{fg}(D_2)] \not\models \bot$. The set of all refined diagnoses is denoted by $D^{\pm,r}(M)$.*

Again, by Occam's razor, we seek refined diagnoses that assume a minimal amount of modifications necessary. To that end, we seek to change a minimal set of bridge rules and within this set, we seek a minimal change of bridge rule bodies, where more preservation of body literals is considered to be preferable. Formally, let fg and fg' be two body-reduction functions on $br(M)$, then fg is more conservative than fg', written $\mathit{fg} \leq \mathit{fg}'$, iff for every $r \in br(M)$ holds $\mathit{body}(\mathit{fg}(r)) \supseteq \mathit{body}(\mathit{fg}'(r))$. Furthermore, we write $\mathit{fg} < \mathit{fg}'$ iff $\mathit{fg} \leq \mathit{fg}'$ and $\mathit{fg} \neq \mathit{fg}'$.

A refined diagnosis $(D_1, D_2, \mathit{fg}) \in D^{\pm,r}(M)$ is called *minimal*, iff for every $(D'_1, D'_2, \mathit{fg}') \in D^{\pm,r}(M)$ such that $D'_1 \subseteq D_1$ and $D'_2 \subseteq D_2$ it holds that $D_1 = D'_1, D_2 = D'_2$, and $\mathit{fg}' \not< \mathit{fg}$. The set of all minimal refined diagnoses is denoted by $D_m^{\pm,r}(M)$. Observe that the conservation of the body-reduction functions only comes into play if the sets of bridge rules are subset-minimal.

Example 3.12. *Consider a slight modification of Example 3.1 where data from the patient history is only imported in the expert system if the patient is currently under treatment in the hospital. So bridge rule r_5 is changed to*

$$r_5\colon \quad (4 : \mathit{allow\_strong\_ab}) \leftarrow (1 : \mathit{under\_treatment}), \mathbf{not}\ (1 : \mathit{allergy\_strong\_ab}).$$

and our patient is in the hospital, i.e., $kb_1 = \{allergy\_strong\_ab, under\_treatment\}$*.*

Let $fg(r_5) = (4: allow\_strong\_ab) \leftarrow (1: under\_treatment)$ *and* $fg(r) = r$ *for all* $r \in br(M)$ *with* $r \neq r_5$*. We denote the modified MCS by* M'*. Then there exists a refined diagnosis which removes from* r_5 *only the literal* $\mathbf{not}\ (4: allow\_strong\_ab)$*, i.e.,* $(\emptyset, \{r_5\}, fg) \in D_m^{\pm,r}(M')$*, because* $fg(r_5)$ *modifies* r_5 *such that the strong antibiotic is allowed if the patient merely is under treatment. Compared to removing the whole body, the refined diagnosis more precisely indicates the beliefs whose presence would make* M' *consistent.*

Note that one could also think of refining rules in D_1, i.e., ensuring that a rule in D_1 is not applicable by adding additional literals to its body. But as there are no hints to which literals should be added, such a process would result in a large and arbitrary search space. For example, adding $\mathbf{not}\ (1: allergy\_strong\_ab)$ to r_2 would result in:

$$r_2': \quad (3:(d, m1) :has\_marker) \leftarrow (2: blood\_marker), \mathbf{not}\ (1: allergy\_strong\_ab).$$

This would make the MCS of Example 3.1 consistent. Given no information on which literals make sense for adding to a bridge rule, any belief of any context may be added. This means that for every ordinary diagnosis $(D_1, D_2) \in D^{\pm}(M)$ one simply needs to find a belief p of some context C_k which is absent in all acceptable belief sets of C_k. Adding the literal $(k:p)$ to every bridge rule $r \in D_1$ then prevents its applicapility. If there are multiple such beliefs, every combination of them results in such a diagnosis where literals are added to bridge rules. In fact, for m such beliefs there are $m^{|D_1|}$ possible diagnoses resulting just from one context and one diagnosis. Given no further information on which literals make sense to add, there seems to be no way of singling out good literals to add among the exponentially many candidates. Therefore, the addition of literals to bridge rules is not considered in this thesis.

But even in the case of minimal refined diagnoses, there is little information gain: every minimal diagnosis $(D_1, D_2) \in D_m^{\pm}(M)$, together with a witnessing equilibrium S_w of (D_1, D_2), can be refined to a minimal refined diagnosis (D_1, D_2, fg) using the following *refine* function. Let $\mathcal{S}$ be the set of belief states of the MCS M; then $refine(D_2, S_w) : 2^{br(M)} \times \mathcal{S} \rightarrow (br(M) \rightarrow br_{ref}(M))$ is given by $(D_2, S_w) \mapsto fg$ where fg is the body-reduction function defined as follows:

$$fg(r) = \begin{cases} head(r) \leftarrow B. & \text{if } r \in D_2, B \subseteq body(r), S_w \mathrel{|\!\!\leadsto} head(r) \leftarrow B., \\ & \text{and for no } B \subset B' \subseteq body(r) \text{ holds } S \mathrel{|\!\!\leadsto} head(r) \leftarrow B'.; \\ r & \text{otherwise.} \end{cases}$$

Observe that a refined diagnosis (D_1, D_2, fg) obtained in such way also admits the equilibrium S_w, as all rules of $fg(D_2)$ are applicable in S_w and therefore all head beliefs of D_2 are added to the respective contexts, which results in the same knowledge bases as for $cf(D_2)$.

Proposition 3.2. *A triple* (D_1, D_2, fg) *is a minimal refined diagnosis of* M *iff there exists a minimal diagnosis* $(D_1, D_2) \in D_m^{\pm}(M)$ *and a witnessing equilibrium* S_w *such that* $refine(D_2, S_w) = fg$ *and no witnessing equilibrium* S'_w *exists where* $refine(D_2, S'_w) = fg'$ *and* $fg' < fg$*.*

Proof. ($\Rightarrow$) Let $D_r = (D_1, D_2, \mathit{fg}) \in D_m^{\pm,r}(M)$, we show that $(D_1, D_2) \in D_m^{\pm}(M)$ and we first show that $(D_1, D_2) \in D^{\pm}(M)$. Since D_r is a refined diagnosis, it holds that $M[br(M) \setminus D_1 \cup \mathit{fg}(D_2)] \not\models \bot$. Let S_w be a witnessing equilibrium of $M[br(M) \setminus D_1 \cup \mathit{fg}(D_2)]$, then it holds for every $r \in D_2$ that $S_w \models\!\!\leadsto \mathit{fg}(r)$ since D_r is minimal. Therefore, S_w is an equilibrium of $M[br(M) \setminus D_1 \cup \mathit{cf}(D_2)]$, hence $(D_1, D_2) \in D^{\pm}(M)$. Since D_r is minimal, it holds for no $r \in D_2$ that $S_w \models\!\!\leadsto \mathit{head}(r) \leftarrow \mathit{body}(\mathit{fg}(r)) \cup B$. where $\mathit{body}(\mathit{fg}(r)) \subset B \subseteq \mathit{body}(r)$, hence $\mathit{refine}(D_2, S_w) = \mathit{fg}$.

It remains to show that $(D_1, D_2) \in D_m^{\pm}(M)$. Assume for contradiction that there exists $(D'_1, D'_2) \subset (D_1, D_2)$ such that $(D'_1, D'_2) \in D_m^{\pm}(M)$. Let S'_w be a witnessing equilibrium of (D'_1, D'_2) and $\mathit{fg}' = \mathit{refine}(D'_2, S'_w)$. Then it holds that $(D'_1, D'_2, \mathit{fg}') \in D^{\pm,r}(M)$ since S'_w is a witnessing equilibrium of $M[br(M) \setminus D'_1 \cup \mathit{fg}'(D'_2)]$. Since $(D'_1, D'_2, \mathit{fg}') \in D^{\pm,r}(M)$ and $(D'_1, D'_2) \subset (D_1, D_2)$ it holds that D_r is not a minimal refined diagnosis, which is a contradiction. Therefore, no such (D'_1, D'_2) exists and $(D_1, D_2) \in D_m^{\pm}(M)$.

($\Leftarrow$) Let $D = (D_1, D_2) \in D_m^{\pm}(M)$, let S_w be a witnessing equilibrium of D, and let $\mathit{refine}(D_2, S_w) = \mathit{fg}$. Furthermore, assume that no witnessing equilibrium S'_w exists with $\mathit{refine}(D_2, S'_w) = \mathit{fg}'$ and $\mathit{fg}' < \mathit{fg}$. We show that (D_1, D_2, fg) is a minimal refined diagnosis of M. By definition of *refine* it holds for every $r \in D_2$ that $S_w \models\!\!\leadsto \mathit{fg}(r)$. Therefore, S_w is an equilibrium of $M[br(M) \setminus D_1 \cup \mathit{fg}(D_2)]$ and $(D_1, D_2, \mathit{fg}) \in D^{\pm,r}(M)$.

Towards contradiction assume (D_1, D_2, fg) is not minimal, then there exists $(D'_1, D'_2, \mathit{fg}') \in D_m^{\pm,r}(M)$ such that (1): $(D'_1, D'_2) \subset (D_1, D_2)$ or (2): $D_1 = D'_1$, $D_2 = D'_2$, and $\mathit{fg}' < \mathit{fg}$. Case (1): there exists a witnessing equilibrium S'_w of $M[br(M) \setminus D'_1 \cup \mathit{fg}'(D'_2)]$. Therefore S'_w is a witnessing equilibrium of $M[br(M) \setminus D'_1 \cup \mathit{cf}(\{r \in D'_2 \mid S'_w \models\!\!\leadsto \mathit{fg}'(r)\})]$, i.e., $D'' = (D'_1, \{r \in D'_2 \mid S'_w \models\!\!\leadsto \mathit{fg}'(r)\}) \in D^{\pm}(M)$. Since $D'' \subset D$ this is a contradiction to $D \in D_m^{\pm}(M)$. In case (2) holds $\mathit{fg}' < \mathit{fg}$ and there exists a witnessing equilibrium S'_w of $M[br(M) \setminus D_1 \cup \mathit{fg}'(D_2)]$. Since $(D'_1, D'_2, \mathit{fg}') \in D_m^{\pm,r}(M)$ and $D'_1 = D_1, D'_2 = D_2$, it holds that S'_w also is an equilibrium of $M[br(M) \setminus D_1 \cup \mathit{cf}(D_2)]$ and $\mathit{refine}(D_2, S'_w) = \mathit{fg}'$. Then, $\mathit{fg}' < \mathit{fg}$ directly contradicts our assumption that no such S'_w and fg' exist. Since all cases are contradicting, it must hold that (D_1, D_2, fg) is a minimal refined diagnosis. □

Example 3.13. *Recall Example 3.12. The set of minimal diagnoses is the same as for the running example, in particular $(\emptyset, \{r_5\})$ is a minimal diagnosis. The refinement of this diagnosis can be computed using its (only) witnessing equilibrium*

$$\begin{aligned} S_v = (&\{\mathit{allergy\_strong\_ab}, \mathit{under\_treatment}\},\\ &\{\mathit{blood\_marker}, \mathit{xray\_pneumonia}\},\\ &\{d : \mathit{Pneumonia}, (d, m1) : \mathit{has\_marker}, d : \mathit{AtypPneumonia}\},\\ &\{\mathit{need\_ab}, \mathit{need\_strong}, \mathit{allow\_strong\_ab}, \mathit{give\_strong}\}), \end{aligned}$$

where only the negated literal of r_5 is deleted, this is sufficient to make the rule applicable under S_w, i.e., $\mathit{fg}(r_5) = (4 : \mathit{allow\_strong\_ab}) \leftarrow (1 : \mathit{under\_treatment})$. and it holds that $(\emptyset, \{r_5\}, \mathit{fg}) \in D_m^{\pm,r}(M)$.

Thus, each minimal refined diagnosis is essentially a minimal diagnosis in recoded form. Minimal refined diagnoses can be simulated by minimal diagnoses and do not convey any more information than minimal diagnoses.

Refined Explanations

Similar to diagnoses, it is possible to consider refined modifications of rules (rather than $cf(R_2)$) in Definition 3.3.

Definition 3.7. *A* refined explanation *is a triple* (E_1, E_2, fg) *consisting of sets of bridge rules* $E_1, E_2 \subseteq br(M)$ *and a body-reduction function fg, such that* $M[R_1 \cup fg'(R_2)] \models \bot$ *holds, for every* $E_1 \subseteq R_1 \subseteq br(M)$*,* $R_2 \subseteq br(M)$*, and every body-reduction function* fg' *where* $r \in E_2$ *implies* $body(fg(r)) \subseteq body(fg'(r))$*.*

Here, we shift the "prevention of inconsistency" expressed by E_2 in Definition 3.3 to the body-reduction fg: we do not add unconditional bridge rules, i.e., from $br(M) \setminus E_2$, but rather consider all body-reductions fg' for which it holds that bridge rules in E_2 retain all literals indicated by fg.

Example 3.14. *Reconsider the modified MCS of Example 3.12. A refined explanation is* (E_1, E_2, fg) *where* $E_1 = \{r_1, r_2, r_4\}$*,* $E_2 = \{r_5\}$*, and*

$$fg(r_5) = (4 : allow\_strong\_ab) \leftarrow \mathbf{not}\ (1 : allergy\_strong\_ab).$$

This indicates that every literal except $\mathbf{not}\ (1 : allergy\_strong\_ab)$ *in the body of* r_5 *might be removed and the inconsistency is still present in the modified MCS. Phrased differently:* $\mathbf{not}\ (1 : allergy\_strong\_ab)$ *must be deleted from the body of* r_5 *to prevent the inconsistency that is caused by* E_1*.*

The notion of a refined explanation is a generalisation of the notion of explanation and there is a 1-to-1 correspondence between them.

Proposition 3.3. *For an inconsistent MCS M, it holds that* $(E_1, E_2) \in E^{\pm}(M)$ *iff there exists a body-reduction function fg such that* (E_1, E_2, fg) *is a refined explanation.*

Proof. ($\Rightarrow$) Let $(E_1, E_2) \in E^{\pm}(M)$, pick fg such that for every $r \in E_2$ holds $fg(r) = r$ and for $r \in br(M) \setminus E_2$ holds $fg(r) = head(r) \leftarrow .$, i.e. $\{fg(r) \mid r \in br(M) \setminus E_2\} = cf(br(M) \setminus E_2)$. Let R_1, R_2 be sets of bridge rules such that $E_1 \subseteq R_1 \subseteq br(M)$, $R_2 \subseteq br(M)$ and let fg' be any body-reduction function such that $body(fg(r)) \subseteq body(fg'(r))$ holds if $r \in E_2$. We have to show that $M[R_1 \cup fg'(R_2)] \models \bot$ holds. Note that by construction of fg and the fact that fg' is a body-reduction function with $body(fg(r)) \subseteq body(fg'(r))$ for all $r \in E_2$, it holds that $fg'(r) = r$ for all $r \in E_2$.

Let $S = (S_1, \ldots, S_n)$ be an arbitrary belief state of M (hence S is also a belief state of $M[R_1 \cup fg'(R_2)]$), let $A \subseteq R_2$ be those bridge rules of R_2 that are applicable in S, i.e., $A = \{r \in R_2 \mid S \models fg'(r)\}$, and let $B = \{r \in E_2 \mid r \in A\}$ be those applicable bridge rules that also occur in E_2.

Now consider $M[R_1 \cup B \cup cf(A \setminus B)]$ and observe that $E_1 \subseteq (R_1 \cup B) \subseteq br(M)$ holds as well as it holds that $(A \setminus B) \subseteq br(M) \setminus E_2$. From $(E_1, E_2) \in E^{\pm}(M)$ it thus follows that $M[R_1 \cup B \cup cf(A \setminus B)] \models \bot$, i.e., there exists some $1 \leq k \leq n$ such that $S_k \notin \mathbf{ACC}_k(kb_k \cup app(br_k(M[R_1 \cup B \cup cf(A \setminus B)]), S))$.

We next show that for all $1 \leq i \leq n$ it holds that:

$$\{\varphi(r) \mid r \in br_i(M[R_1 \cup fg'(R_2)]), S \models\!\leadsto r\} = \{\varphi(r) \mid r \in br_i(M[R_1 \cup B \cup cf(A \setminus B)]), S \models\!\leadsto r\}.$$

In the following, let $1 \leq i \leq n$ be arbitrary and let $F_1 = \{\varphi(r) \mid r \in br_i(M[fg'(R_2)]), S \models\!\leadsto r\}$ and $F_2 = \{\varphi(r) \mid r \in br_i(M[B \cup cf(A \setminus B)]), S \models\!\leadsto r\}$. Clearly, to prove the above equality, it suffices to prove that $F_1 = F_2$.

($\subseteq$) Let $r \in br_i(M[fg'(R_2)])$ and $S \models\!\leadsto r$ hold, then it holds that $r \in fg'(R_2)$, i.e., there exists $r' \in br(M)$ such that $r = fg'(r')$. If $r' \in E_2$ holds, then $fg'(r') = r$. By construction, it holds that $r' \in B$ and since $S \models\!\leadsto r$, it holds that $S \models\!\leadsto r'$. Since $\varphi(r) = \varphi(r')$, it therefore holds that $\varphi(r) \in F_2$. If $r' \notin E_2$ holds, then $r' \in A \setminus B$ there exists $r'' \in cf(A \setminus B)$ such that $r'' = cf(r)$, i.e., $S \models\!\leadsto r''$ and by $\varphi(r'') = \varphi(r') = \varphi(r)$ it holds that $\varphi(r) \in F_2$.

($\supseteq$) Let $r \in br_i(M[B \cup cf(A \setminus B)])$ hold and $S \models\!\leadsto r$ hold. If $r \in B$ holds, then $r \in E_2$ and $fg'(r) = r$, hence $S \models\!\leadsto fg'(r)$. Since $\varphi\left(fg'(r)\right) = \varphi(r)$ it then follows that $\varphi(r) \in F_1$. If $r \in cf(A \setminus B)$ holds, then there exists $r' \in A \setminus B$ such that $r' \notin E_2$ and by the construction of A it holds that $S \models\!\leadsto fg'(r')$. Since $\varphi(r') = \varphi(r)$ it then follows that $\varphi(r) \in F_1$. In summary, for every $\varphi(r) \in F_2$ holds $\varphi(r) \in F_1$.

Therefore $F_1 = F_2$ holds for all $1 \leq i \leq n$ since i was chosen arbitrarily. Especially, for $i = k$ it holds that $S_k \notin \mathbf{ACC}_k(kb_k \cup app(br_k(M[R_1 \cup fg'(R_2)]), S))$, i.e., S is no equilibrium of that MCS. Since S also was chosen arbitrarily, it holds for all belief states that there exists some such k, and consequently it holds that $M[R_1 \cup fg'(R_2)] \models \bot$. Furthermore, since R_1, R_2 and fg' are also arbitrary with $E_1 \subseteq R_1 \subseteq br(M)$, $R_2 \subseteq br(M)$, and $body(fg'(r)) \supseteq body(fg(r))$ holds for all $r \in E_2$, it holds for all such R_1, R_2 and fg' that $M[R_1 \cup fg'(R_2)] \models \bot$. Thus, (E_1, E_2, fg) is a refined explanation.

($\Leftarrow$) Let (E_1, E_2, fg) be a refined explanation, i.e., $M[R_1 \cup fg'(R_2)] \models \bot$ for every $E_1 \subseteq R_1 \subseteq br(M)$, $R_2 \subseteq br(M)$, and body-reduction function fg' with $body(fg(r)) \subseteq body(fg'(r))$ for every $r \in E_2$. Consider the body-reduction function fg' such that for all $r \in br(M) \setminus E_2$ it holds that $fg'(r) = head(r) \leftarrow .$ and $fg'(r) = r$ for every $r \in E_2$, i.e., $fg'(R'_2) = cf(R'_2)$ for every $R'_2 \subseteq br(M) \setminus E_2$. Observe that $body(fg(r)) \subseteq body(fg'(r))$ holds for every $r \in E_2$, therefore $M[R_1 \cup fg'(R_2)] \models \bot$ for every $E_1 \subseteq R_1 \subseteq br(M)$ and $R_2 \subseteq br(M)$ by Definition 3.7. Hence, $M[R_1 \cup cf(R'_2)] \models \bot$ for every $R'_2 \subseteq br(M) \setminus E_2$ and thus $(E_1, E_2) \in E^{\pm}(M)$. $\square$

In contrast to diagnoses, an explanation does not admit a witnessing equilibrium. Therefore, we cannot infer from an explanation whether the addition of a reduced version of a bridge rule would yield consistency.

However, this can be achieved considering a transformed MCS: Consider $M = (C_1, \ldots, C_n)$, then $M^r = (C_1, \ldots, C_n, C_\alpha)$ is the transformed MCS where C_α is a context whose acceptable belief states contain exactly those formulas added to it via bridge rules, e.g., C_α uses the logic L^{asp} and an empty knowledge base $kb_\alpha = \emptyset$. Furthermore, the bridge rules of $br(M^r)$ are obtained from $br(M)$ in such a way that every bridge rule $r \in br(M)$ of form (2.1) in Definition 2.2 is split into a core rule ($r^{(0)}$) and a supplementary rule for each body atom ($r^{(1)}, \ldots, r^{(m)}$). The set

$tr(r)$ of transformed bridge rules corresponding to the bridge rule $r \in br(M)$ is then given by:

$$
\begin{aligned}
tr(r) = \{ \quad & r^{(0)}: && (k:s) \leftarrow (C_\alpha:p_1), \ldots, (C_\alpha:p_j), (C_\alpha:p_{j+1}), \ldots, (C_\alpha:p_m). \\
& r^{(1)}: && (C_\alpha:p_1) \leftarrow (c_1:p_1). \\
& && \ldots \\
& r^{(j)}: && (C_\alpha:p_j) \leftarrow (c_j:p_j). \\
& r^{(j+1)}: && (C_\alpha:p_{j+1}) \leftarrow \mathbf{not}\ (c_{j+1}:p_{j+1}). \\
& && \ldots \\
& r^{(m)}: && (C_\alpha:p_m) \leftarrow \mathbf{not}\ (c_m:p_m). \quad \}
\end{aligned}
$$

Finally, M^r contains for each bridge rule of M the corresponding transformed rules, i.e., $br(M^r) = \bigcup_{r \in br(M)} tr(r)$. Note that, for readability, this transformation assumes w.l.o.g. beliefs of different contexts to be disjoint. This can always be achieved by renaming the elements in the new context C_α.

For example, a bridge rule

$$(c_1:h) \leftarrow (c_2:a), \mathbf{not}\ (c_3:b).$$

of M is transformed to the following bridge rules of M^r:

$$
\begin{aligned}
(c_1:h) &\leftarrow (c_\alpha:a'), (c_\alpha:b'). \\
(c_\alpha:a') &\leftarrow (c_2:a). \\
(c_\alpha:b') &\leftarrow \mathbf{not}\ (c_3:b).
\end{aligned}
$$

An explanation $(E_1, E_2) \in E^\pm(M^r)$ then allows to construct a refined explanation (E_1, E_2^r, fg) for M as follows: For every $r \in br(M)$, it holds that $r \in E_2^r$ iff $tr(r) \cap E_2 \neq \emptyset$. Furthermore, for $r \in br(M)$, let $sup(r) = \{body(r') \mid r' \in tr(r) \wedge r' \neq r^{(0)}\}$, then fg is a body-reduction function on $br(M)$ such that $fg(r) = head(r) \leftarrow sup(r)$ if $r \in E_2$ and $fg(r) = (r)$ otherwise.

For example, if the supplementary rule $(c_\alpha:a') \leftarrow (c_2:a)$. is in E_2, then the removal of the corresponding literal, here $(c_2:a)$, from the original bridge rule in M contributes to preventing the inconsistency in M that is caused by E_1. Similarly as for refined diagnoses, Proposition 3.3 together with the above transformation yielding M^r allows to simulate refined explanations by ordinary explanations.

3.3 Properties

In this section we first show that, to some extent, diagnoses can be converted to explanations and vice versa; specifically, minimal diagnoses and minimal explanations point out the same bridge rules, a property we call duality. We then prove a useful non-intersection property of minimal diagnoses, and show how modularity of an MCS (defined in the spirit of splitting sets of logic programs) is reflected in the structure of its diagnoses and explanations.

Converting between Diagnoses and Explanations

Chronologically, we first discovered the duality between minimal diagnoses and explanations, which is covered in the section after this. The author of this thesis first proved that duality holds and later Peter Schüller (cf. [117]) proved the more specific conversion results given below. Since this later proof is more elegant and it already implies duality, only the later proof is given here.

In the following, we show that it is possible to characterise explanations in terms of diagnoses, and vice versa minimal diagnoses in terms of minimal explanations. To this end, we generalise the notion of a hitting set [113] from sets to pairs of sets. Given a collection $\mathcal{C} = \{(A_1, B_1), \ldots, (A_n, B_n)\}$ of pairs of sets (A_i, B_i), $A_i, B_i \subseteq U$ over a set U, a *hitting set of* $\mathcal{C}$ is a pair of sets (X, Y), $X, Y \subseteq U$ such that for every pair $(A_i, B_i) \in \mathcal{C}$, (i) $A_i \cap X \neq \emptyset$ or (ii) $B_i \cap Y \neq \emptyset$. A hitting set (X, Y) of $\mathcal{C}$ is *minimal*, if no $(X', Y') \subset (X, Y)$ is a hitting set of $\mathcal{C}$.

We consider hitting sets over pairs of sets of bridge rules, and denote by $\mathit{HS}_M(\mathcal{C})$ (respectively, $\mathit{minHS}_M(\mathcal{C})$) the set of all (respectively, all minimal) hitting sets of $\mathcal{C}$ over $U = br(M)$. Note that in particular $\mathit{HS}_M(\emptyset) = \{(\emptyset, \emptyset)\}$, and $\mathit{HS}_M(\{(\emptyset, \emptyset)\}) = \emptyset$.

Theorem 3.1 (cf. [54]). *For every MCS M,*

(a) a pair (E_1, E_2) with $E_1, E_2 \subseteq br(M)$ is an inconsistency explanation of M iff $(E_1, E_2) \in \mathit{HS}_M(D^{\pm}(M))$, i.e., (E_1, E_2) is a hitting set of $D^{\pm}(M)$; and

(b) a pair (E_1, E_2) with $E_1, E_2 \subseteq br(M)$ is a minimal inconsistency explanation of M iff $(E_1, E_2) \in \mathit{minHS}_M(D^{\pm}(M))$, i.e., (E_1, E_2) is a minimal hitting set of $D^{\pm}(M)$.

Proof. In this proof, for variables E_i, D_i, and R_i with $i \in \{1, 2\}$, we assume that $E_i, D_i, R_i \subseteq br(M)$. Furthermore, we denote by $\overline{X}$ the complement of set X wrt. $br(M)$, i.e., $\overline{X} = br(M) \setminus X$.

((a)) Given a pair (E_1, E_2). For all diagnoses $(D_1, D_2) \in D^{\pm}(M)$, $D_1 \cap E_1$ or $D_2 \cap E_2$ or both are nonempty iff

for all (D_1, D_2) we have that
$$M[\overline{D_1} \cup \mathit{cf}(D_2)] \not\models \bot \text{ implies } D_1 \cap E_1 \neq \emptyset \text{ or } D_2 \cap E_2 \neq \emptyset$$

which (by reversing the implication and simplifying) is equivalent to

for all (D_1, D_2) we have that
$$(D_1 \cap E_1 = \emptyset \text{ and } D_2 \cap E_2 = \emptyset) \text{ implies } M[\overline{D_1} \cup \mathit{cf}(D_2)] \models \bot.$$

As $A \cap B = \emptyset$ with $A, B \subseteq br(M)$ is equivalent to $A \subseteq \overline{B}$ we next obtain

for all (D_1, D_2) we have that
$$(E_1 \subseteq \overline{D_1} \text{ and } D_2 \subseteq \overline{E_2}) \text{ implies } M[\overline{D_1} \cup \mathit{cf}(D_2)] \models \bot.$$

If we let $D_1 = \overline{R_1}$ and $D_2 = R_2$ this amounts to

for all (R_1, R_2) we have that
$$(E_1 \subseteq R_1 \text{ and } R_2 \subseteq \overline{E_2}) \text{ implies } M[R_1 \cup \mathit{cf}(R_2)] \models \bot. \quad (3.1)$$

This proves the result ((a)) as this last condition is the one of an explanation (E_1, E_2) in Definition 3.3. Note that, if $(\emptyset, \emptyset) \in D^{\pm}(M)$, then no explanation exists; this is intentional and corresponds to the definitions of diagnosis and explanation for consistent systems.

((b)) As $minHS_M(X)$ contains the $\subseteq$-minimal elements in $HS_M(X)$, and $E^{\pm}_m(M)$ contains the $\subseteq$-minimal elements in $E^{\pm}(M)$, ((b)) follows from ((a)). □

Clearly, a hitting set of a collection X is the same as a hitting set of the collection of the $\subseteq$-minimal elements in X; from Theorem 3.1. we therefore immediately obtain the following.

Corollary 3.1 (cf. [54]). *For every MCS M,*

(a) a pair (E_1, E_2) with $E_1, E_2 \subseteq br(M)$ is an inconsistency explanation of M iff $(E_1, E_2) \in HS_M(D^{\pm}_m(M))$; and

(b) a pair (E_1, E_2) with $E_1, E_2 \subseteq br(M)$ is a minimal inconsistency explanation of M iff $(E_1, E_2) \in minHS_M(D^{\pm}_m(M))$.

Proof. Let $min(X)$ be the set of $\subseteq$-minimal elements in a collection X of sets. Then for every $(A, B) \in X \setminus min(X)$ there is a pair $(A', B') \in min(X)$ with $(A', B') \subseteq (A, B)$. Given $HS_M(min(X))$, every pair $(A, B) \in X \setminus min(X)$ is hit by every pair $(C, D) \in HS_M(min(X))$. Therefore $HS_M(min(X)) = HS_M(X)$. Then ((a)) immediately follows from Theorem 3.1 ((a)), and ((b)) immediately follows from Theorem 3.1 ((b)). □

For the next result, we use the following generalisation of a well-known result for minimal hitting sets [10].

Lemma 3.1 (cf. [54]). *For every collection $X = \{X^1, \ldots, X^n\}$ of pairs $X^i = (X^i_1, X^i_2)$ of sets, $1 \leq i \leq n$, such that X is an anti-chain wrt. $\subseteq$, i.e., elements in X are pairwise incomparable ($X^i \subseteq X^j$ with $1 \leq i, j \leq n$ implies $X^i = X^j$) it holds that $minHS_M(minHS_M(X)) = X$.*

Proof. A collection of sets $C = \{C_1, \ldots, C_n\}$ over a universe, i.e., $C_i \subseteq U$, $1 \leq i \leq n$, can be seen as a *hypergraph* $\mathcal{H} = (U, C)$ with vertices U and hyperedges $C_i \in C$. If no hyperedge C_i is contained in any hyperedge C_j, $i \neq j$, it is called *simple*. A hitting set on C is called *transversal*, and the hypergraph (U, C') containing as hyperedges C' all minimal hitting sets of the hypergraph $\mathcal{H}$ is called *transversal hypergraph* $Tr(\mathcal{H})$.

We can map a collection $X = \{X^1, \ldots, X^n\}$ of pairs $X^i = (X^i_1, X^i_2)$ of sets, X^i_1,$X^i_2 \subseteq U$ bijectively to a collection $\mu(X) = \{\mu(X^1), \ldots, \mu(X^n)\}$ over $U \cup \{u' \mid u \in U\}$ where $\mu(X^i_1, X^i_2) = X^i_1 \cup \{u' \mid u \in X^i_2\}$. Then, (A, B) is a hitting set of X iff $\mu(A, B)$ is a hitting set of $\mu(X)$, and well-known results for transversal hypergraphs [10] carry over to minimal hitting sets over pairs.

In particular, given a simple hypergraph $\mathcal{H} = \mu(X)$, it holds that $Tr(Tr(\mu(X))) = \mu(X)$. This directly translates into the lemma, because $\mu(X)$ is a simple hypergraph due to incomparability (also called the anti-chain property) of X, and μ is bijective, therefore transversal hypergraphs can be mapped back to minimal hitting sets. □

Combined with Corollary 3.1 ((b)) we thus obtain.

Theorem 3.2 (cf. [54]). *A pair (D_1, D_2) with $D_1, D_2 \subseteq br(M)$ is a minimal diagnosis of M iff (D_1, D_2) is a minimal hitting set of $E^{\pm}_m(M)$, formally $D^{\pm}_m(M) = minHS_M(E^{\pm}_m(M))$.*

Proof. From Corollary 3.1 (b) we have that $E^{\pm}_m(M) = minHS_M(D^{\pm}_m(M))$. Applying $minHS_M$ on both sides of this formula and then using Lemma 3.1 yields $minHS_M(E^{\pm}_m(M)) = minHS_M(minHS_M(D^{\pm}_m(M))) = D^{\pm}_m(M)$. □

As for computation, Theorem 3.1 provides a way to compute the set $E^{\pm}(M)$ of explanations from the set $D^{\pm}(M)$ of diagnoses, while Theorem 3.2 allows us to compute the set $D^{\pm}_m(M)$ of minimal diagnoses from the set of minimal explanations $E^{\pm}_m(M)$. Corollary 3.1 shows that, for computing $E^{\pm}(M)$ and $E^{\pm}_m(M)$, it is sufficient to know the set $D^{\pm}_m(M)$ of minimal diagnoses.

Note that Theorem 3.2 generalises a result of Reiter's approach to diagnosis [113], since the former describes relationships between minimal hitting sets in a sense similar to the relationship between diagnoses and conflict sets of the latter.

In contrast, note that Theorem 3.1 ((a)) uses hitting sets without the requirement of $\subseteq$-minimality.

Example 3.15. *In Example 3.1 we have $E^{\pm}_m(M) = \{(\{r_1, r_2, r_4\}, \{r_5\})\}$ and $D^{\pm}_m(M) = \{(\{r_1\}, \emptyset), (\{r_2\}, \emptyset), (\{r_4\}, \emptyset), (\emptyset, \{r_5\})\}$. An explanation (E_1, E_2) has a nonempty intersection $E_1 \cap D_1 \neq \emptyset$ or $E_2 \cap D_2 \neq \emptyset$ with every minimal diagnosis (D_1, D_2). We thus obtain exactly one minimal explanation $E = (\{r_1, r_2, r_4\}, \{r_5\})$ by Corollary 3.1; furthermore, all component-wise supersets of E are explanations, as they also hit every minimal diagnosis, e.g. $(\{r_1, r_2, r_3, r_4, r_5\}, \{r_1, r_2, r_3, r_4, r_5\})$, and $(\{r_1, r_2, r_4\}, \{r_1, r_2, r_3, r_4, r_5\})$.*

For illustrating Theorem 3.2, consider the single minimal explanation (E_1, E_2) of M with $E_1 = \{r_1, r_2, r_4\}$ and $E_2 = \{r_5\}$. Then any minimal diagnosis (D_1, D_2) must fulfill $E_1 \cap D_1 \neq \emptyset$ or $E_2 \cap D_2 \neq \emptyset$, and there is no smaller pair (D_1, D_2) with that property. This condition holds for all minimal diagnoses in $D^{\pm}_m(M)$, and as they contain singleton sets only, and all rules in $E^{\pm}_m(M)$ have been 'hit' that way, it is easy to see that the condition cannot be true for any smaller pair $(D_1, D_2) \subset (D_1, D_2)$.

Duality

As it appears, explanations and diagnoses point out bridge rules as causes of inconsistency on a dual basis. Intuitively, bridge rules in E_1 of an explanation (E_1, E_2) cause inconsistency, while bridge rules in D_1 of a diagnosis (D_1, D_2) remove inconsistency; furthermore, adding unconditional forms of bridge rules from E_2 spoils inconsistency, while not adding unconditional forms of bridge rules from D_2 spoils consistency.

Both notions point out rules that are erroneous in the way that those rules contribute to inconsistency. This naturally gives rise to the question whether diagnoses and explanations point out the same rules of an MCS as erroneous, or whether they characterise different aspects.

To formalise this question, we introduce relevance for inconsistency. Given an MCS M, a bridge rule $r \in br(M)$ is *relevant for diagnosis* (*d-relevant*) iff there exists a minimal diagnosis (D_1, D_2) of M with $r \in D_1 \cup D_2$. Analogously, r is *relevant for explanation* (*e-relevant*) iff there exists a minimal explanation with $r \in E_1 \cup E_2$.

Example 3.16. *Recall our running example where* $D^{\pm}_{m}(M) = \big\{(\{r_1\}, \emptyset), (\{r_2\}, \emptyset), (\{r_4\}, \emptyset), (\emptyset, \{r_5\})\big\}$ *while* $E^{\pm}_{m}(M) = \{(\{r_1, r_2, r_4\}, \{r_5\})\}$.

Here the set of d-relevant bridge rules is $\{r_1, r_2, r_4, r_5\}$*. The set of e-relevant bridge rules is identical to that; in fact, even identical component-wise, i.e.,*

$$\bigcup\{D_1 \mid (D_1, D_2) \in D^{\pm}(M)\} = \{r_1, r_2, r_4\} = \bigcup\{E_1 \mid (E_1, E_2) \in E^{\pm}(M)\}$$

and

$$\bigcup\{D_2 \mid (D_1, D_2) \in D^{\pm}(M)\} = \{r_5\} = \bigcup\{E_2 \mid (E_1, E_2) \in E^{\pm}(M)\}.$$

As the following proposition shows, the component-wise coincidence is not accidental. Not only are the d-relevant rules exactly the same that are e-relevant, but this even holds if the components of diagnoses and explanations are treated separately. Formalising this, for any set X of pairs (A, B) we write $\bigcup X$ for $(\bigcup\{A \mid (A, B) \in X\}, \bigcup\{B \mid (A, B) \in X\})$.

Proposition 3.4. *For every inconsistent MCS* M*,* $\bigcup D^{\pm}_{m}(M) = \bigcup E^{\pm}_{m}(M)$*, i.e., the unions of all minimal diagnoses and all minimal inconsistency explanations coincide.*

Proposition 3.4 is an immediate consequence of the close structural relationships between diagnoses and explanations, which are shown by Theorems 3.1 and 3.2.

This provides evidence for our view that both notions capture exactly those parts of an MCS that are relevant for inconsistency, as duality shows that, in total, two very different perspectives on inconsistency state exactly the same parts of the MCS as erroneous.

In practice this allows one to compute the set of all bridge rules which are relevant for making an MCS consistent (i.e., appear in at least one diagnosis) in two ways: either to compute all minimal explanations, or to compute all minimal diagnoses. Furthermore, the duality result allows to exclude, under Occam's razor, all bridge rules that are not part of any diagnosis (or explanation) from further investigation as they can be skipped safely.

Our running example suggests, that duality also holds for deletion-diagnoses and deletion-explanations, which indeed is true:

Theorem 3.3. *For every inconsistent MCS* M*,* $\bigcup D^{-}_{m}(M) = \bigcup E^{+}_{m}(M)$*, i.e., the unions of all minimal deletion-diagnoses and all minimal deletion-inconsistency explanations coincide.*

Proof. This is a direct consequence of Proposition 3.4; set in its proof the second components of diagnoses and explanations to $\emptyset$. □

Asymmetry of Conversion

One notable aspect of the above conversion results is that minimal explanations and minimal diagnoses can be converted into one another, while for the respective non-minimal notions, only diagnoses can be converted to explanations, but not vice versa.

Intuitively, the reason why conversion is not symmetric stems from the fact that explanations ignore irrelevant inconsistencies, i.e., contrary to diagnoses, explanations are an order-increasing concept. This manifests in Proposition 3.1, which states that the supersets of an explanation are explanations again.

Conversely, a similar property does not hold for diagnoses, i.e., the supersets of a diagnosis may yield an inconsistent system. Consider an MCS M such that there is a diagnosis (D_1, D_2)

and $r \in br(M) \setminus D_1$ is a bridge rule not removed by the diagnosis. If r inhibits an inconsistency, then $(D_1 \cup \{r\}, D_2)$ is no diagnosis as the respective MCS is inconsistent. Although $E^{\pm}_m(M)$ is sufficient to determine all minimal diagnoses, it is not sufficient to determine which supersets of a minimal diagnosis actually yield a consistent system.

The following example illustrates this.

Example 3.17. *Consider the MCS $M = (C_1)$ where C_1 is an ASP context with knowledge base $kb_1 = \{\bot \leftarrow a. \quad \bot \leftarrow b, not\, c.\}$, and the bridge rules are:*

$$r_1 : (1{:}a) \leftarrow \mathbf{not}\ (1 : d).$$
$$r_2 : (1{:}b) \leftarrow \mathbf{not}\ (1 : d).$$
$$r_3 : (1{:}c) \leftarrow \mathbf{not}\ (1 : d).$$

Consider $D = (\{r_1\}, \{\})$ and $D' = (\{r_1, r_3\}, \{\})$. Clearly, $D \subset D'$ and $D \in D^{\pm}_m(M)$ while $D' \notin D^{\pm}_m(M)$. On the other hand, $E^{\pm}_m(M) = \{(\{r_1\}, \{\})\}$, so by Proposition 3.1, D is a hitting set on all (non-minimal) explanations. D' also is a hitting set of all explanations, but it is no diagnosis.

Non-Overlap in Minimal Diagnoses

We conclude a simple but useful property of minimal diagnoses. Definition 3.1 reveals that, if (D_1, D_2) with $r \in D_2$ is a diagnosis, then $(D_1 \setminus \{r\}, D_2)$ and $(D_1 \cup \{r\}, D_2)$ also are diagnoses. For minimal diagnoses we therefore conclude the following.

Proposition 3.5. *Every minimal diagnosis (D_1, D_2) of an MCS M, fulfils $D_1 \cap D_2 = \emptyset$, i.e., no rule occurs in both components.*

Proof. Let $(D_1, D_2) \in D^{\pm}_m(M)$ and let S be a witnessing belief state for it, i.e., S is an equilibrium of $M[br(M) \setminus D_1 \cup cf(D_2)]$. Towards contradiction, assume that $D_1 \cap D_2 \neq \emptyset$. Consider any bridge rule $r \in D_1 \cap D_2$ and let $C_h(r) = i$ and $\varphi(r) = p$. Furthermore, consider $r' = cf(r) = (i : p) \leftarrow .$, then $body(r') = \emptyset$ and thus r' is applicable in any belief state. Therefore, $r' \in app(br_i(M[br(M) \setminus D_1 \cup cf(D_2)]), S)$ and consequently $p \in \{\varphi(r) \mid r \in app(br_i(M[br(M) \setminus D_1 \cup cf(D_2)]), S)\}$. For $(D'_1, D'_2) = (D_1 \setminus \{r\}, D_2)$, we thus obtain that $p \in \{\varphi(r) \mid r \in app(br_i(M[br(M) \setminus D'_1 \cup cf(D'_2)]), S)\}$ and since all other bridge rules are as before, we conclude that $app(br_i(M[br(M) \setminus D'_1 \cup cf(D'_2)]), S) = app(br_i(M[br(M) \setminus D_1 \cup cf(D_2)]), S)$ for all $i \in C(M)$. Consequently S is an equilibrium of $M[br(M) \setminus D'_1 \cup cf(D'_2)]$ and $(D'_1, D'_2) \in D^{\pm}(M)$. But $(D'_1, D'_2) \subset (D_1, D_2)$ contradicts $(D_1, D_2) \in D^{\pm}_m(M)$, which proves the result. □

An analogue property does not hold for inconsistency explanations; as shown by Example 3.5: the minimal explanation (E_1, E_2) with $E_1 = \{r_1, r_2, r_4, r_6, r_7\}$ and $E_2 = \{r_6, r_7\}$ is such that r_6 and r_7 are present in both E_1 and E_2.

A minimal diagnosis also is such that only bridge rules that otherwise fire are removed and only those that otherwise do not fire are made unconditional, as the following proposition shows.

Proposition 3.6. *Let M be an MCS and $(D_1, D_2) \in D^{\pm}_m(M)$ hold. Then, if holds for every $S \in \text{EQ}(M[D_1, D_2])$ that $r \in D_1$ implies $S \models\!\!\leadsto r$ and that $r \in D_2$ implies that $S \not\models\!\!\leadsto r$.*

Proof. Let $(D_1, D_2) \in D^{\pm}_m(M)$ hold for some MCS $M = (C_1, \ldots, C_n)$. Assume towards contradiction that there exists $S \in \text{EQ}(M[D_1, D_2])$ such that (1) there exists $r \in D_1$ with $S \not\models\!\!\leadsto r$ or (2) there exists $r \in D_2$ with $S \models\!\!\leadsto r$.

Case (1): consider $(D'_1, D'_2) = (D_1 \setminus \{r\}, D_2)$ and note that $M[D_1, D_2]$ differs from $M[D'_1, D'_2]$ only by the fact that $r \in br_i(M[D'_1, D'_2])$ holds for $C_h(r) = i$ while it holds that $r \notin br_i(M[D_1, D_2])$. Consider the belief state $S = (S_1, \ldots, S_n)$ and observe that for all $1 \leq j \leq n$ with $j \neq i$ holds that $S_j \in \mathbf{ACC}_j(kb_j \cup \{\varphi(r) \mid r \in app(br_j(M[D'_1, D'_2]), S)\})$ since $S \in \text{EQ}(M[D_1, D_2])$ holds and $br_j(M[D_1, D_2]) = br_j(M[D'_1, D'_2])$ holds for all $1 \leq j \leq n$ with $j \neq i$. To show that $S \in \text{EQ}(M[D'_1, D'_2])$ holds, it therefore only remains to show that $S_i \in \mathbf{ACC}_i(kb_i \cup \{\varphi(r) \mid r \in app(br_j(M[D'_1, D'_2]), S)\})$ holds: since $br_j(M[D'_1, D'_2]) = br_j(M[D_1, D_2]) \setminus \{r\}$ and $S \not\models\!\!\leadsto r$ holds, it follows that $app(br_j(M[D'_1, D'_2]), S) = app(br_j(M[D_1, D_2]), S)$ and consequently that $S \in \text{EQ}(M[D'_1, D'_2])$ holds. Since $(D'_1, D'_2) \subseteq (D_1, D_2)$ and $(D'_1, D'_2) \in D^{\pm}(M)$ holds, this contradicts that $(D_1, D_2) \in D^{\pm}_m(M)$ holds.

Case (2): consider $(D'_1, D'_2) = (D_1, D_2 \setminus \{r\})$ and observe that again $S = (S_1, \ldots, S_n)$ is such that for all $1 \leq j \leq n$ with $j \neq C_h(r)$ holds $S_j \in \mathbf{ACC}_j(kb_j \cup \{\varphi(r) \mid r \in app(br_j(M[D'_1, D'_2]), S)\})$ since it holds that $S \in \text{EQ}(M[D_1, D_2])$ and $br_j(M[D_1, D_2]) = br_j(M[D'_1, D'_2])$ for all $1 \leq j \leq n$ with $j \neq i$. Similarly, as in the other case, it holds that $S \models\!\!\leadsto r$ and hence $app(br_j(M[D'_1, D'_2]), S) = app(br_j(M[D_1, D_2]), S)$. Again, $(D'_1, D'_2) \in D^{\pm}(M)$ holds and because $(D'_1, D'_2) \subset (D_1, D_2)$ holds, this contradicts that $(D_1, D_2) \in D^{\pm}_m(M)$ holds.

Since both cases lead to contradiction, it follows that there exists no such $S \in \text{EQ}(M[D_1, D_2])$ with either $r \in D_1$ implies $S \not\models\!\!\leadsto r$ or $r \in D_2$ implies $S \models\!\!\leadsto r$. Consequently, it holds for all $S \in \text{EQ}(M[D_1, D_2])$ that $r \in D_1$ implies $S \models\!\!\leadsto r$ and that $r \in D_2$ implies that $S \not\models\!\!\leadsto r$. □

Modularity of Explanations and Diagnoses

We next give a syntactic criterion which enables the computation of explanations for an MCS M in a divide-and-conquer fashion. In particular, minimal explanations of M are then just combinations of the minimal explanations of the smaller parts. Based on the results about conversion between explanations and diagnoses, these results then carry over to diagnoses as well. This can be exploited to compute minimal explanations and minimal diagnoses for certain classes of MCS more efficiently.

An approach to modularisation (in particular for hierarchical and partitionable MCS) is that some part does not impact the rest of the system. To this end, we adapt the notion of *splitting set* as introduced by [96] in the context of logic programming; a splitting set characterises a subset of a logic program which is independent of other rules in the program by a syntactic property.

Since an MCS may include contexts with arbitrary logics, a purely syntactical criterion can only be obtained by resorting to beliefs occurring in bridge rules, under the implicit assumption that every output belief of a context depends on every input belief of the context. Hence, we split at the level of contexts, i.e., a splitting set is a set of contexts rather than a set of literals.

Definition 3.8. *A set of contexts* $U \subseteq C(M)$ *is a* splitting set *of an MCS* M*, if every rule* $r \in br(M)$ *is such that* $C_h(r) \in U$ *satisfies* $C_b(r) \subseteq U$*. More formally,* U *is a splitting set iff* $U \supseteq \bigcup\{C_b(r) \mid r \in br(M), C_h(r) \in U\}$*.*

For such U*, the set* $b_U = \{r \in br(M) \mid C_h(r) \in U\}$ *is called the* bottom *relative to* U*.*

Example 3.18. *In our running example, we have* $C(M) = \{C_1, \ldots, C_4\}$*, with e.g.,* $C_h(r_1) = C_h(r_2) = C_3$*, and* $C_b(r_1) = C_b(r_2) = \{C_2\}$*. So the set* $U_1 = \{C_2, C_3\}$ *is a splitting set of* M*; its bottom is* $b_{U_1} = \{r_1, r_2\}$*.*

The further splitting sets of M *are* $U_2 = \{C_1\}$ *with* $b_{U_2} = \emptyset$*,* $U_3 = \{C_2\}$ *with* $b_{U_3} = \emptyset$*, and* $U_4 = \{C_4, C_3, C_2, C_1\}$ *with bottom* $b_{U_4} = br(M)$*.*

Intuitively, if U is a splitting set of M, then the consistency (respectively inconsistency) of contexts in U does not depend on the contexts in $C(M) \setminus U$. Thus, if $M[b_U]$ is inconsistent, M stays inconsistent (under the assumption that $M[\emptyset] \not\models \bot$).

Lemma 3.2. *Let* U *be a splitting set of an MCS* M *and let* $R_1, R_2 \subseteq br(M)$*. Then,* U *is also a splitting set of* $M[R_1 \cup cf(R_2)]$*.*

Proof. Towards contradiction assume that U is not a splitting set for $M[R_1 \cup cf(R_2)]$, i.e., there exists a rule $r \in br(M[R_1 \cup cf(R_2)])$ such that $C_h(r) \in U$ and $C_b(r) \not\subseteq U$. Thus, there exists $(i : p) \in body^{\pm}(r)$ such that $i \notin U$. Since $body^{\pm}(r') = \emptyset$ for all $r' \in cf(R_2)$, it follows that $r \in R_1$ and since $R_1 \subseteq br(M)$, it follows that $r \in br(M)$. By the assumption that $C_h(r) \in U$ and because U is a splitting set of M, it follows that $i \in U$ for all $(i : p) \in body^{\pm}(r)$, which contradicts that $C_b(r) \not\subseteq U$. Therefore, no such r can exist and U is also a splitting set of $M[R_1 \cup cf(R_2)]$. □

Lemma 3.3. *Let* M *be an MCS, let* B *be a set of bridge rules compatible with* M*, and let* $U \subseteq C(M)$ *be a splitting set for* $M[B]$*. Then, for every* $i \in U$ *and belief state* $S = (S_1, \ldots, S_n)$ *of* M *it holds that:*

$$S_i \in \mathbf{ACC}_i(kb_i \cup app(br_i(M[b_U]), S)) \text{ iff } S_i \in \mathbf{ACC}_i(kb_i \cup app(br_i(M[B]), S)).$$

Proof. We first show that $br_i(M[b_U]) = br_i(M[B])$ holds for all $i \in U$:

($\subseteq$) From the definition of the bottom, b_U, it follows that $b_U \subseteq B$, thus $br_i(M[b_U]) \subseteq br_i(M[B])$.

($\supseteq$) Consider $r \in br_i(M[B])$. It holds that $C_h(r) = i$. Since U is a splitting set and $i \in U$ it follows that $r \in b_U$ by definition of the bottom b_U. Hence, $br_i(M[b_U]) \supseteq br_i(M[B])$.

As a consequence of $br_i(M[b_U]) = br_i(M[B])$, it follows that $app(br_i(M[b_U]), S) = app(br_i(M[B]), S)$ holds for all $i \in U$, and therefore it is also the case that $\mathbf{ACC}_i(kb_i \cup app(br_i(M[b_U]), S)) = \mathbf{ACC}_i(kb_i \cup app(br_i(M[B]), S))$, which proves the lemma. □

Observe that splitting sets preserve acceptability not only when bridge rules in the remainder of the MCS are modified (as in Lemma 3.3), but also when belief sets in the remainder are exchanged. For two belief states $S = (S_1, \ldots, S_n)$ and $S' = (S'_1, \ldots, S'_n)$ of an MCS, we say that S coincides with S' on U, written $S =_U S'$, if for all $i \in U$ holds $S_i = S'_i$.

Lemma 3.4. *Let M be an MCS, let B be a set of bridge rules compatible with M, and let U be a splitting set for $M[B]$. Furthermore, let $S = (S_1, \ldots, S_n)$ and $S' = (S'_1, \ldots, S'_n)$ be belief states of M, and let $b_U \subseteq R \subseteq B$. Then, $S =_U S'$ and $i \in U$ implies $\mathbf{ACC}_i(kb_i \cup app(br_i(M[B]), S)) = \mathbf{ACC}_i(kb_i \cup app(br_i(M[R]), S'))$.*

Proof. Since $b_U \subseteq R$ it holds for all $i \in U$ that $br_i(M[B]) = br_i(M[R])$. Furthermore, because U is a splitting set, it follows that $c \in U$ for all $(c : p) \in body^{\pm}(r)$ such that $r \in br_i(M[B])$ and $i \in U$. As a consequence $p \in S_c$ iff $p \in S'_c$ since S and S' coincide on U and $r \in br_i(M[B])$ iff $r \in br_i(M[R])$. □

For a pair $R = (R_1, R_2)$ of sets of bridge rules compatible with M and a set U of contexts we say that R is *U-headed* iff $r \in (R_1 \cup R_2)$ implies $C_h(r) \in U$.

Proposition 3.7. *Suppose U is a splitting set of an MCS M. Then,*

(i) $E \in E^{\pm}(M[b_U])$ iff $E \in E^{\pm}(M)$ and E is U-headed, and

(ii) $D \in D^{\pm}(M[b_U])$ iff there exists some $D' \in D^{\pm}(M)$ such that $D \subseteq D'$.

Proof. For reasoning about explanations, the concept of explanation range proves to be useful. For a given pair $E = (E_1, E_2) \in 2^{br(M)} \times 2^{br(M)}$ of sets of bridge rules and $B \subseteq br(M)$, the *explanation range* of E with respect to B is $Rg(E, B) = \{(R_1, R_2) \mid E_1 \subseteq R_1 \subseteq B \text{ and } R_2 \subseteq B \setminus E_2\}$. Intuitively, $Rg(E, B)$ are "relevant pairs" for E with respect to the upper bound B. It follows directly from Definition 3.3 that, $E = (E_1, E_2) \in E^{\pm}(M)$ iff $M[R_1 \cup cf(R_2)] \models \bot$ for all $(R_1, R_2) \in Rg(E, br(M))$.

In the following we prove Item (i): $E \in E^{\pm}(M[b_U])$ holds iff $E \in E^{\pm}(M)$ holds and E is U-headed.

($\Rightarrow$) Let $(R'_1, R'_2) \in Rg(E, br(M))$ be arbitrary, then both $R'_1 \subseteq br(M)$ and $R'_2 \subseteq br(M)$. By Lemma 3.2, U is also a splitting set for the MCS $N' = M[R'_1 \cup cf(R'_2)]$.

Let $R_1 = R'_1 \cap b_U$ and let $R_2 = R'_2 \cap b_U$. As $E_1, E_2 \subseteq b_U$, it follows that $(R_1, R_2) \in Rg(E, b_U)$. Because E is an explanation of $M[b_U]$, it holds for $N = M[R_1 \cup cf(R_2)]$ that $N \models \bot$, i.e., for every belief state S exists a context $i \in U$ with $S_i \notin \mathbf{ACC}_i(kb_i \cup app(br_i(N), S))$.

Since $B = R'_1 \cup cf(R'_2)$ is compatible with M and U is a splitting set for $N' = M[B]$, we conclude from Lemma 3.3 that for every belief state S it holds that $S_i \in \mathbf{ACC}_i(kb_i \cup app(br_i(N'), S))$ iff $S_i \in \mathbf{ACC}_i(kb_i \cup app(br_i(N), S))$. Since $N \models \bot$ this implies that for every S there exists some $i \in U$ such that $S_i \notin \mathbf{ACC}_i(kb_i \cup app(br_i(N'), S))$ and thus $N' \models \bot$. Since $(R'_1, R'_2) \in Rg(E, br(M))$ is arbitrary, it follows that $E \in E^{\pm}(M)$. Furthermore, E is U-headed by definition.

($\Leftarrow$) Let $E = (E_1, E_2) \in E^{\pm}(M)$ such that E is U-headed, and consider some arbitrary $(R_1, R_2) \in Rg(E, b_U)$. Since $b_U \subseteq br(M)$, we conclude that $(R_1, R_2) \in Rg(E, br(M))$. Since E is an explanation of M, it follows that $N = M[R_1 \cup cf(R_2)]$ is such that $N \models \bot$. As this holds for every $(R_1, R_2) \in Rg(E, b_U)$, it follows that $(E_1, E_2) \in E^{\pm}(M[b_U])$.

This establishes item (i).

Next we prove Item (ii): $D \in D^{\pm}(M[b_U])$ holds iff there exists $D' \in D^{\pm}(M)$ such that $D \subseteq D'$.

($\Rightarrow$) Let $D = (D_1, D_2) \in D^{\pm}(M[b_U])$. Then, there exists an equilibrium S of $M[R]$ where $R = (b_U \setminus D_1) \cup cf(D_2)$. Consider $(D'_1, D'_2) = (D_1 \cup (br(M) \setminus b_U), D_2)$ and observe that $(br(M) \setminus D'_1) \cup cf(D'_2) = R$, because $br(M) \setminus D'_1 = b_U \setminus D_1$. Since S is an equilibrium of $M[R]$, it follows that $D' \in D^{\pm}(M)$.

($\Leftarrow$) Assume $D' \in D^{\pm}(M)$ where $D' = (D'_1, D'_2)$. First assume that $E^{\pm}(M[b_U]) = \emptyset$, i.e., $M[b_U]$ is consistent. Then, $D = (\emptyset, \emptyset) \in D^{\pm}(M[b_U])$, hence $D \subseteq D'$ and $D \in D^{\pm}(M[b_U])$.

Otherwise, $E^{\pm}(M[b_U]) \neq \emptyset$. Consider $(D_1, D_2) = (D'_1 \cap b_U, D'_2 \cap b_U)$ and let $R' = br(M) \setminus D'_1 \cup cf(D'_2)$ and $R = b_U \setminus D_1 \cup cf(D_2)$. Observe that $br_j(M[R]) = \emptyset$ for all $j \in C(M) \setminus U$, because $R \subseteq b_U \cup cf(b_U)$ and for no rule $r \in b_U \cup cf(b_U)$ it holds that $C_h(r) = j$.

As $M[\emptyset]$ is consistent, there exists some $S_j^0 \in \mathbf{ACC}_j(kb_j)$ for every $j \in C(M)$. Let $S' = (S'_1, \ldots, S'_n)$ be an equilibrium for $M[R']$ (which exists because $D' \in D^{\pm}(M)$). Let $S = (S_1, \ldots, S_n)$ such that $S_i = S'_i$ if $i \in U$, and $S_i = S_i^0$ otherwise. Then, S is an equilibrium for $M[R]$. Indeed, first consider $i \in C(M) \setminus U$. Since $br_i(M[R]) = \emptyset$, it follows that $app(br_i(M[R]), S) = \emptyset$, hence $S_i^0 \in \mathbf{ACC}_i(kb_i \cup app(br_i(M[R]), S))$. Second, consider $i \in U$. Note that U is a splitting set of $M[R]$, because $br_j(M[R]) = \emptyset$ for all $j \in C(M) \setminus U$. Since $b_U \subseteq R \subseteq R'$ and $S =_U S'$, it follows from Lemma 3.4 that $\mathbf{ACC}_i(kb_i \cup app(br_i(M[R]), S)) = \mathbf{ACC}_i(kb_i \cup app(br_i(M[R']), S'))$. From $S'_i \in \mathbf{ACC}_i(kb_i \cup app(br_i(M[R']), S'))$ and $S_i = S'_i$, it thus follows that $S_i \in \mathbf{ACC}_i(kb_i \cup app(br_i(M[R]), S))$.

Consequently, $S_i \in \mathbf{ACC}_i(kb_i \cup app(br_i(M[R]), S))$ for all $i \in C(M)$; hence S is an equilibrium of $M[R]$. Since $R_1 \cup R_2 \subseteq b_U$, it follows that $D \in D^{\pm}(M[b_U])$. □

Corollary 3.2. *Every minimal explanation of $M[b_U]$ is a minimal explanation of M.*

Proof. Let $E \in E_m^{\pm}(M[b_U])$, then it follows from Proposition 3.7 that $E \in E^{\pm}(M)$ and E is U-headed. Assume for a contradiction that $E \notin E_m^{\pm}(M)$. Hence, there exists some $E' \in E^{\pm}(M)$ such that $E' \subset E$. Since E is U-headed, it follows that E' also is U-headed. Thus by Proposition 3.7 it follows that $E' \in E^{\pm}(M[b_U])$, which contradicts that $E \in E_m^{\pm}(M[b_U])$. □

Note that $M[b_U]$ does not yield all explanations that contain rules from b_U, but it yields all explanations that contain only rules from $M[b_U]$.

Example 3.19. *Reconsider the MCS M from Example 3.1, where the laboratory database together with the disease ontology forms a splitting set $U = \{C_2, C_3\}$ with $b_U = \{r_1, r_2\}$. Now $M[b_U]$ is consistent, so $E^{\pm}(M[b_U]) = \emptyset$, but the overall MCS is inconsistent with the minimal explanation $E = (\{r_1, r_2, r_4\}, \{r_5\})$. In line with Proposition 3.7, E contains rules from b_U but E is not b_U-headed.*

In the particular case that two splitting sets form a partitioning of the MCS, then both partitions can be treated without considering the other one. This means that explanations only contain rules from one partition and diagnoses of the whole MCS are obtained by simply combining diagnoses of each of the partitions.

Proposition 3.8. *Suppose that both $U \subseteq C(M)$ and $U' = C(M) \setminus U$ are splitting sets of an MCS M. Then, every $E \in E_m^{\pm}(M)$ is either U-headed or U'-headed.*

Proof. As in the proof of Proposition 3.7, let $(R_1, R_2) \in \mathit{Rg}(E, B)$ iff $E_1 \subseteq R_1 \subseteq B$ and $R_2 \subseteq B \setminus E_2$. Given a splitting set $V \subseteq C(M)$ and sets of bridge rules $R_1, R_2 \subseteq br(M)$, we call $M[(R_1 \cap b_V) \cup \mathit{cf}(R_2 \cap b_V)]$ the V*-projection* of M wrt. $R = (R_1, R_2)$.

W.l.o.g. assume that $M = (C_1, \dots, C_n)$, $U = \{1, \dots, k\}$, and $U' = \{k+1, \dots, n\}$, where $1 \leq k < n$. Towards contradiction assume that some $E = (E_1, E_2) \in E^{\pm}_m(M)$ exists which contains rules from both, b_U and $b_{U'}$. Consider an arbitrary $(R_1, R_2) \in \mathit{Rg}(E, br(M))$. Since E is an explanation, it holds that $M[R_1 \cup \mathit{cf}(R_2)] \models \bot$.

We prove that for every $R = (R_1, R_2) \in \mathit{Rg}(E, br(M))$ either its U-projection or its U'-projection is inconsistent, or both.

Towards contradiction assume that neither projection is inconsistent. Then, there exists an equilibrium $S = (S_1, \dots, S_n)$ of the U-projection of M wrt. R and an equilibrium $S' = (S'_1, \dots, S'_n)$ of the U'-projection of M wrt. R. Consider the belief state $S'' = (S_1, \dots, S_k, S'_{k+1}, \dots, S'_n)$. By Lemma 3.4, it holds that $S_i \in \mathbf{ACC}_i(kb_i \cup \mathit{app}(br_i(M[R_1 \cup \mathit{cf}(R_2)]), S''))$ for all $i \in U$, because U is a splitting set of $M[R_1 \cup \mathit{cf}(R_2)]$, $b_U \subseteq (R_1 \cap b_U) \cup \mathit{cf}(R_2 \cap b_U) \subseteq R_1 \cup \mathit{cf}(R_2)$, and $S =_U S''$.
Analogously, it holds that $S'_i \in \mathbf{ACC}_i(kb_i \cup \mathit{app}(br_i(M[R_1 \cup \mathit{cf}(R_2)]), S''))$ for all $i \in U'$. Consequently, S'' is an equilibrium of $M[R_1 \cup \mathit{cf}(R_2)]$, which contradicts that E is an explanation. Therefore, for every $R \in \mathit{Rg}(E, br(M))$ it holds that either the U-projection of R, the U'-projection of R, or both are inconsistent.

Next, we distinguish for all $R \in \mathit{Rg}(E, br(M))$ which projections are inconsistent.

Case (1): for every $R \in \mathit{Rg}(E, br(M))$ its U-projection is inconsistent. Then, $E' = (E_1 \cap b_U, E_2 \cap b_U)$ is an explanation, since for every $R' \in \mathit{Rg}(E', br(M))$ it holds that R' is a U-projection of some $R \in \mathit{Rg}(E, br(M))$, which is inconsistent. Since $E_1 \cup E_2 \not\subseteq b_U$, we have $E' \subset E$. Since $E' \in E^{\pm}(M)$, it follows that $E \notin E^{\pm}_m(M)$, which contradicts the assumption that $E \in E^{\pm}_m(M)$.
Case (2): for all $R \in \mathit{Rg}(E, br(M))$ it holds that the U'-projection is inconsistent. Analogously to the previous case, we conclude that $E' = (E_1 \cap b_{U'}, E_2 \cap b_{U'})$ is an explanation of M such that $E' \subset E$, which contradicts the assumption that $E \in E^{\pm}_m(M)$.
Case (3): Neither case (1) nor case (2) applies. That is, for some $R = (R_1, R_2) \in \mathit{Rg}(E, br(M))$ the U-projection is consistent, and also for some $R' = (R'_1, R'_2) \in \mathit{Rg}(E, br(M))$ the U'-projection is consistent. This means that there exists some belief state $S = (S_1, \dots, S_n)$ such that $S_i \in \mathbf{ACC}_i(kb_i \cup \mathit{app}(br_i(M[(R_1 \cap b_U) \cup \mathit{cf}(R_2 \cap b_U)]), S))$ holds for all $i \in C(M)$ and there exists some belief state $S' = (S'_1, \dots, S'_n)$ such that for all $i \in C(M)$ it holds that $S'_i \in \mathbf{ACC}_i(kb_i \cup \mathit{app}(br_i(M[(R'_1 \cap b_{U'}) \cup \mathit{cf}(R'_2 \cap b_{U'})]), S))$.

Now consider $R'' = (R''_1, R''_2) = ((R_1 \cap b_U) \cup (R'_1 \cap b_{U'}), (R_2 \cap b_U) \cup (R'_2 \cap b_{U'}))$. First, we show that $R'' \in \mathit{Rg}(E, br(M))$. Since U and U' partition $C(M)$, it holds that $E_1 = (E_1 \cap b_U) \cup (E_1 \cap b_{U'})$; since $E_1 \subseteq R_1$, clearly $E_1 \cap b_U \subseteq R_1 \cap b_U$. Analogously, it holds that $E_1 \cap b_{U'} \subseteq R_1 \cap b_{U'}$. Consequently, $E_1 = (E_1 \cap b_U) \cup (E_1 \cap b_{U'}) \subseteq (R_1 \cap b_U) \cup (R'_1 \cap b_{U'})$; hence $E_1 \subseteq R''_1 \subseteq br(M)$. For R''_2 observe that $(R_2 \cup R'_2) \cap E_2 = \emptyset$ since both R_2 and R'_2 are disjoint with E_2 by definition. Therefore $((R_2 \cap b_U) \cup (R'_2 \cap b_{U'})) \cap E_2 = \emptyset$; hence $R''_2 \subseteq br(M) \setminus E_2$. In conclusion, it holds that $R'' \in \mathit{Rg}(E, br(M))$.

Second, we show that $S'' = (S_1, \dots, S_k, S'_{k+1}, \dots, S'_n)$ is an equilibrium of the MCS $M[R''_1 \cup \mathit{cf}(R''_2)]$. Since $S'' =_U S$ and, as already shown $R_1 \cap b_U \subseteq R''_1$ and $\mathit{cf}(R_2 \cap b_U) \subseteq$

$cf(R''_2)$, it follows by Lemma 3.4 that $S_i \in \mathbf{ACC}_i(kb_i \cup app(br_i(M[R''_1 \cup cf(R''_2)]), S''))$ for all $i \in U$. Analogously, the same is shown for U', i.e., for all $i \in U'$ it holds that $S'_i \in \mathbf{ACC}_i(kb_i \cup app(br_i(M[R''_1 \cup cf(R''_2)]), S''))$. Therefore, S'' is an equilibrium of $M[R''_1 \cup cf(R''_2)]$. Since $R'' \in Rg(E, br(M))$, it follows that $E \notin E^{\pm}(M)$. This is a contradiction to the assumption that $E \in E^{\pm}_m(M)$.

Since all cases yield a contradiction, it follows that every $E \in E^{\pm}_m(M)$ is either U-headed or U'-headed. □

Corollary 3.3. *Suppose $U \subseteq C(M)$ and $U' = C(M) \setminus U$ are splitting sets of an MCS M. Then, $E^{\pm}_m(M) = E^{\pm}_m(M[b_U]) \cup E^{\pm}_m(M[b_{U'}])$.*

Proof. ($\subseteq$) Let $E \in E^{\pm}_m(M)$. Then by Proposition 3.8, E is either U-headed or U'-headed. If E is U-headed, then $E \in E^{\pm}(M[b_U])$ by Proposition 3.7. Assume that $E \notin E^{\pm}_m(M[b_U])$. Hence some $E' \subset E$ exists such that $E' \in E^{\pm}_m(M[b_U])$. By Proposition 3.8, $E' \in E^{\pm}_m(M)$. This contradicts that $E \in E^{\pm}_m(M)$, which gives $E \in E^{\pm}_m(M[b_U])$. Analogously, if E is U'-headed, then $E \in E^{\pm}_m(M[b_{U'}])$. It follows that $E \in E^{\pm}_m(M[b_U]) \cup E^{\pm}_m(M[b_{U'}])$.

($\supseteq$) Let $E \in E^{\pm}_m(M[b_U])$ (respectively $E \in E^{\pm}_m(M[b_{U'}])$). Since U (respectively U') is a splitting set of M, from Corollary 3.2 it follows that $E \in E^{\pm}_m(M)$. In conclusion it holds that $E^{\pm}_m(M) \supseteq E^{\pm}_m(M[b_U]) \cup E^{\pm}_m(M[b_{U'}])$ □

Thus, using U, U' the MCS M can be partitioned into two parts where minimal explanations can be computed independently. From this and Theorem 3.2 we can conclude that for a partitionable MCS, the set of all minimal diagnoses can be obtained by combining the minimal diagnoses of each partition.

Proposition 3.9. *Suppose that U and $U' = C(M) \setminus U$ are splitting sets of an MCS M. Then,*

$$D^{\pm}_m(M) = \{(A_1 \cup B_1, A_2 \cup B_2) \mid (A_1, A_2) \in D^{\pm}_m(M[b_U]) \text{ and } (B_1, B_2) \in D^{\pm}_m(M[b_{U'}])\}.$$

Proof. By Corollary 3.3, $E^{\pm}_m(M) = E^{\pm}_m(M[b_U]) \cup E^{\pm}_m(M[b_{U'}])$, while by Theorem 3.2 each diagnosis is a minimal hitting set on $E^{\pm}_m(M)$. Because U and U' partition M, $E^{\pm}_m(M[b_U])$ and $E^{\pm}_m(M[b_{U'}])$ are on disjoint sets. Therefore the minimal hitting set of their unions is the pairwise combination of their minimal hitting sets. That is, $(D_1, D_2) \in minHS_M(E^{\pm}_m(M))$ iff $(D_1, D_2) = (A_1 \cup B_1, A_2 \cup B_2)$ with $(A_1, A_2) \in minHS_M(E^{\pm}_m(M[b_U])$ and $(B_1, B_2) \in minHS_M(E^{\pm}_m(M[b_{U'}])$. From Theorem 3.2 it follows that $D^{\pm}_m(M) = minHS_M(M)$. This proves the proposition. □

We combine the MCS from Example 3.8 and Example 3.9 to obtain a partitionable MCS.

Example 3.20. *Consider the MCS $M_c = (C''_1, C''_2, C''_3, C''_4, C''_5) = (C_1, C_2, C'_1, C'_2, C'_3)$ which combines the MCS M from Example 3.8 and a primed version M' of the MCS from Example 3.9. This requires some rewriting of context identifiers in bridge rules. The full details of this follow in the next section. The resulting MCS is depicted in Figure 3.7. Obviously M_c has a partitioning (U, U') where $U = \{1, 2\}$ and $U' = \{4, 5, 6\}$.*

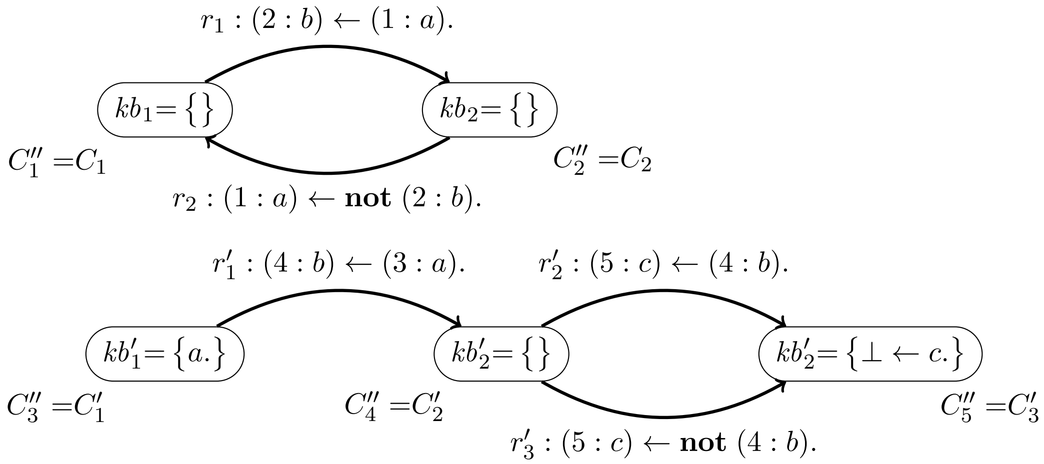


Figure 3.7: The combination of the MCS from Example 3.8 and Example 3.9. Context identifiers in bridge rules are adapted (cf. details in the following section).

Then, $E^{\pm}_m(M_c) = \{(\{r_1, r_2\}, \{r_1, r_2\}), (\{r'_1, r'_2\}, \emptyset), (\{r'_2, r'_3\}, \emptyset)\} = E^{\pm}_m(M) \cup E^{\pm}_m(M')$ *while*

$$
\begin{aligned}
D^{\pm}_m(M_c) =& \{(\{r_1, r'_2\}, \emptyset), (\{r_1, r'_1, r'_3\}, \emptyset) \\
& (\{r_2, r'_2\}, \emptyset), (\{r_2, r'_1, r'_3\}, \emptyset) \\
& (\{r'_2\}, \{r_1\}), (\{r'_1, r'_3\}, \{r_1\}) \\
& (\{r'_2\}, \{r_2\}), (\{r'_1, r'_3\}, \{r_2\}) \\
=& \{(A_1 \cup B_1, A_2 \cup B_2) \mid (A_1, A_2) \in D^{\pm}_m(M_U), (B_1, B_2) \in D^{\pm}_m(M_{U'})\}.
\end{aligned}
$$

Shifting and Decomposition of Contexts

In this section, we investigate how MCS can be combined or broken up. Based on the previous splitting-set results, we show that the diagnoses of a combined MCS are cross-products of the diagnoses of the combined parts and vice versa, i.e., for finding diagnoses, an MCS may be broken into smaller partitions, given that all bridge rules of the original MCS also appear in one of the partitions.

Since the contexts and bridge rules in an MCS are identified by their position, a way of shifting/manipulating indices of such positional identifiers is necessary. For this, we shift indices according to a permutation $I : \mathbb{N} \to \mathbb{N}$, i.e., I is a bijective mapping. Given a bridge rule r of form (2.1), then $I(r)$ is the bridge rule $(I(k) : s) \leftarrow (I(c_1) : p_1), \dots, (I(c_j) : p_j), \mathbf{not}\ (I(c_{j+1}) : p_{j+1}), \dots, \mathbf{not}\ (I(c_m) : p_m)$; furthermore, for a set R of bridge rules we have $I(R) = \{I(r) \mid r \in R\}$ and for a context $C_i = (L_i, kb_i, br_i)$ we have $I(C_i) = (L_i, kb_i, I(br_i))$. Given an MCS $M = (C_1, \dots, C_n)$, a permutation I is *compatible* with M if $I(x) \leq n$ holds for all $x \leq n$, i.e., I is a permutation on $C\,(M)$; the "shuffled" version of M wrt. a compatible I then is $I(M) = (I(C_{I^{-1}(1)}), \dots, I(C_{I^{-1}(n)}))$. Given a belief state $S = (S_1, \dots, S_n)$ we have $I(S) = (S_{I^{-1}(1)}, \dots, S_{I^{-1}(n)})$

To combine two existing MCS into a new one, we use the following $\otimes$ operator: given the MCS $M = (C_1, \ldots, C_n)$ and $M' = (C'_1, \ldots, C'_m)$ their combination is

$$M \otimes M' = (C_1, \ldots, C_n, I(C'_1), \ldots, I(C'_m))$$

where $I(x) = \begin{cases} n + x & \text{for } 1 \leq x \leq m, \\ x - m & \text{for } m+1 \leq x \leq n+m, \\ x & \text{otherwise.} \end{cases}$

In the following, we call I the *permutation wrt.* $M \otimes M'$. Note that by construction the permutation I wrt. $M \otimes M'$ is compatible with $M \otimes M'$. Recall that $M[R_1, R_2] = M[br(M) \setminus R_1 \cup cf(R_2)]$. Regarding modifications and diagnosis candidates, we then observe that

$$M[A_1, A_2] \otimes M'[B_1, B_2] = (M \otimes M')[A_1 \cup I(B_1), A_2 \cup I(B_2)]$$

where I is the mapping wrt. $M \otimes M'$.

The following lemma shows that shifting alone has no influence on acceptability.

Lemma 3.5. *Given an MCS* $M = (C_1, \ldots, C_n)$ *and a compatible permutation* I*, then* $S \in \text{EQ}(M)$ *holds iff* $I(S) \in \text{EQ}(I(M))$ *holds. Furthermore,* $S \in \text{EQ}(M[D_1, D_2])$ *holds iff* $I(S) \in \text{EQ}(I(M[D_1, D_2]))$ *holds.*

Proof. Observe that I is a bijection on $\{1, \ldots, n\}$ which simply renames context identifiers. Therefore, one can directly conclude that $S \in \text{EQ}(M)$ holds iff $I(S) \in \text{EQ}(I(M))$ holds. In the following, we show in full detail that this renaming indeed is correct.

Let $S = (S_1, \ldots, S_n)$ and $I(S) = (S_{I^{-1}(1)}, \ldots, S_{I^{-1}(n)}) = (S'_1, \ldots, S'_n)$. and let $1 \leq i \leq n$. Note that $S \in \text{EQ}(M)$ holds iff for all $1 \leq i \leq n$ holds $S_i \in \mathbf{ACC}_i(kb_i \cup app(br_i(M), S))$; additionally $I(S) \in \text{EQ}(I(M))$ holds iff for all $1 \leq j \leq n$ holds $S_j \in \mathbf{ACC}_j(kb_j \cup app(br_j(I(M), I(S))$. Given that I is bijective and compatible to M, there exists $j \in \{1, \ldots, n\}$ for every $i \in \{1, \ldots, n\}$ such that $j = I(i)$ and vice versa, i.e., for every $j \in \{1, \ldots, n\}$ exists a $i \in \{1, \ldots, n\}$ such that $i = I^{-1}(j)$. We now show that for any $1 \leq i, j \leq n$ such that $j = I(i)$ it holds that $S_i \in \mathbf{ACC}_i(kb_i \cup app(br_i(M), S)$ iff $S_j \in \mathbf{ACC}_j(kb_j \cup app(br_j(I(M)), I(S))$. Observe that by construction of $I(M)$ it holds that $S_i = S_j$, $\mathbf{ACC}_i = \mathbf{ACC}_j$, and $kb_i = kb_j$. Hence it suffices to show that $app(br_i(M), S) = app(br_j(I(M)), I(S))$. Note that $br_j(I(M)) = I(br_i(M))$, hence there exists a bijection from $br_j(I(M))$ to $br_i(M)$, namely I; furthermore I also maps bijectively each $r \in br_i(M)$ and every $(c : p) \in body^{\pm}(r)$ to $I(r)$ and $(I(c) : p)$. Since $\varphi(r) = \varphi(I(r))$ it suffices to show that $p \in S_c$ holds iff $p \in S'_{I(c)}$ holds. This is true since $S'_{I(c)} = S_{I^{-1}(I(c))} = S_c$, thus it follows that $app(br_i(M), S) = app(br_j(I(M)), I(S))$ which in turn implies that $S \in \text{EQ}(M)$ iff $I(S) \in \text{EQ}(I(M))$.

From this we also conclude that $S \in \text{EQ}(M[D_1, D_2])$ holds iff $I(S) \in \text{EQ}(I(M[D_1, D_2]))$ holds, because $M[D_1, D_2]$ is an MCS, hence the above statement also applies to $M[D_1, D_2]$. □

To show that the set of diagnoses of $M \otimes M'$ is the product of the set of diagnoses of M and of M', we use the following lemma, which states that if M' has no bridge rules, the set of diagnoses of M coincides with the set of diagnoses of $M \otimes M'$.

Lemma 3.6. *Given an MCS $M = (C_1, \dots, C_n)$ and an MCS $M' = (C'_1, \dots, C'_m)$ with $br(M') = \emptyset$. Then for every belief state $(S_1, \dots, S_n)$ of M exist belief sets $S_{n+1}, \dots, S_{n+m}$ such that $(S_1, \dots, S_{n+m}) \in \text{EQ}(M \otimes M')$ holds iff $(S_1, \dots, S_n) \in \text{EQ}(M)$ holds.*

Proof. Let $M^o = M \otimes M'$.

"⇒": Let $S = (S_1, \dots, S_{n+m}) \in \text{EQ}(M \otimes M')$ be such that for every $1 \leq i \leq n + m$ holds $S_i \in \mathbf{ACC}_i(kb_i \cup app(br_i(M^o), S))$. Note that by construction of M^o it holds for every bridge rule $r \in br_i(M^o)$ with $1 \leq i \leq n$ that $(c : p) \in body^{\pm}(r)$ implies that $c \in \{1, \dots, n\}$ holds. Hence by $br_i(M^o) = br_i(M)$ follows that $app(br_i(M^o), S) = app(br_i(M), (S_1, \dots, S_n))$. Therefore, for all $i \in C(M)$ it holds that $S_i \in \mathbf{ACC}_i(kb_i \cup app(br_i(M), (S_1, \dots, S_n)))$, i.e., $(S_1, \dots, S_n) \in \text{EQ}(M)$.

"⇐": Let $S = (S_1, \dots, S_n) \in \text{EQ}(M)$ hold. Since $br(M') = \emptyset$, it holds for all $n + 1 \leq j \leq n + m$ that $br_j(M^o) = \emptyset$. Recall that contexts are consistent without bridge rules, i.e., there exists $S^{\emptyset}_j \in \mathbf{ACC}_j(kb_j \cup \emptyset)$ for all $n + 1 \leq j \leq n + m$. Consider the belief state $S' = (S_1, \dots, S_n, S^{\emptyset}_{n+1}, \dots, S^{\emptyset}_{n+m})$ and observe that for all $1 \leq i \leq n$ it holds that $app(br_i(M^o), S') = app(br_i(M), S)$ since $br_i(M^o) = br_i(M)$. It therefore follows that $S' \in \text{EQ}(M^o)$ holds. □

Since shifting has no influence on acceptability, we can turn around the above lemma to show that the set of diagnoses of $M \otimes M'$ equals the set of diagnoses of M' if $br(M) = \emptyset$.

Corollary 3.4. *Given an MCS $M = (C_1, \dots, C_n)$ and an MCS $M' = (C'_1, \dots, C'_{n'})$ with $br(M) = \emptyset$. Then, for every belief state $(S'_1, \dots, S'_{n'})$ of M' exist belief sets $S_1, \dots, S_n$ such that $(S_1, \dots, S_n, S'_1, \dots, S'_{n'}) \in \text{EQ}(M \otimes M')$ holds iff $(S'_1, \dots, S'_{n'}) \in \text{EQ}(M')$ holds.*

Proof. Consider a permutation I' that exchanges the positions of contexts of M and M' in $M \otimes M'$, formally: let I be the permutation wrt. $M \otimes M'$ and recall that I is compatible with $M \otimes M'$. Let $I' = I^{-1}$ and $M^s = I'(M \otimes M')$. Note that M^s equals $M' \otimes M$, hence by Lemma 3.5 we obtain that $(S_1, \dots, S_n, S'_1, \dots, S'_{n'}) \in \text{EQ}(M \otimes M')$ iff $I'((S_1, \dots, S_n, S'_1, \dots, S'_{n'})) \in \text{EQ}(M^s)$ iff $(S'_1, \dots, S'_{n'}, S_1, \dots, S_n) \in \text{EQ}(M' \otimes M)$.

Since $br(M) = \emptyset$ it holds by Lemma 3.6 that for every belief state $(S'_1, \dots, S'_{n'})$ of M' exist belief sets $S_{n'+1}, \dots, S_{n'+n}$ such that $(S'_1, \dots, S'_{n'}, S_{n'+1}, \dots, S_{n'+n}) \in \text{EQ}(M' \otimes M)$ holds iff $(S'_1, \dots, S'_{n'}) \in \text{EQ}(M')$ holds. In summary, $(S_1, \dots, S_n, S'_{n+1}, \dots, S'_{n+n'}) \in \text{EQ}(M \otimes M')$ holds iff $(S'_{n+1}, \dots, S'_{n+n'}) \in \text{EQ}(M')$ holds. □

We now continue with our main argument, namely that $M \otimes M'$ admits exactly those diagnoses which are a combination of a diagnosis of M and a diagnosis of M'.

Proposition 3.10. *Given two MCS M and M', then $D^{\pm}(M \otimes M') = \{(A_1 \cup I(B_1), A_2 \cup I(B_2)) \mid (A_1, A_2) \in D^{\pm}(M), (B_1, B_2) \in D^{\pm}(M')\}$ where I is the permutation wrt. $M \otimes M'$.*

Proof. W.l.o.g. let $M = (C_1, \dots, C_n)$, let $M' = (C'_1, \dots, C'_{n'})$, and let $M^o = M \otimes M'$. Observe that by construction, there is no bridge rule whose head belongs to M (resp. M') and whose body contains a belief from M' (resp. M). Consequently, $U = \{1, \dots, n\}$ and $U' = \{n+1, \dots, n+n'\} = C(M^o) \setminus U$ are both splitting sets of M^o. Let $S^{\emptyset} = (S^{\emptyset}_1, \dots, S^{\emptyset}_{n+n'})$

be an equilibrium of $M^o[\emptyset]$, which exists by our assumption that all contexts (of M and M') are consistent without bridge rules; additionally let $B = br(M^o) \setminus D_1 \cup cf(D_2)$.

"⇒": Let $(D_1, D_2) \in D^{\pm}(M^o)$ hold. Then there exists a belief state $S = (S_1, \dots, S_{n+n'})$ such that for every $1 \leq i \leq n + n'$ it holds that $S_i \in \mathbf{ACC}_i(kb_i \cup app(br_i(M^o[D_1, D_2]), S))$.

Consider $S_U = (S_1, \dots, S_n, S^{\emptyset}_{n+1}, \dots, S^{\emptyset}_{n+n'})$ and observe that $S_U =_U S$; hence by Lemma 3.4 it follows for all $i \in U$ that

$$\mathbf{ACC}_i(kb_i \cup app(br_i(M^o[B]), S)) = \mathbf{ACC}_i(kb_i \cup app(br_i(M^o[R]), S_U))$$

holds for all $b_U \subseteq R \subseteq B$, specifically for $R = b_U$. Note that U, U', and b_U meant here are relative to the MCS $M^o[B]$, where by Lemma 3.2 U and U' are also splitting sets of $M^o[B]$. Consequently, for all $i \in U$ it holds that $S_i \in \mathbf{ACC}_i(kb_i \cup app(br_i(M^o[b_U]), S_U))$ and for all $j \in C(M^o) \setminus U$ it holds that $S^{\emptyset}_j \in \mathbf{ACC}_j(kb_j \cup app(br_j(M^o[b_U]), S_U))$, because $br_j(M^o[b_U]) = \emptyset$; thus it holds that $S_U \in \text{EQ}(M^o[b_U])$. Recall that b_U is defined relative to $M^o[B]$, hence $b_U = br(M) \setminus (D_1 \cap br(M)) \cup cf(D_2 \cap br(M))$, i.e., for $A_1 = D_1 \cap br(M)$ and $A_2 = D_2 \cap br(M)$ it holds that $M^o[b_U] = M^o[br(M) \setminus A_1 \cup cf(A_2)]$ and it follows that $S_U \in \text{EQ}(M^o[br(M) \setminus A_1 \cup cf(A_2)])$, i.e., it holds that $(A_1, A_2) \in D^{\pm}(M^o[br(M)])$. Since $M^o[br(M)] = M \otimes M'[\emptyset]$, Lemma 3.6 applies, i.e., it holds that $(S_1, \dots, S_n) \in \text{EQ}(M[A_1, D_2])$ and we conclude that $(A_1, A_2) \in D^{\pm}(M)$.

The proof that $(B_1, B_2) \in D^{\pm}(M')$ for $B_1 = D_1 \cap I(br(M'))$ and $B_2 = D_2 \cap I(br(M'))$ is analogous; it is based on the belief state $S_{U'} = (S^{\emptyset}_1, \dots, S^{\emptyset}_n, S_{n+1}, \dots, S_{n+n'})$ which is a witness of $(I(B_1), I(B_2)) \in D^{\pm}(M^o[b_{U'}])$; applying Corollary 3.4 (for $(M \otimes M')[b_{U'}] = M \otimes M'[B_1, B_2]$) then yields that $(B_1, B_2) \in D^{\pm}(M')$.

"⇐": Let $(A_1, A_2) \in D^{\pm}(M)$ and $(B_1, B_2) \in D^{\pm}(M')$ hold. Then there exists some $S^A = (S^A_1, \dots, S^A_n) \in \text{EQ}(M[A_1, A_2])$ and $S^B = (S^B_1, \dots, S^B_{n'}) \in \text{EQ}(M'[B_1, B_2])$. Consider the belief state $S = (S_1, \dots, S_{n+n'})$ such that $S_i = S^A_i$ for $1 \leq i \leq n$ and $S_{n+j} = S^B_j$ for $1 \leq j \leq n'$. Observe that S is a belief state of the MCS $M^d = M^o[A_1 \cup I(B_1), A_2 \cup I(B_2)]$. Thus it suffices to show $S \in \text{EQ}(M^d)$, because this implies that $(A_1 \cup I(B_1), A_2 \cup I(B_2)) \in D^{\pm}(M \otimes M')$.

We first show that for all $1 \leq i \leq n$ it holds that $S_i \in \mathbf{ACC}_i(kb_i \cup app(br_i(M^d), S))$. Let $B = br(M^d)$; hence $M^d = M^d[B]$, and note that U and U' are splitting sets of $M^d[B]$ by Lemma 3.2. Next we consider $M^d[b_U]$ (with b_U relative to M^d) and $R = b_U$. Since $M^d[R] = M^d[b_U] = (M[A_1, A_2] \otimes M'[\emptyset])$ and $S^A \in \text{EQ}(M[A_1, A_2])$, it holds by Lemma 3.6 that there exist $S'_{n+1}, \dots, S'_{n+n'}$ such that $S^M = (S_1, \dots, S_n, S'_{n+1}, \dots, S'_{n+n'}) \in \text{EQ}(M[A_1, A_2] \otimes M'[\emptyset])$, i.e., for all $1 \leq i \leq n$ it holds that $S_i \in \mathbf{ACC}_i(kb_i \cup app(br_i(M^d[R]), S^M))$

It holds that $S^M =_U S$ and $b_U \subseteq R \subseteq B$; hence by Lemma 3.4 it holds for all $1 \leq i \leq n$ that $\mathbf{ACC}_i(kb_i \cup app(br_i(M^d[B]), S)) = \mathbf{ACC}_i(kb_i \cup app(br_i(M^d[R]), S^M))$. Consequently, it holds that $S_i \in \mathbf{ACC}_i(kb_i \cup app(br_i(M^d[B]), S))$ for all $1 \leq i \leq n$.

Second, we show that for all $n+1 \leq j \leq n'$ it holds that $S_j \in \mathbf{ACC}_i(kb_i \cup app(br_i(M^d), S)$. Consider $M^d[b_{U'}]$ (with $b_{U'}$ relative to M^d) and $R' = b_{U'}$. Since $M^d[R'] = M^d[b_{U'}] = M[\emptyset] \otimes M'[B_1, B_2]$ and $S^B \in \text{EQ}(M[B_1, B_2])$ hold, it follows by Corollary 3.4 that there exist $S'_1, \dots, S'_n$ such that $S^{M'} = (S'_1, \dots, S'_n, S_{n+1}, \dots, S_{n+n'}) \in \text{EQ}(M[\emptyset] \otimes M'[B_1, B_2])$, i.e., for all $n+1 \leq j \leq n'$ it holds that $S_j \in \mathbf{ACC}_i(kb_i \cup app(br_i(M^d), S^{M'}))$. Since it holds that $S^{M'} =_{U'} S$ and $b_{U'} \subseteq R' \subseteq B$, Lemma 3.4 applies and it follows that for all $n+1 \leq j \leq n+n'$

it holds that $\mathbf{ACC}_j(kb_j \cup app(br_j(M^d[B]), S)) = \mathbf{ACC}_j(kb_j \cup app(br_j(M^d[R']), S^{M'}))$. Consequently, it holds that $S_j \in \mathbf{ACC}_j(kb_j \cup app(br_j(M^d[B]), S))$ with $n + 1 \leq j \leq n + n'$.

In summary, it holds for every $1 \leq i \leq n + n'$ that S_i is accepted, i.e, $S \in \text{EQ}(M^d)$, hence $(A_1 \cup I(A_2), B_1 \cup I(B_2)) \in D^{\pm}(M \otimes M')$. □

Decomposition of Contexts

In this section, we investigate conditions that allow to decompose a context into two independent contexts, such that the overall semantics is preserved. Such decomposition can aid in computing the set of acceptable belief sets (since the input size for deciding whether a belief set is accepted may be reduced by the decomposition). Furthermore, decomposition can be used to de-centralise the information flow in an MCS. In the later sections, we develop and investigate several transformations of MCS, where the transformations and proofs are significantly simpler if we allow one central context where all information flows through. Using the decomposition techniques presented in the following allows to de-centralise the information flow again, after it has been centralised for ease of presentation and proving. This allows to keep the information flow as local as possible.

We proceed by first introducing output-projected equilibria and belief states. We define when a context is decomposable, subsequently we show that an MCS where one context is decomposed has the same diagnoses and equilibria as the original MCS. Finally, we give some syntactic criteria to decide whether a context can be decomposed.

Output-projected Equilibria

Output-projected equilibria have originally been introduced to analyse the computational complexity of recognising diagnoses and explanations. Intuitively, an output-projected belief state of an MCS M is the same as a belief state of M, except that only those beliefs are considered which occur in the body of some bridge rule of $br(M)$. Since the belief sets of an output-projected belief state contain only those beliefs occurring in bridge rules, decomposing a context with respect to output-projected beliefs is more general than decomposing a context with respect to all beliefs. We therefore use output-projected beliefs states and review this notion here.

Computing equilibria by guessing and verifying so-called "kernels of context belief sets" has been outlined in [48]. For the purpose of recognising diagnoses and explanations, it is sufficient to check for consistency, i.e., for existence of an arbitrary equilibrium in an MCS.

Here we first define *output beliefs*, which are the beliefs used in bodies of bridge rules. Then we show that for checking consistency of an MCS, it is sufficient to consider equilibria *projected to output beliefs*.

Definition 3.9. *Given an MCS $M = (C_1, \ldots, C_n)$, the* set of output beliefs of C_i, *$OUT_i = \{p \mid (i:p) \in body^{\pm}(r), r \in br(M)\}$, is the set of beliefs p of C_i that occur in the bodies of bridge rules.*

Example 3.21. *In Example 3.1 we have*

$$
\begin{aligned}
&OUT_1 = \{\mathit{allergy\_strong\_ab}\},\\
&OUT_2 = \{\mathit{xray\_pneumonia}, \mathit{blood\_marker}\},\\
&OUT_3 = \{d : \mathit{BacterialDisease}, d : \mathit{AtypPneumonia}\}, \text{ and}\\
&OUT_4 = \emptyset.
\end{aligned}
$$

Note that no bridge rule contains a belief at context C_4, hence $OUT_4 = \emptyset$.

Using the notion of output beliefs, we let $S_i^o = S_i \cap OUT_i$ be the projection of S_i to OUT_i, and for any belief state $S = (S_1, \ldots, S_n)$ we let $S^o = (S_1^o, \ldots, S_n^o)$ be the *output-projected belief state* S^o of S.

An output-projected belief state provides sufficient information for evaluating the applicability of bridge rules. We next show how to obtain witnesses for equilibria using this projection.

Definition 3.10. *An output-projected equilibrium of an MCS M is an output-projected belief state $T = (T_1, \ldots, T_n)$ such that for all $1 \leq i \leq n$,*

$$T_i \in \left\{S_i^o \mid S_i \in \mathbf{ACC}_i(kb_i \cup \{\varphi(r) \mid r \in app(br_i, T_i)\})\right\}.$$

Here T contains information about all (and only about) output beliefs. As these are the beliefs that determine bridge rule applicability, in every equilibrium S, $app(R, S) = app(R, S^o)$; thus we obtain:

Lemma 3.7 (cf. [117])**.** *For each equilibrium S of an MCS M, S^o is an output-projected equilibrium. Conversely, for each output-projected equilibrium T of M, there exists some equilibrium S of M such that $S^o = T$.*

Given an MCS M, we denote by $\text{EQ}^o(M)$ the set of output-projected equilibria of M.

Example 3.22 (continued)**.** *In our running example, the equilibrium*

$$
\begin{aligned}
S = (&\{\mathit{allergy\_strong\_ab}\}, \{\neg \mathit{blood\_marker}, \mathit{xray\_pneumonia}\},\\
&\{d : \mathit{Pneumonia}, d : \mathit{BacterialDisease}\}, \{\mathit{need\_ab}, \mathit{give\_weak}\})
\end{aligned}
$$

is witnessed by the output-projected equilibrium

$$
\begin{aligned}
S^o = (&\{\mathit{allergy\_strong\_ab}\}, \{\mathit{xray\_pneumonia}\},\\
&\{d : \mathit{Pneumonia}, d : \mathit{BacterialDisease}\}, \emptyset).
\end{aligned}
$$

Here we can observe that, for consistency of the overall system, it is not relevant which belief set is accepted at C_i, only that some belief set is.

Every equilibrium is witnessed by a single output-projected equilibrium, and every output-projected equilibrium witnesses at least one equilibrium.

Context Decomposition

Intuitively, a context is decomposable, if we can partition the head-formulas of its bridge rules, hence the set br_i of bridge rules into br_i^A and br_i^B, and if we further can partition OUT_i into OUT_i^A and OUT_i^B such that there exists knowledge bases kb_i^A, kb_i^B such that every accepted output-projected belief state of the original context corresponds to the union of an accepted output-projected belief state of the two contexts. In formal terms:

Definition 3.11. *Given an MCS $M = (C_1, \ldots, C_n)$, we call $C_n = (L_n, kb_n, br_n)$ with logic $L_n = (\mathbf{BS}_n, \mathbf{KB}_n, \mathbf{ACC}_n)$* decomposable *iff there exist bridge rules $br_n^A, br_n^B \subset br_n$, output beliefs $OUT_n^A, OUT_n^B \subset OUT_n$, and knowledge bases $kb_n^A, kb_n^B \in \mathbf{KB}_n$ such that the following all hold:*

- *$br_n^A \cup br_n^B = br_n$ and $br_n^A \cap br_n^B = \emptyset$,*
- *$\{\varphi(r) \mid r \in br_n^A\} \cap \{\varphi(r) \mid r \in br_n^B\} = \emptyset$,*
- *$OUT_n^A \cup OUT_n^B = OUT_n$ and $OUT_n^A \cap OUT_n^B = \emptyset$, and*
- *for all $H \subseteq \{\varphi(r) \mid r \in br_n\}$ holds:*

$$\begin{gathered}\textit{there exists } S_n \in \mathbf{ACC}_n(kb_n \cup H)\\ \textit{iff}\\ \textit{there exist } S^A \in \mathbf{ACC}_n(kb_n^A \cup (H \cap \{\varphi(r) \mid r \in br_n^A\})) \textit{ and}\\ S^B \in \mathbf{ACC}_n(kb_n^B \cup (H \cap \{\varphi(r) \mid r \in br_n^B\}))\\ \textit{such that } S_n \cap OUT_n = (S^A \cap OUT_n^A) \cup (S^B \cap OUT_n^B)\end{gathered}$$

Notice that this notion only considers the interfacing of C_n to other contexts, it does not consider the internals of the context. So, for a decomposable context C_n its knowledge base kb_n is not required to be decomposable in kb_n^A and kb_n^B. In fact, kb_n^A and kb_n^B may be independent of kb_n and the resulting accepted belief sets are only required to match on the output-projected beliefs, i.e., S^A and S^B may contain additional beliefs not present in S_n. This means that the requirements for a decomposable context are not as strict as they could be; hence the propositions below hold for a more general set of contexts.

In order to realise the decomposition of a (decomposable) context C_n into C_n^A and C_n^B, we remove C_n and introduce contexts C_n^A and C_n^B, i.e., the form of the resulting MCS is $M' = (C_1, \ldots, C_{n-1}, C_n^A, C_n^B)$. Since the beliefs of OUT_n^B are now present only in the context at position $n+1$, all bridge rules of M must be adapted to refer to $n+1$ instead of n, when referring to beliefs of OUT_n^B. Beliefs of OUT_n^A need not be changed since C_n^A already is at position n in M'. To formally express that change we use the mapping $B(i:p)$ which is:

$$B(i:p) = \begin{cases} (n+1:p) & \text{if } i = n \text{ and } p \in OUT_n^B \\ (i:p) & \text{otherwise.} \end{cases}$$

Given a bridge rule of form (2.1), then $B(r)$ is the bridge rule

$$(k : s) \leftarrow B(c_1 : p_1), \ldots, B(c_j : p_j), \mathbf{not}\ B(c_{j+1} : p_{j+1}), \ldots, \mathbf{not}\ B(c_m : p_m).$$

We extend this notion to sets R of bridge rules by $B(R) = \{B(r) \mid r \in R\}$. The formal definition for decomposing an MCS then is as follows:

Definition 3.12. *Let $M = (C_1, \ldots, C_n)$ be an MCS where C_n is a decomposable context wrt. bridge rules $br_n^A, br_n^B \subset br_n$, output beliefs $OUT_n^A, OUT_n^B \subset OUT_n$, and knowledge bases $kb_n^A, kb_n^B \in \mathbf{KB}_n$. Then, the* MCS decomposed wrt. C_n *is $M' = (C'_1, \ldots, C'_{n-1}, C_n^A, C_n^B)$ where*

- *for $1 \leq i \leq n-1$, the context C'_i is $C'_i = (L_i, kb_i, B(br_i))$ where $C_i = (L_i, kb_i, br_i)$,*
- *$C_n^A = (L_n, kb_n^A, B(br_n^A))$, and*
- *$C_n^B = (L_n, kb_n^B, \{(n+1 : \varphi(r)) \leftarrow body(r). \mid r \in B(br_n^B)\})$.*

Observe that there is a one-to-one mapping from bridge rules of C_n in M to the bridge rules of C_n^A and C_n^B, i.e., there exists a bijective mapping s from $br_n(M)$ to $br_n(M') \cup br_{n+1}(M')$ where M' is decomposed wrt. C_n. Since all other bridge rules of M' also stem from exactly one bridge rule of M, we extend s to all bridge rules, formally: $s : br(M) \rightarrow br(M')$ with $s(r) \mapsto \begin{cases} (n+1 : \varphi(r)) \leftarrow body(B(r)). & \text{if } r \in br_n^B \\ B(r) & \text{otherwise.} \end{cases}$

To prove that M and M' behave the same for any diagnosis candidate, we first extend the bijection s on bridge rules to a bijection s' on modified bridge rules. Second, we show that a bridge rule's body is satisfied by a belief state S iff the body of the corresponding bridge rule is satisfied by a corresponding belief state S', i.e., we show that $S \models r$ iff $S' \models s'(r)$ if S' agrees with S wrt. output-projected beliefs and the context decomposition.

Using the bijection s between bridge rules of M and M', we can establish a bijection s' of the two MCS under modifications as follows. Given $D_1, D_2 \subseteq br(M)$, the bijection between bridge rules of $M[D_1, D_2]$ and $M'[s(D_1), s(D_2)]$ is as follows:

$$\begin{aligned} s'(r, r') \text{ holds iff } (r, r') \in \{(r, s(r)) \mid r \in br(M), s(r) \in br(M')\} \setminus \\ \{(r, s(r)) \mid r \in D_1, s(r) \in s(D_1)\} \\ \cup \{(cf(r), cf(s(r))) \mid r \in D_2, s(r) \in s(D_2)\}. \end{aligned}$$

Note that for any $r \in br(M[D_1, D_2])$ it holds that $\varphi(r) = \varphi(s'(r))$.

The following lemma now shows the connection between belief states and applicable bridge rules of M and those of M' given a diagnosis candidate.

Lemma 3.8. *Given an MCS $M = (C_1, \ldots, C_n)$ and the MCS M' decomposed wrt. C_n. Let $S = (S_1, \ldots, S_n)$ be a belief state of M and let $S' = (S'_1, \ldots, S'_{n+1})$ be a belief state of M'. If it holds for all $1 \leq i \leq n-1$ that $S_i \cap OUT_i = S'_i \cap OUT_i$ and if it holds that $S_n \cap OUT_n = (S'_n \cap OUT_n^A) \cup (S'_{n+1} \cap OUT_n^B)$, then*

1. *for any* $r \in br(M[D_1, D_2])$ *holds that* $S \mid\!\leadsto r$ *iff* $S' \mid\!\leadsto s'(r)$*, and*

2. *for any* $1 \leq j \leq n-1$ *holds that*

$$\{\varphi(r) \mid r \in app(br_j(M[D_1, D_2]), S)\} = \{\varphi(r) \mid r \in app(br_j(M'[s(D_1), s(D_2)]), S')\}.$$

Proof. 1. Let $r \in br(M[D_1, D_2])$. Then $r \in D_2$ holds or $r \in br(M)$ holds. In the former case $body(r) = \emptyset$, hence $S \mid\!\leadsto r$ and $body(s'(r)) = \emptyset$, hence $S' \mid\!\leadsto s'(r)$; therefore it also holds that $S \mid\!\leadsto r$ iff $S' \mid\!\leadsto s'(r)$. In the latter case where $r \in br(M)$ holds, we consider an arbitrary $(c : p) \in body^{\pm}(r)$ and we distinguish on the value of c:

- $c \neq n$: Then $B(c : p) = (c : p)$ and since $S_c \cap OUT_c = S'_c \cap OUT_c$, it follows that $S \mid\!\leadsto (c : p)$ iff $S' \mid\!\leadsto B(c : p)$. It follows analogously that $S \mid\!\leadsto \mathbf{not}\ (c : p)$ iff $S' \mid\!\leadsto \mathbf{not}\ B(c : p)$.
- $c = n$: Then either $p \in OUT_n^A$ holds or $p \in OUT_n^B$ holds but not both. Note that $OUT_n^A \cap OUT_n^B = \emptyset$, hence $S_n \cap OUT_n = (S'_n \cap OUT_n^A) \cup (S'_{n+1} \cap OUT_n^B)$ implies that $S_n \cap OUT_n^A = S'_n \cap OUT_n^A$ and $S_n \cap OUT_n^B = S'_{n+1} \cap OUT_n^B$. For $p \in OUT_n^A$ it holds that $B(c : p) = (n : p)$ and $S \mid\!\leadsto (c : p)$ iff $S' \mid\!\leadsto (n : p)$ iff $S' \mid\!\leadsto B(c : p)$, because $S_n \cap OUT_n^A = S'_n \cap OUT_n^A$. Analogously, it holds that $S \mid\!\leadsto \mathbf{not}\ (c : p)$ iff $S' \mid\!\leadsto \mathbf{not}\ B(c : p)$. For $p \in OUT_n^B$ it holds that $B(c : p) = (n+1 : p)$ and $S \mid\!\leadsto (c : p)$ iff $S' \mid\!\leadsto (n+1 : p)$ iff $S' \mid\!\leadsto B(c : p)$, because $S_n \cap OUT_n^B = S'_{n+1} \cap OUT_n^B$. Analogously, it holds that $S \mid\!\leadsto \mathbf{not}\ (c : p)$ iff $S' \mid\!\leadsto \mathbf{not}\ B(c : p)$.

Therefore it holds for all $(c : p) \in body^{\pm}(r)$ that $S \mid\!\leadsto r$ iff $S' \mid\!\leadsto B(r)$ and by the definition of s' and s' it follows that $S \mid\!\leadsto r$ iff $S' \mid\!\leadsto s'(r)$. For all bridge rules $r \in br(M[D_1, D_2])$, it therefore holds that $S \mid\!\leadsto r$ iff $S' \mid\!\leadsto s'(r)$.

2. To show that for all $1 \leq j \leq n-1$ holds

$$\{\varphi(r) \mid r \in app(br_j(M[D_1, D_2]), S)\} = \\ \{\varphi(r) \mid r \in app(br_j(M'[s(D_1), s(D_2)]), S')\},$$

we first observe that $r \in br_j(M[D_1, D_2])$ holds iff $s'(r) \in br_j(M'[s(D_1), s(D_2)])$ holds. Since $S \mid\!\leadsto r$ iff $S' \mid\!\leadsto s'(r)$, it holds that $r \in app(br_j(M[D_1, D_2]), S)$ if and only if it holds that $s'(r) \in app(br_j(M'[s(D_1), s(D_2)]), S')$. Since $\varphi(r) = \varphi(s'(r))$ also holds, it follows immediately that $\{\varphi(r) \mid r \in app(br_j(M[D_1, D_2]), S)\} = \{\varphi(r) \mid r \in app(br_j(M'[s(D_1), s(D_2)]), S')\}$ for all $1 \leq j \leq n-1$.

□

We can now show the output-projected equilibria of M under some modification of its bridge rules are the same as the output-projected equilibria of M' under the same modifications applied to the corresponding bridge rules of M'. Since C_n of M is decomposed into C_n^A and C_n^B, the resulting belief states of M are such that the belief set at the n-th position corresponds in a belief state of M' to the union of the belief sets at the n-th and $n+1$-th position.

Proposition 3.11. *Let M be an MCS with decomposable context C_n, let M' be the MCS decomposed wrt. C_n, and let s be the corresponding bijective mapping $s : br(M) \to br(M')$. Then, for any diagnosis candidate $(D_1, D_2) \in 2^{br(M)} \times 2^{br(M)}$ holds that*

$$\text{EQ}^o(M[D_1, D_2]) = \\ \{(T_1, \ldots, T_{n-1}, T_n \cup T_{n+1}) \mid (T_1, \ldots, T_{n+1}) \in \text{EQ}^o(M'[s(D_1), s(D_2)])\}.$$

Proof. Note that there is a one-to-one mapping between output beliefs of M and those of M', since OUT_i for $1 \leq i \leq n-1$ is the same for M and M', and OUT_n and OUT_{n+1} of M' are a partitioning of OUT_n of M. Thus any output-projected belief state of M corresponds one-to-one to an output-projected belief state of M'; hence the correspondence also holds for $M[D_1, D_2]$ and $M'[s(D_1), s(D_2)]$. In the following, we use this correspondence and we write $T = (T_1, \ldots, T_n)$ to denote a belief state of M and we write $T' = (T'_1, \ldots, T'_n, T'_{n+1})$ to denote the corresponding belief state of M', i.e., it holds that $T_n = T'_n \cup T'_{n+1}$ and $T_i = T'_i$ for all $1 \leq i \leq n-1$.

In the following, we write OUT'_i to denote the set of output-beliefs of context C_i of the MCS M', while OUT_i denotes the output-beliefs of C_i of the MCS M. Note that only OUT_n differs from OUT'_n while for $1 \leq j \leq n-1$ it holds that $OUT_j = OUT'_j$.

We now prove the actual proposition.

"$\subseteq$": Let $T \in \text{EQ}^o(M[D_1, D_2])$ hold. We have to show that $T' \in \text{EQ}^o(M'[s(D_1), s(D_2)])$ holds. First observe that $T \in \text{EQ}^o(M[D_1, D_2])$ implies that there exists an equilibrium $S = (S_1, \ldots, S_n) \in \text{EQ}(M[D_1, D_2])$ such that for all $1 \leq i \leq n$ it holds that $T_i = S_i \cap OUT_i$. Let $H = app(br_n(M[D_1, D_2]), S)$ and note that because C_n is a decomposable context, it holds that there exists $S^A \in \mathbf{ACC}_n(kb^A_n \cup (H \cap \{\varphi(r) \mid r \in br^A_n\}))$ and $S^B \in \mathbf{ACC}_n(kb^B_n \cup (H \cap \{\varphi(r) \mid r \in br^B_n\}))$ such that $S_n \cap OUT_n = (S^A \cap OUT^A_n) \cup (S^B \cap OUT^B_n)$. Furthermore, it holds that $(S^A \cap OUT^A_n) \cup (S^B \cap OUT^B_n) = S_n \cap OUT_n = T_n = T'_n \cup T'_{n+1}$ and since $OUT^A_n \cap OUT^B_n = \emptyset$, it holds that $S^A \cap OUT^A_n = T'_n$ and $S^B \cap OUT^B_n = T'_{n+1}$.

Now consider the belief state $S' = (S'_1, \ldots, S'_{n+1}) = (S_1, \ldots, S_{n-1}, S^A, S^B)$ of M' and observe that $S'_i \cap OUT'_i = T'_i$ holds for all $1 \leq i \leq n+1$. Thus T' is the output-projected belief state wrt. S'. By the construction of S and S', Lemma 3.8 is applicable and it holds for any $1 \leq j \leq n-1$ that $\{\varphi(r) \mid r \in app(br_j(M[D_1, D_2]), S)\} = \{\varphi(r) \mid r \in app(br_j(M'[s(D_1), s(D_2)]), S')$. Since $S \in \text{EQ}([M[D_1, D_2])$ holds, it furthermore follows that that $S_j \in \mathbf{ACC}_j(kb_j \cup \{\varphi(r) \mid r \in app(br_j(M'[s(D_1), s(D_2)]), S')\})$ for all $1 \leq j \leq n-1$. Since $S^A \in \mathbf{ACC}_n(kb^A_n \cup (H \cap \{\varphi(r) \mid r \in br^A_n\}))$ and $S^B \in \mathbf{ACC}_n(kb^B_n \cup (H \cap \{\varphi(r) \mid r \in br^B_n\}))$ both hold, it follows that $S' \in \text{EQ}(M'[s(D_1), s(D_2)]$. Since it holds that $T'_i = S'_i \cap OUT'_i$ for all $1 \leq i \leq n+1$, it thus holds that $T' \in \text{EQ}^o(M'[s(D_1), s(D_2)])$.

"$\supseteq$": Let $T' = (T'_1, \ldots, T'_{n+1}) \in \text{EQ}^o(M'[s(D_1), s(D_2)])$. We have to show that $T = (T'_1, \ldots, T'_{n-1}, T'_n \cup T'_{n+1}) \in \text{EQ}^o(M[D_1, D_2])$. For easier reference, we let $T = (T_1, \ldots, T_n) = (T'_1, \ldots, T'_{n-1}, T'_n \cup T'_{n+1})$. Since $T' \in \text{EQ}^o(M'[s(D_1), s(D_2)])$ holds, there exists $S' = (S'_1, \ldots, S'_{n+1}) \in \text{EQ}(M'[s(D_1), s(D_2)])$ such that $T'_i = S'_i \cap OUT'_i$ with $1 \leq i \leq n+1$. Let $S^A = S'_n$, $S^B = S'_{n+1}$, let

$$H^A = \{\varphi(r) \mid r \in app(br_n(M'[s(D_1), s(D_2)], S')\},$$
$$H^B = \{\varphi(r) \mid r \in app(br_{n+1}(M'[s(D_1), s(D_2)], S')\}, \text{ and}$$

let $H = H^A \cup H^B$. Note that $H^A = H \cap \{\varphi(r) \mid r \in br_n^A\}$ and $H^B = H \cap \{\varphi(r) \mid r \in br_n^B\}$ since M' is the MCS decomposed wrt. the decomposable context C_n, i.e., $\{\varphi(r) \mid r \in br_n^A\} \cap \{\varphi(r) \mid r \in br_n^B\} = \emptyset$. From that it furthermore follows that there exists $S_n \in \mathbf{ACC}_n(kb_n \cup H)$ such that $S_n \cap OUT_n = (S^A \cap OUT_n^A) \cup (S^B \cap OUT_n^B)$.

Now consider the belief state $S = (S_1, \ldots, S_n) = (S'_1, \ldots, S'_{n-1}, S_n)$ which agrees with S' on all but the last belief set. Note that $S_j \cap OUT_j = S'_j \cap OUT'_j$ holds for all $1 \leq j \leq n-1$ and it holds that $S_n \cap OUT_n = (S'_n \cap OUT_n^A) \cup (S'_{n+1} \cap OUT_n^B)$. Therefore Lemma 3.8 applies and it holds for all $r \in br(M[D_1, D_2])$ that $S \models r$ iff $S' \models s'(r)$. Consider the set $R = \{r \in br_n(M[D_1, D_2]) \mid S \models r\}$ of bridge rules of C_n that are applicable in S and let R' be those bridge rules of C'_n and C'_{n+1} that are applicable under S', i.e.,

$$R' = \{r \in br_n(M'[s(D_1), s(D_2)]) \mid S' \models r\} \cup \{r \in br_{n+1}(M'[s(D_1), s(D_2)]) \mid S' \models r\}.$$

We substitute r by $s'(r)$ in the equation and obtain that:

$$R' = \{s'(r) \mid r \in br_n(M[D_1, D_2]), S' \models s'(r)\} \cup \{s'(r) \mid r \in br_n(M[D_1, D_2]), S' \models s'(r)\}.$$

Using $S' \models s'(r)$ iff $S \models r$ we get:

$$R' = \{s'(r) \mid r \in br_n(M[D_1, D_2]), S \models r\} \cup \{s'(r) \mid r \in br_n(M[D_1, D_2]), S \models r\}.$$

Recalling that $R = \{r \in br_n(M[D_1, D_2]) \mid S \models r\}$ thus gives $R' = \{s'(r) \mid r \in R\}$. Since $\varphi(r) = \varphi(s'(r))$ it follows that $\{\varphi(r) \mid r \in R'\} = \{\varphi(r) \mid r \in R\}$, i.e., $H = \{\varphi(r) \mid r \in app(br_n(M[D_1, D_2]), S)\}$. Since $S_n \in \mathbf{ACC}_n(kb_n \cup H)$ holds, it thus holds that $S_n \in \mathbf{ACC}_n(kb_n \cup \{\varphi(r) \mid r \in app(br_n(M[D_1, D_2]), S)\})$, i.e., context C_n accepts the belief set S_n of the belief state S.

Since Lemma 3.8 applies to S and S', it holds that $\{\varphi(r) \mid r \in app(br_j(M[D_1, D_2]), S)\} = \{\varphi(r) \mid r \in app(br_j(M'[s(D_1), s(D_2)]), S')\}$ for all $1 \leq j \leq n$. Consequently, it holds that $S_j \in \mathbf{ACC}_j(kb_j \cup app(br_j(M[D_1, D_2]), S))$ for all $1 \leq j \leq n-1$ and in summary with the above, the same holds for all $1 \leq j \leq n$. Thus $S \in \text{EQ}(M[D_1, D_2])$ holds.

Recall that $S_j \cap OUT_j = T_j = T'_j$ holds for all $1 \leq j \leq n-1$ and note that $S_n \cap OUT_n = (S^A \cap OUT_n^A) \cup S^B \cap OUT_n^B) = T'_n \cup T'_{n+1} = T_n$ since $OUT_n^A \cap OUT_n^B = \emptyset$. Therefore T is the output-projected belief state wrt. S, i.e., $T \in \text{EQ}^o(M[D_1, D_2])$ holds. □

Note that output-projected equilibria are sufficient for computing diagnoses and explanations (cf. Lemma 5.7), hence this notion is sufficient for replacing a context by two others. Also note that the above restriction on just replacing the last context in an MCS can be lifted immediately, since Lemma 3.5 shows that contexts may be re-arranged freely by shuffling (permutation). Since it preserves all equilibria, any context may be decomposed. Furthermore, if one of the contexts resulting from a decomposition is by itself a decomposable context, then this context may be decomposed further. Overall, this allows to decompose a context into arbitrary many other contexts and if some shifting is applied, all contexts of an MCS that are decomposable may be decomposed, possibly multiple times.

Instances of decomposable contexts In the following, we present specific instances of decomposable contexts, hence we show that our previous definition of a decomposable context is not vacuous. We first observe that contexts employing L_{Σ}^{asp} admit a syntactic criterion by which a decomposable context can be recognised. If we consider the undirected dependency graph of the ASP program and this graph contains at least two connected components, then one of the components may induce the decomposition of the context.

Formally, let $C_n = (L_{\Sigma}^{asp}, kb_n, br_n)$ be a context employing ASP. We consider the program P obtained by adding to kb_n all facts (or rules) which may be added by bridge rules, i.e., $P = kb_n \cup \{\varphi(r) \mid r \in br_n\}$. Let A be the set of atoms that occur in P; then the (undirected) dependency graph is $G_P = (A, E)$ where $\{a_1, a_2\} \in E$ holds iff there exists a rule r in P such that a_1 is the head of r and a_2 occurs in the body of r. If there exists a connected component $G' = (A', E')$ such that $G' \neq G_P$, then G' may be used to decompose C_n as follows:

$$
\begin{aligned}
kb_n^A &= \{p \in kb_n \mid \exists a \in A' : a \text{ occurs in } p\} \\
kb_n^B &= \{p \in kb_n \mid \forall a \in A' : a \text{ occurs not in } p\} \\
br_n^A &= \{r \in br_n \mid \exists a \in A' : a \text{ occurs in } \varphi(r)\} \\
br_n^B &= \{r \in br_n \mid \forall a \in A' : a \text{ occurs not in } \varphi(r)\} \\
OUT_n^A &= \{b \in OUT_n \mid b \in A'\} \\
OUT_n^B &= \{b \in OUT_n \mid b \notin A'\}
\end{aligned}
$$

Observe that $br_n^A \cap br_n^B = \emptyset$ and $br_n^A \cup br_n^B = br_n$ as well as $\{\varphi(r) \mid r \in br_n^A\} \cap \{\varphi(r) \mid r \in br_n^A\} = \emptyset$. It also holds that $OUT_n^A \cap OUT_n^B = \emptyset$ and $OUT_n^A \cup OUT_n^B = OUT_n$. It remains to show that for any $H \subseteq \{\varphi(r) \mid r \in br_n\}$ it holds that there exists $S_n \in \mathbf{ACC}_n(kb_n \cup H)$ iff there exist $S^A \in \mathbf{ACC}_n(kb_n^A \cup (H \cap \{\varphi(r) \mid r \in br_n^A\}))$ and $S^B \in \mathbf{ACC}_n(kb_n^B \cup (H \cap \{\varphi(r) \mid r \in br_n^B\}))$ such that $S_n \cap OUT_n = (S^A \cap OUT_n^A) \cup (S^A \cap OUT_n^B)$. Here, an even stronger property holds, namely that $S_n = S^A \cup S^B$, i.e., every answer-set of C_n is the union of an answer set of C_n^A and C_n^B. For that, we note that A' is a splitting set (cf. [96]) and $A \setminus A'$ is a splitting set as well for $P \cup H$ for any $H \subseteq \{\varphi(r) \mid r \in br_n\}$. Following the Splitting Set Theorem of [96] it holds that S_n is an answer-set of $P \cup H$ if $S_n = S^A \cup S^B$ where $\langle S^A, S^B \rangle$ is a so-called solution to $P \cup H$ with respect to A'. The pair is a solution, iff S^A is an answer-set for the bottom relative to A', which is $kb_n^A \cup (H \cap \{\varphi(r) \mid r \in br_n^A\})$, and if S^B is an answer-set to the remainder relative to S^A. In the terminology of [96], this is $b_{A'}(P \cup H \setminus b_{A'}(P \cup H), S^A)$ where $b_{A'}(P \cup H)$ is the bottom relative to A'. Since $A \setminus A'$ also is a splitting set of $P \cup H$, it holds that the remainder is not changed by S^A, i.e., $b_{A'}(P \cup H \setminus b_{A'}(P \cup H), S^A) = kb_n^B \cup (H \cap \{\varphi(r) \mid r \in br_n^B\})$. In other words, S_n is an answer-set of $kb_n \cup H$ iff S^A is an answer-set of $kb_n^A \cup (H \cap \{\varphi(r) \mid r \in br_n^A\})$ and S^B is an answer-set of $kb_n^B \cup (H \cap \{\varphi(r) \mid r \in br_n^B\})$. By the employed logic L_{Σ}^{asp}, S is an answer-set iff if S is an accepted belief set, hence this shows that the above condition of connected components in the dependency graph indeed identifies a decomposable context.

Context complexity $\mathcal{CC}(M)$	Consistency checking MCSEQ	$(A,B) \overset{?}{\in}$ $D^{\pm}(M)$ MCSD	$D^{\pm}_m(M)$ MCSD$_m$	$E^{\pm}(M)$ MCSE	$E^{\pm}_m(M)$ MCSE$_m$
$\mathbf{P}$	$\mathbf{NP}$	$\mathbf{NP}$	$\mathbf{D^P_1}$	$\mathbf{coNP}$	$\mathbf{D^P_1}$
$\mathbf{NP}$	$\mathbf{NP}$	$\mathbf{NP}$	$\mathbf{D^P_1}$	$\mathbf{coNP}$	$\mathbf{D^P_1}$
$\mathbf{\Sigma^P_i}$, $i \geq 1$	$\mathbf{\Sigma^P_i}$	$\mathbf{\Sigma^P_i}$	$\mathbf{D^P_i}$	$\mathbf{\Pi^P_i}$	$\mathbf{D^P_i}$
$\mathbf{PSPACE}$	$\mathbf{PSPACE}$				
$\mathbf{EXPTIME}$	$\mathbf{EXPTIME}$				

Table 3.1: Complexity of consistency checking and recognising (minimal) diagnoses and explanations, given (A, B) and an MCS M for complexity classes of typical knowledge-representation formalisms. Membership holds for all cases, completeness holds if at least one context is complete for the respective context complexity (cf. [117]).

3.4 Computational Complexity

This sections states the computational complexity of various important decision problems regarding inconsistency in MCS. The complexity results are shown here to give a full picture of our basic notions and they form the basis for the complexity analysis of extended notions in later chapters. Detailed results as well as proofs are shown in [54] and [117].

We next consider the complexity of consistency checking, and of diagnosis and explanation recognition in MCS in a parametric fashion. To this end, we recall the complexity classes that we will use, and show that we can abstract an MCS to beliefs used in bridge rules. We use *context complexity* as a parameter to characterise the overall complexity and we establish for hardness generic results for all complexity classes that are closed under conjunction and projection. Table 3.1 summarises our results for complexity classes that are typically encountered in knowledge representation.

Complexity Classes

Recall that $\mathbf{P}$, $\mathbf{EXPTIME}$, and $\mathbf{PSPACE}$ are the classes of problems that can be decided using a deterministic Turing machine in polynomial time, exponential time, and polynomial space, respectively. Furthermore $\mathbf{NP}$ (resp., $\mathbf{coNP}$) is the class of problems that can be decided on a non-deterministic Turing machine in polynomial time, where one (resp., all) execution paths accept. Recall the polynomial hierarchy, where $\mathbf{\Sigma^P_0} = \mathbf{\Pi^P_0} = \mathbf{P}$, $\mathbf{\Sigma^P_i}$ is $\mathbf{NP}$ with a $\mathbf{\Sigma^P_{i-1}}$ oracle, and $\mathbf{\Pi^P_i}$ is $\mathbf{coNP}$ with a $\mathbf{\Sigma^P_{i-1}}$ oracle.

Given complexity class C, we denote by $\mathbf{D}(C)$ the "difference class" of C, i.e., $\mathbf{D}(C) = \{L_1 \times L_2 \mid L_1 \in C,\ L_2 \in \mathbf{co}\text{-}C\}$ denotes the complexity class of decision problems that are the "conjunction" of a problem L_1 in C and a problem L_2 in $\mathbf{co}$-C. For example, $\mathbf{D}(\mathbf{NP}) = \mathbf{D_1^P}$ and $\mathbf{D}(\mathbf{\Sigma_i^P}) = \mathbf{D_i^P}$. A prototypical problem complete for $\mathbf{D_1^P}$ is deciding, given a pair (F_1, F_2) of propositional Boolean formulas, where F_1 is satisfiable and F_2 is unsatisfiable. Note in particular that $\mathbf{D}(\mathbf{PSPACE}) = \mathbf{PSPACE}$ and that $\mathbf{D}(\mathbf{EXPTIME}) = \mathbf{EXPTIME}$.

Closure under Conjunction and Projection A complexity class C is *closed under conjunction*, if the following holds: given a problem L in C, it holds that the problem L^n where L^n is the n-fold Cartesian product of L, and $I = (I_1, \ldots, I_n)$ is a 'yes' instance of L^n iff every instance I_j, $1 \leq j \leq n$ is a 'yes' instances of L, is such that $\bigcup_{n \geq 1} L^n$ is also a problem in C.

All classes $\mathbf{P}$, $\mathbf{NP}$, $\mathbf{\Sigma_i^P}$, $\mathbf{\Pi_i^P}$, $D(\mathbf{\Sigma_i^P})$, $\mathbf{PSPACE}$, etc. here are closed under conjunction.

A decision problem $L \subseteq \Sigma^\star \times \Sigma^\star$ is *polynomially balanced*, if some polynomial p exists such that $|I'| \leq p(|I|)$ for all $(I, I') \in L$. Moreover, L is a *polynomial projection* of $L' \subseteq \Sigma^\star \times \Sigma^\star$ if $L = \{I \mid \exists I' : (I, I') \in L'\}$ and L' is polynomially balanced (intuitively, I' is a witness of polynomial size for I). Given a complexity class C, let $\pi(C)$ contain all problems which are a polynomial projection of a problem L' in C. Then a complexity class C is *closed under projection* if $\pi(C) \subseteq C$.

The classes $\mathbf{\Sigma_i^P}$, $\mathbf{NP}$, $\mathbf{EXPTIME}$, $\mathbf{PSPACE}$ are closed under projection, while $\mathbf{coNP}$ and $\mathbf{\Pi_i^P}$ are presumably not. For further background see [104].

Context Complexity

The complexity of consistency checking for an MCS clearly depends on the complexity of its contexts. We next define a notion of *context complexity* by considering the roles which contexts play in the problem of consistency checking.

For all complexity considerations, we represent logics L_i of contexts C_i *implicitly*; they are fixed and we do not consider these (possibly infinite) objects to be part of the input of the decision problems we investigate. Accordingly, the instance size of a given MCS M will be denoted by $|M| = |kb_M| + |br(M)|$ where $|kb_M|$ denotes the size of knowledge bases in M and $|br(M)|$ denotes the size of its set of bridge rules. Recall that for consistency checking (i.e., equilibrium existence) it is sufficient to consider output-projected equilibria.

Consistency of an MCS M can be decided by a Turing machine with input M which (a) guesses an output-projected belief state $S^o \in OUT_1 \times \cdots \times OUT_n$, (b) evaluates the bridge rules on S^o, yielding for each context C_i a set of active bridge rule heads H_i wrt. S^o, and (c) checks for each context whether it accepts the guessed S_i^o wrt. H_i. We call the complexity of step (c) *context complexity*, formalised as follows.

Definition 3.13. *Given a context $C_i = (kb_i, br_i, L_i)$ and a pair (H, T_i), with $H \subseteq IN_i$ and $T_i \subseteq OUT_i$, the* context complexity $\mathcal{CC}(C_i)$ *of C_i is the computational complexity of deciding whether there exists an $S_i \in \mathbf{ACC}_i(kb_i \cup H)$ such that $S_i \cap OUT_i = T_i$.*

Example 3.23. *Contexts with propositional logic L_Σ^c (see Example 2.1) have $\mathbf{D_1^P}$-complete context complexity, while the restricted logic L_Σ^{pl}, that is used in our running example for contexts C_1 and C_2 (see Example 3.1), is tractable; more precisely, the context complexity is $\mathcal{O}(n)$.*

A context that captures a propositional answer set program is complete for $\mathbf{NP}$ [42].

Default Logic programs and disjunctive logic programs (cf. Example 2.5) have $\mathbf{\Sigma_2^P}$*-complete acceptability checking and thus complexity [42, 78].*

For contexts hosting ontological reasoning in the Description Logic $\mathcal{ALC}$ *(see Example 2.3), the logic* $L_\Sigma^{\mathcal{ALC}}$ *can be used with context complexity* $\mathbf{EXPTIME}$*.*

Given an MCS M, we say M has *upper context complexity* C, denoted $\mathcal{CC}(M) \leq C$, if $\mathcal{CC}(C_i) \subseteq C$ for every context C_i of M; We say M has *lower context complexity* C, denoted $\mathcal{CC}(M) \geq C$, if $C \subseteq \mathcal{CC}(C_i)$ for some context C_i of M. We say that M has *context complexity* C, denoted $\mathcal{CC}(M) = C$, iff $\mathcal{CC}(M) \leq C$ and $\mathcal{CC}(M) \geq C$. That is, if $\mathcal{CC}(M) = C$ all contexts in M have complexity at most C, and some context in M has C-complete complexity, provided the class C has complete problems.

Example 3.24 (continued)**.** *In our running example, for* $M = (C_1, C_2, C_3, C_4)$ *we have the following context complexities:* $\mathcal{CC}(C_1) = \mathcal{CC}(C_2) = \mathcal{O}(n)$*,* $\mathcal{CC}(C_3) = \mathbf{EXPTIME}$*, and* $\mathcal{CC}(C_4) = \mathbf{\Sigma_2^P}$*. As* $\mathcal{O}(n) \subseteq \mathbf{\Sigma_2^P} \subseteq \mathbf{EXPTIME}$*, we obtain* $\mathcal{CC}(M) \leq \mathbf{EXPTIME}$*, and as* C_2 *is* $\mathbf{EXPTIME}$*-complete, we obtain* $\mathcal{CC}(M) \geq \mathbf{EXPTIME}$*; hence* $\mathcal{CC}(M) = \mathbf{EXPTIME}$*.*

Complexity Results (cf. [117])

We consider the decision problem for consistency (MCSEQ) and recognition problems for diagnoses (MCSD), minimal diagnoses (MCSD$_m$), explanations (MCSE), and minimal explanations (MCSE$_m$). Note that *existence* of diagnoses and explanations is trivial by our basic assumptions that M is inconsistent and that $M[\emptyset]$ is consistent.

Table 3.1 summarises the results for context complexities that are present in typical monotonic and nonmonotonic KR formalisms.

For a given context complexity $\mathcal{CC}(M)$ of an MCS M, MCSEQ has the same computational complexity as MCSD. If the context complexity is $\mathbf{NP}$ or above, this complexity is equal to context complexity; for context complexity $\mathbf{P}$, it is $\mathbf{NP}$. Intuitively, this is explained as follows. For context complexity $\mathbf{NP}$ and above, guessing a belief state and checking whether it is an equilibrium can be incorporated into context complexity without exceeding checking cost; if the context complexity is $\mathbf{P}$, this complexity is $\mathbf{NP}$.

Recognising minimal diagnoses MCSD$_m$ is complete for the complexity of MCSD, which captures diagnosis recognition, and an additional complementary problem of refuting MCSD, which captures diagnosis minimality recognition. For context complexity $\mathbf{P}$ it holds that MCSD$_m$ is $\mathbf{D^P}$-complete.

The complexity of MCSE is in the complementary class of the corresponding problem MCSD. Intuitively this is because diagnosis involves existential quantification and explanation involves universal quantification. Accordingly, the complexity of MCSE$_m$ is complementary to MCSD$_m$. As the complexity classes of MCSD$_m$ are closed under complement, MCSE$_m$ and MCSD$_m$ have the same complexity.

These results show that minimal diagnosis and minimal explanation recognition are harder than checking consistency (under usual complexity assumptions), while they are polynomially reducible to each other.

3.5 Computation

In this section, we show how to compute explanations for MCS using HEX-programs. This approach is implemented in the tool MCS-IE,[1] which is an open source experimental prototype.

First we recall HEX-programs, which extend answer set programs, then show how to compute explanations, and finally give an overview of the MCS-IE tool.

Preliminaries: HEX-Programs

HEX-programs [58, 59] extend disjunctive logic programs by allowing for access to external information with *external atoms*, and by *predicate variables*.

In this thesis, we only use ground (variable-free) HEX-programs and thus recall simplified definitions.

Syntax Let $\mathcal{C}$ and $\mathcal{G}$ be mutually disjoint sets of *constants* and *external predicate names*, respectively. Elements from $\mathcal{G}$ are prefixed with "&".

An *ordinary atom* is a formula $p(c_1, \ldots, c_n)$ where $p, c_1, \ldots, c_n$ are constants. An *external atom* is a formula $\&g[\vec{v}](\vec{w})$, where $\vec{v} = Y_1, \ldots, Y_n$ and $\vec{w} = X_1, \ldots, X_m$ are two lists of constants (called *input* and *output* lists, respectively), and $\&g \in \mathcal{G}$ is an external predicate name. Intuitively, an external atom provides a way for deciding the truth value of tuple $\vec{w}$ depending on the extension of input predicates $\vec{v}$.

A HEX *rule* r is of the form

$$\alpha_1 \vee \ldots \vee \alpha_k \leftarrow \beta_1, \ldots, \beta_m, not\, \beta_{m+1}, \ldots, not\, \beta_n \qquad m, k \geq 0, \tag{3.2}$$

where all α_i are ordinary atoms and all β_j are ordinary or external atoms. Rule r is a *constraint*, if $k = 0$; it is a *fact* if $n = 0$ (in this case we omit $\leftarrow$). A HEX*-program* (or *program*) is a finite set of HEX rules: it is *ordinary*, if it contains only ordinary atoms.

Semantics The (ordinary) *Herbrand base* HB_P^o of a HEX-program P is the set of all ordinary atoms $p(c_1, \ldots, c_n)$ occurring in P. An *interpretation* I of P is any subset $I \subseteq HB_P^o$; I satisfies (is a *model* of)

- an atom α, denoted $I \models \alpha$, if $\alpha \in I$ for an ordinary atom α, or if $f_{\&g}(I, \vec{v}, \vec{w}) = 1$ in the case where $\alpha = \&g[\vec{v}](\vec{w})$ and $f_{\&g} : 2^{HB_P^o} \times \mathcal{C}^n \times \mathcal{C}^m \to \{0, 1\}$ is a (fixed) $(|\vec{v}|+|\vec{w}|+1)$-ary Boolean function associated with $\&g$;
- a rule r of form (3.2) ($I \models r$), if either $I \models \alpha_i$ for some α_i, or $I \models \beta_j$ for some $j \in \{m+1, \ldots, n\}$, or $I \not\models \beta_i$ for some $i \in \{1, \ldots, m\}$;
- a program P ($I \models P$), iff $I \models r$ for all $r \in P$.

[1] http://www.kr.tuwien.ac.at/research/systems/mcsie/

The *FLP-reduct* [64] of a program P wrt. an interpretation I is the set $fP^I \subseteq P$ of all rules r of form (3.2) in P such that $I \models \beta_i$, for all $i \in \{1, \ldots, m\}$ and $I \not\models \beta_j$ for all $j \in \{m+1, \ldots, n\}$ (i.e., I satisfies the body of (3.2)). Then, I is an *answer set of* P iff I is a $\subseteq$-minimal model of fP^I. We denote by $\mathcal{AS}(P)$ the collection of all answer sets of P.

For P without external atoms, this coincides with answer sets as in [72], for a discussion on the relation between FLP-reduct and GL-reduct see [64]. HEX programs can be evaluated using the dlvhex solver. [2] A detailed comparison of HEX programs and MCS, showing similarities and differences, is given in [48].

To check whether a context accepts a given belief state under a given knowledge base, we create an external atom $\&con\_out_i[pres_i, b_i]()$ which computes $\mathbf{ACC}_i$ in an external computation. This external atom returns true iff context C_i, when given $B_i(I)$, accepts a belief set S_i such that its projection to output-beliefs OUT_i is equal to $A_i(I)$. Formally,

$$f_{\&con\_out_i}(I, pres_i, in_i) = 1 \text{ iff } A_i(I) \in \{S_i^o \mid S_i \in \mathbf{ACC}_i(kb_i \cup B_i(I))\}.$$

Computing Explanations

We now address the computation of explanations and present an encoding in HEX. Given the conversion results in Section 3.3, explanations either can be computed from the set of diagnoses, or directly by a suitable encoding. In [54] two encodings in HEX are given, one to compute diagnoses and the other to compute explanations. We only show the latter here, since the author of this thesis only is involved in this one.

To identify a diagnosis, it is only necessary to find an equilibrium, while identifying an explanation requires the absence of an equilibrium and it requires that the inconsistency is not irrelevant, i.e, all "relevant pairs" (R_1, R_2), such that $E_1 \subseteq R_1$ and $R_2 \subseteq br(M) \setminus E_2$, must yield an inconsistent system. This is also mirrored by the computational complexity, since the computational complexity of diagnosis recognition is not the same as the one for explanation recognition. For context complexity $\mathcal{CC}(M)$ being $\mathbf{P}$ it is $\mathbf{NP}$ for diagnosis recognition versus $\mathbf{coNP}$ for explanation recognition. Such a check can be realised in HEX, but an involved encoding is needed.

To formally capture the "relevant pairs" of an explanation candidate, we recall the notion of explanation range, which so far has only been used in proofs: given an explanation candidate $E = (E_1, E_2) \in 2^{br(M)} \times 2^{br(M)}$, the *explanation range* of E is

$$Rg(E) = \{(R_1, R_2) \mid E_1 \subseteq R_1 \subseteq br(M) \text{ and } R_2 \subseteq br(M) \setminus E_2\}$$

It follows directly from Definition 3.3 that, $E = (E_1, E_2) \in E^{\pm}(M)$ iff $M[R_1 \cup cf(R_2)] \models \bot$ for all $(R_1, R_2) \in Rg(E)$.

We check explanations by utilising the saturation technique (cf. [56, 94]) from answer-set programming. The underlying idea of saturation hinges on the following observation: a model I of a HEX-program P is an answer-set only if it is a $\subseteq$-minimal model of fP^I. To check whether a forall-statement holds, first an interpretation $I^\star$ is designed such that $I^\star$ is a $\subseteq$-maximal model of $fP^{I^\star}$; second, using disjunctive rules, a model I representing a possible counter-example

[2] http://www.kr.tuwien.ac.at/research/systems/dlvhex/

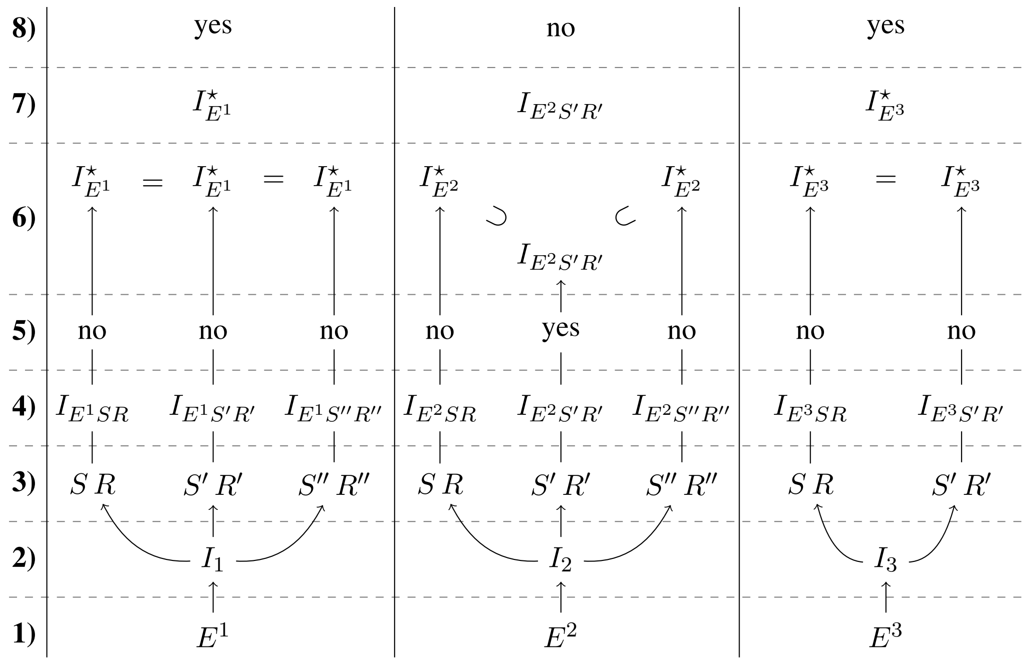


Figure 3.8: Saturation conceptual overview with three explanation candidates E^1, E^2, and E^3. Steps are: **1)** Guess explanation candidate; **2)** Partial interpretation corresponding to guess; **3)** Guess belief state S and bridge rules $R = (R_1, R_2) \in Rg(E^j)$; **4)** Partial interpretation I_{E^jSR} corresponding to guesses of E^j, S, R; **5)** Is S equilibrium of $M[R_1 \cup cf(R_2)]$; **6)** Resulting model, saturated if S is no equilibrium; **7)** Select $\subseteq$-minimal model; **8)** Answer sets (saturated minimal models): $I^\star_{E^1}$ and $I^\star_{E^3}$.

is guessed such that $I \subseteq I^\star$ holds. The logic program P is designed such that if I indeed is a counter-example to the forall-statement, then I is a model of the reduct $fP^{I^\star}$ of P wrt. $I^\star$; therefore $I \subseteq I^\star$ holds and it follows that $I^\star$ is not a $\subseteq$-minimal model of $fP^{I^\star}$, which ensures that $I^\star$ is not an answer-set of P (although $I^\star$ still is a model of $fP^{I^\star}$). On the other hand, if no counter-example exists, then no $I \subseteq I^\star$ exists such that I is a model of $fP^{I^\star}$ and $I^\star$ is a $\subseteq$-minimal model of $fP^{I^\star}$.

Figure 3.8 gives a conceptual and exemplary overview of the saturation technique. It shows three explanation candidates and the conceptual steps and partial interpretations to derive whether a candidate indeed is an explanation. There are eight steps, which are as follows for a given MCS M:

1) An explanation candidate $E = (E_1, E_2)$ is guessed.

2) Each candidate corresponds to a partial interpretation I_E.

3) A "relevant pair" $R = (R_1, R_2) \in Rg(E)$ is guessed together with a belief set S for $M[R_1 \cup cf(R_2)]$.

4) The result is a partial interpretation I_{ESR} corresponding to the guessed sets above.

5) It is checked whether I encodes an equilibrium, i.e., whether for each $C_i \in C\,(M)$ holds $S_i \in \mathbf{ACC}_i(kb_i \cup \{\varphi\,(r) \mid r \in app(br_i(M[R_1 \cup cf\,(R_2)]), S)\})$.

6) If the previous condition does not hold for every $C_i \in C\,(M)$, then I_{ESR} encodes not an equilibrium and I_{ESR} is saturated with respect to those atoms encoding S and R.

   Saturation in more detail works as follows: any guess of S and R is encoded by two atoms, for example the statement whether a bridge rule $r \in br(M)$ is contained in R_1 is encoded via two atoms $r1\,(r)$ and $nr1\,(r)$. For an interpretation I the statement that $r \in R_1$ holds is encoded by the condition that $r1\,(r) \in I$ and $nr1\,(r) \notin I$; the statement $r \notin R_1$ is encoded by the condition that $r1\,(r) \notin I$ and $nr1\,(r) \in I$. Saturation then makes both atoms true, i.e., if I is saturated, then, I $r1\,(r) \in I$ and $nr1\,(r) \in I$ both hold. Observe that from the perspective of HEX this is not a contradiction and it holds that the saturated interpretation is a superset of both possibilities regarding the containment of r in R_1.

   This saturated interpretation, denoted by $I^{\star}_{E}$, is a maximum interpretation wrt. the guess for E. Conversely, if S is an equilibrium, then I is not saturated which means that the resulting interpretation I is a subset of $I^{\star}_{E}$.

7) Following the semantics of HEX, a minimal model is selected. Since $I^{\star}_{E}$ is a maximum model, it will only be the minimal model, if there is no other $I \subset I^{\star}_{E}$.

8) A constraint finally ensures that only a saturated interpretation $I^{\star}_{E}$ is an answer-set. Hence only an explanation candidate such that in its range no equilibrium exists admits an answer-set.

In the following we present a direct encoding, $P^{E}_{p}(M)$, in HEX using saturation. We first guess an explanation candidate $E = (E_1, E_2)$ and then ensure via saturation, that for all pairs of sets $(R_1, R_2) \in Rg(E)$ the modified system is inconsistent, i.e., we check for every $(R_1, R_2) \in Rg(E)$ and for every belief state S, that some context of $M[R_1 \cup cf\,(R_2)]$ does not accept S.

For all $r \in br(M)$, $P^{E}_{p}(M)$ contains the following rules to guess an explanation candidate.

$$e1\,(r) \vee ne1\,(r). \tag{3.3}$$

$$e2\,(r) \vee ne2\,(r). \tag{3.4}$$

To give some intuition of the saturation technique, assume that I is the (partial) interpretation corresponding to an explanation candidate guessed by the above rules. To check that every $(R_1, R_2) \in Rg(E)$ yields an inconsistent system, saturation is used as follows: via disjunctive rules, $(R_1, R_2) \in Rg(E)$ is guessed as well as a belief state S. If S is not an equilibrium for $M[R_1 \cup cf\,(R_2)]$, then the atom *spoil* is concluded to be true. This in turn leads to the truth of all other atoms that occur in rules to guess R_1, R_2, S, and all other atoms that are necessary to check that S is not an equilibrium. The resulting interpretation, $I^{\star}$, is said to be saturated (or spoiled);

formally, it contains I_{spoil}, which is given by:

$$\begin{aligned} I_{spoil} = &\{r1(r), nr1(r), r2(r), nr2(r), brbody(r) \mid r \in br(M)\} \cup \\ &\{in_i(b) \mid r \in br(M) \wedge C_h(r) = i \wedge \varphi(r) = b\} \cup \{spoil\} \cup \\ &\bigcup_{a \in OUT_i} \{pres_i(a), abs_i(a)\} \cup \bigcup_{b \in IN_i} \{in_i(b)\}. \end{aligned}$$

Most importantly, $I^\star$ is a maximal model of $fP_p^E(M)^I$ and every other guess for (R_1, R_2) and S will result in the same interpretation $I^\star$, if S is not an equilibrium of $M[R_1 \cup cf(R_2)]$.

On the other hand, if there is a guess for (R_1, R_2) and S such that S is an equilibrium of $M[R_1 \cup cf(R_2)]$, then the corresponding interpretation I' will not be saturated. Since $I^\star$ is a maximal model, it then holds that $I' \subset I^\star$, hence $I^\star$ is not a minimal model of $fP_p^E(M)^I$. Thus, if $I^\star$ is indeed the minimal model of $fP_p^E(M)^I$, then there can not exist such an I', i.e., for all (R_1, R_2) and S it then holds that S is not an equilibrium of $M[R_1 \cup cf(R_2)]$.

Since we are only interested in explanation candidates E where no equilibrium exists for any $(R_1, R_2) \in Rg(E)$, a constraint is added to ensure that only saturated models comprise an answer set, i.e, we ensure that only $I^\star$ may yield an answer set.

To generate $(R_1, R_2) \in Rg(E)$, for every $r \in br(M)$ the following rules are in $P_p^E(M)$:

$$r1(r) :- e1(r). \tag{3.5}$$

$$r1(r) \vee nr1(r) :- ne1(r). \tag{3.6}$$

$$nr2(r) :- e2(r). \tag{3.7}$$

$$r2(r) \vee nr2(r) :- ne2(r). \tag{3.8}$$

We further guess a belief state of M, so $P_p^E(M)$ contains for every $a \in OUT_i$ with $1 \leq i \leq n$ the following rule:

$$pres_i(a) \vee abs_i(a). \tag{3.9}$$

Recall that I is an answer set of $P_p^E(M)$ iff I is a $\subseteq$-minimal model of $fP_p^E(M)^I$. As we use saturation and external atoms, this can lead to the undesired effect that some $r \in fP_p^E(M)^I$ is unsupported, i.e., for a being the head of r it can happen that $a \in I$ but the body of r is false under I and no other rule's body with head a is true. To avoid this, each bridge rule of M is encoded such that $a \in I$ implies that a corresponding body also evaluates to true. This is achieved by the addition of a unique atom $brbody(r)$ for each $r \in br(M)$ and further rules ensuring that each literal in the body of r holds if $brbody(r) \in I$. So, $P_p^E(M)$ contains for each $r \in br(M)$ of form $(i : b) \leftarrow (i_1 : b_1), \ldots, (i_{k-1} : b_{k-1}), not(i_k : b_k), \ldots, not(i_m : b_m)$ the following rules:

$$brbody(r) :- r1(r), pres_{i_1}(b_1), \ldots, pres_{i_{k-1}}(b_{k-1}), abs_{i_k}(b_k), \ldots, abs_{i_m}(b_m). \quad (3.10)$$

$$r1(r) :- brbody(r). \quad (3.11)$$

$$pres_{i_1}(b_1) :- brbody(r). \quad (3.12)$$

$$\ldots$$

$$pres_{i_{k-1}}(b_{k-1}) :- brbody(r). \quad (3.13)$$

$$abs_{i_k}(b_k) :- brbody(r). \quad (3.14)$$

$$\ldots$$

$$abs_{i_m}(b_m) :- brbody(r). \quad (3.15)$$

$$in_i(b) :- brbody(r). \quad (3.16)$$

$$in_i(b) :- r2(r). \quad (3.17)$$

Rules (3.16) and (3.17) ensure that the head of r is derived if either the body holds, or if r is unconditional, i.e., $r \in R_2$. For the head $(i : b)$ of r, let $[(i : b)]$ be the set of bridge rules whose head is the same, i.e., $[(i : b)] = \{r \in br(M) \mid C_h(r) = i \wedge \varphi(r) = b\}$. For each head $(i : b)$ of a bridge rule with $[(i : b)] = \{r_1, \ldots, r_k\}$ the following rule of $P_p^E(M)$ ensures that $(i : b)$ is supported:

$$brbody(r_1) \vee \ldots \vee brbody(r_k) \vee r2(r_1) \vee \ldots \vee r2(r_k) :- in_i(b). \quad (3.18)$$

So far $P_p^E(M)$ guesses an explanation candidate E, a pair $(R_1, R_2) \in Rg(E)$, a belief state encoded by $pres$ and abs, and the beliefs of applicable bridge rule heads are computed. To ensure that E is an explanation it must be the case that for every pair (R_1, R_2) and belief state S some context C_i does not accept S_i given the input encoded by in_i. If some context does not accept S_i then a special atom $spoil$ is derived, i.e., if the external atom $\&con\_out'_i[spoil, pres_i, in_i, out_i]()$ is false then $spoil$ is derived. This atom is also derived if the guess of S and (R_1, R_2) is contradictory by itself. So for every $r \in br(M), a \in OUT_i, i \in \{1, \ldots, n\}$ the following rules are in $P_p^E(M)$:

$$spoil :- not\ \&con\_out'_i[spoil, pres_i, in_i](). \quad (3.19)$$

$$spoil :- r1(r), nr1(r). \quad (3.20)$$

$$spoil :- r2(r), nr2(r). \quad (3.21)$$

$$spoil :- pres_i(a), abs_i(a). \quad (3.22)$$

We slightly extend the external atom $\&con\_out_i[pres_i, in_i]()$ for checking consistency of a context: if $spoil$ is present, then the external atom must be false. This is needed, since a spoiled interpretation $I^\star$ must be a model of the HEX program, which is only guaranteed if the external atom is false in $I^\star$. So, $\&con\_out'_i[spoil, pres_i, in_i]()$ is based on $\&con\_out_i[pres_i, in_i]()$ as follows:

$$f_{\&con\_out'_i}(I, spoil, pres_i, in_i) = 0 \text{ iff } f_{\&con\_out_i}(I, pres_i, b_i) = 0 \vee spoil \in I.$$

To saturate all guesses, we add the following rules, for all $r \in br(M), i \in C(M)$, $a \in OUT_i, b \in IN_i$, to $P_p^E(M)$:

$$r1(r) :- spoil. \qquad r2(r) :- spoil. \tag{3.23}$$

$$nr1(r) :- spoil. \qquad nr2(r) :- spoil. \tag{3.24}$$

$$abs_i(a) :- spoil. \qquad pres_i(a) :- spoil. \tag{3.25}$$

$$in_i(b) :- spoil. \qquad brbody(r) :- spoil. \tag{3.26}$$

As an interpretation I of a program P is only an answer set if it is a minimal model of fP^I, it follows that I is not an answer set if there is a model I' of fP^I with $I' \subset I$. If the guess for (R_1, R_2) and the belief state S is not acceptable at context C_i, then *spoil* is derived and saturation takes place, i.e., I' becomes $\subseteq$-maximal. If, however, some guess for (R_1, R_2) and S yields an equilibrium of M, then the corresponding interpretation I' is a subset of the saturated guesses, thus making the explanation candidate no minimal model of its reduct.

To obtain only valid explanations, $P_p^E(M)$ contains the following constraint:

$$:- not\ spoil. \tag{3.27}$$

It ensures that only saturated interpretations $I^\star$ can be answer sets. But it only is a $\subseteq$-minimal model of $fP_p^E(M)^I$, if no $I' \subset I$ exists, i.e., if all $(R_1, R_2) \in Rg(E)$ yield an inconsistent system. For more details on the saturation technique we refer to [60,95].

The answer sets of $P_p^E(M)$ now exactly encode all explanations of the inconsistent MCS M. To prove this in the following, we utilise several lemmas and introduce some notation first.

For the following proofs we assume $M = (C_1, \dots, C_n)$ to be an arbitrary but fixed MCS and $P_p^E(M)$ to be the explanation encoding for M. Given a HEX rule r of form (3.2), we write $B_{\text{HEX}}(r) = \{\beta_1, \dots, \beta_n\}$ and $H_{\text{HEX}}(r) = \{\alpha_1, \dots, \alpha_k\}$ to denote body and head of r respectively. For an interpretation I and a HEX rule r, we write $I \models B_{\text{HEX}}(r)$ iff $I \models \beta_i$ for all $i \in \{1, \dots, m\}$ and $I \not\models \beta_j$ for all $j \in \{m+1, \dots, n\}$. Similarly, we write $I \models H_{\text{HEX}}(r)$ iff $I \models \alpha_i$ for some $i \in \{1, \dots, k\}$.

For referring to a specific rule of $P_p^E(M)$, we write $tr_N(v_1, \dots, v_\ell)$ where N is the rule of form (N) instantiated with $v_1 \dots, v_\ell$. We denote by $TR_n(M)$ the set of all instantiations of a rule wrt. an MCS M. For example, let $r_7 \in br(M)$, then $tr_{3.5}(r_7)$ denotes the HEX rule $r1(r_7) :- e1(r_7).$, while $TR_{3.5}(M) = \{tr_{3.5}(r) \mid r \in br(M)\}$. For brevity, we write only those values necessary to identify the instantiation, e.g., for rules of form (3.10) we write $tr_{3.10}(r)$ where $r \in br(M)$; for a rule of form (3.18), we write $tr_{3.18}(i, b)$ where $(i : b)$ is the head of some $r \in br(M)$.

We say an interpretation I *consistently encodes* an explanation candidate $E = (E_1, E_2)$ where $E_1 = \{r \in br(M) \mid e1(r) \in I\}$, $E_2 = \{r \in br(M) \mid e2(r) \in I\}$, for all $r \in br(M)$: (i) $e1(r) \in I$ iff $ne1(r) \notin I$, and (ii) $e2(r) \in I$ iff $ne2(r) \notin I$.

Lemma 3.9. *Every answer set I of $P_p^E(M)$ consistently encodes an explanation candidate.*

Proof. Let I be an answer set of $P_p^E(M)$. Then, by definition I must be a minimal model of $fP_p^E(M)^I$. Assume for contradiction that I does not consistently encode an explanation candidate. Then, for some $r \in br(M)$ one of the following cases holds.

(i) $e1(r) \in I$ and $ne1(r) \in I$: Consider $I' = I \setminus \{e1(r)\}$. For all $tr \in fP_p^E(M)^I$ with $e1(r) \notin H_{\text{HEX}}(tr)$ it holds that $I' \models tr$ since $I \models tr$. There is only one rule tr' such that $e1(r) \in H_{\text{HEX}}(tr')$, namely $tr' = tr_{3.3}(r)$. Since $ne1(r) \in I'$ and $ne1(r) \in H_{\text{HEX}}(tr_{3.3}(r))$ it holds that $I' \models tr$, hence $I' \models fP_p^E(M)^I$. Since $I' \subset I$ this contradicts that I is a minimal model of $fP_p^E(M)^I$.

(ii) $e1(r) \notin I$ and $ne1(r) \notin I$. Since $B_{\text{HEX}}(tr_{3.3}(r)) = \emptyset$, it holds that $tr_{3.3}(r) \in fP_p^E(M)^I$ while $I \not\models H_{\text{HEX}}(tr_{3.3}(r))$. Hence, in contradiction to the assumption, it holds that $I \not\models fP_p^E(M)^I$.

(iii) $e2(r) \in I$ and $ne2(r) \in I$: This is similar to case (i), just replace $e1$ by $e2$ and $tr_{3.3}(r)$ by $tr_{3.4}(r)$.

(iv) $e2(r) \notin I$ and $ne2(r) \notin I$: This is similar to case (ii), just replace $e1$ by $e2$ and $tr_{3.3}(r)$ by $tr_{3.4}(r)$.

Since each case yields a contradiction, it follows that I consistently encodes an explanation candidate. □

Lemma 3.10. *If I is an answer set for $P_p^E(M)$ and $E = (E_1, E_2)$ is the explanation candidate consistently encoded by I, then $fP_p^E(M)^I$ exactly contains*

1. $TR_{3.9}(M) \cup \ldots \cup TR_{3.26}(M)$.
2. $\{tr_{3.5}(r) \mid r \in E_1\} \cup \{tr_{3.6}(r) \mid r \in br(M) \setminus E_1\} \cup \{tr_{3.7}(r) \mid r \in E_2\} \cup \{tr_{3.8}(r) \mid r \in br(M) \setminus E_2\}$.

Proof. Let I be an answer set for $P_p^E(M)$ encoding an explanation candidate $E = (E_1, E_2)$.
1. By the constraint rule (3.27), it holds that $spoil \in I$, thus rules $TR_{3.23}(M) \cup \ldots \cup TR_{3.26}(M)$ are in $fP_p^E(M)^I$. Let $tr \in TR_{3.23}(M) \cup \ldots \cup TR_{3.26}(M)$, then it holds that $I \models B_{\text{HEX}}(tr)$, hence it follows that $I \models H_{\text{HEX}}(tr)$. Therefore, $I \models B_{\text{HEX}}(tr')$ and $tr' \in fP_p^E(M)^I$, where $tr' \in TR_{3.9}(M) \cup \ldots \cup TR_{3.22}(M)$. Specifically, it holds for $tr_{3.19}(i)$, where $1 \leq i \leq n$, that $I \models B_{\text{HEX}}(tr_{3.19}(i))$, because $spoil \in I$ which implies that $f_{\&con\_out'_i}(I, spoil, pres_i, in_i) = 0$.
2. Let $r \in E_1$. Then $e1(r) \in I$ and $ne1(r) \notin I$ since I consistently encodes E. Thus, $I \models B_{\text{HEX}}(tr_{3.5}(r))$, therefore $tr_{3.5}(r) \in fP_p^E(M)^I$. Furthermore, $I \not\models B_{\text{HEX}}(tr_{3.6}(r))$, hence $tr_{3.6}(r) \notin fP_p^E(M)^I$.

Let $r \in br(M) \setminus E_1$. Then $e1(r) \notin I$ and $ne1(r) \in I$ since I consistently encodes E. Thus, $I \models B_{\text{HEX}}(tr_{3.6}(r))$, therefore $tr_{3.6}(r) \in fP_p^E(M)^I$. Furthermore, $I \not\models B_{\text{HEX}}(tr_{3.5}(r))$, hence $tr_{3.5}(r) \notin fP_p^E(M)^I$.

The remaining cases for E_2 are analogous. □

Definition 3.14. *An interpretation I of $P_p^E(M)$ is called* contradiction-free *(regarding $r1$, $nr1$, $r2$, $nr2$, $pres_i$, abs_i) if and only if the following conditions hold:*

$$r1(r) \in I \textit{ iff } nr1(r) \notin I \qquad \textit{for every } r \in br(M)$$
$$r2(r) \in I \textit{ iff } nr2(r) \notin I \qquad \textit{for every } r \in br(M)$$
$$pres_i(a) \in I \textit{ iff } abs_i(a) \notin I \qquad \textit{for every } a \in OUT_i, 1 \leq i \leq n$$

We say that a contradiction-free interpretation I consistently encodes *a belief state $S = (S_1, \ldots, S_n)$ and a pair (R_1, R_2) of sets of bridge rules such that: $a \in S_i$ iff $pres_i(a) \in I$, $r \in R_1$ iff $r1(r) \in I$, and $r \in R_2$ iff $r2(r) \in I$.*

Notice that rule (3.27) and Lemma 3.10 ensure that no answer set I of $P_p^E(M)$ is contradiction-free, because it holds that $spoil \in I$ and the rules of $TR_{3.23}(M) \cup \ldots \cup TR_{3.26}(M)$ ensure the saturation of I. The notion, however, is useful for reasoning about (minimal) models of $fP_p^E(M)^I$.

$P_p^E(M)$ guarantees that a contradiction-free interpretation I that encodes a belief state S and a pair (R_1, R_2) of sets of bridge rules also contains a representation of the set of heads of bridge rules applicable under S and (R_1, R_2), as the following lemma shows.

Lemma 3.11. *Let I be a contradiction-free interpretation that encodes the belief state $S = (S_1, \ldots, S_n)$ of M, and let (R_1, R_2) such that $R_1, R_2 \subseteq br(M)$. If I is a minimal model of $P \subseteq P_p^E(M)$ such that $TR_{3.10}(M) \cup \ldots \cup TR_{3.18}(M)$ is a subset of P, then $\{b \in IN_i \mid in_i(b) \in I\} = \{\varphi(r) \mid r \in app(br_i(M[R_1 \cup cf(R_2)]), S)\}$ for every $1 \leq i \leq n$.*

Proof. Recall that, given an MCS M', $[(i : b)]$ denotes the set of bridge rules whose head is $(i : b)$, i.e., $[(i : b)] = \{r \in br(M') \mid C_h(r) = i \wedge \varphi(r) = b\}$.
($\subseteq$): Let $b \in \{b \in IN_i \mid in_i(b) \in I\}$ and le $\{r_1, \ldots, r_k\} = [(i : b)]$ be the set of bridge rules of $M[R_1 \cup cf(R_2)]$ whose head is $(i : b)$. Since $I \models B_{\text{HEX}}(tr_{3.18}(i, b))$, it must hold for some rule r_j with $1 \leq j \leq k$ that $r2(r_j) \in I$ or $brbody(r_j) \in I$.

In the former case it follows that $r_j \in R_2$ and thus $r_j \in app(br_i(M[R_1 \cup cf(R_2)]), S)$, hence $b \in \{\varphi(r) \mid r \in app(br_i(M[R_1 \cup cf(R_2)]), S)\}$.

In the latter case, $brbody(r_j) \in I$ together with rules $tr_{3.12}(r_j), \ldots, tr_{3.15}(r_j)$ implies that each literal in the body of r_j is satisfied by the belief state S. Furthermore, from $I \models tr_{3.11}(r_j)$ it follows that $r1(r_j) \in I$, hence $r_j \in R_1$. Therefore, $r_j \in app(br_i(M[R_1 \cup cf(R_2)]), S)$, hence $b \in \{\varphi(r_j) \mid app(br_i(M[R_1 \cup cf(R_2)]), S)\}$.
($\supseteq$) Let $b \in app(br_i(M[R_1 \cup cf(R_2)]), S)\}$ and let $\{r_1, \ldots, r_k\} = [(i : b)]$ be the bridge rules in $M[R_1 \cup cf(R_2)]$ whose head is $(i : b)$. By definition of applicability, it must hold for some r_j with $1 \leq j \leq k$ that either $r_j \in R_2$ or $r_j \in R_1$ and the body of r_j is satisfied wrt. S. In the former case $r2(r_j) \in I$ and by $tr_{3.17}(r_j)$ it must hold that $in_i(b) \in I$, hence $b \in \{b \in IN_i \mid in_i(b) \in I\}$. In the latter case observe that $S \models r_j$ and as I consistently encodes S and (R_1, R_2), it holds that $I \models B_{\text{HEX}}(tr_{3.10}(r_j))$. Therefore $in_i(b) \in I$, hence $b \in \{b \in IN_i \mid in_i(b) \in I\}$. $\square$

Theorem 3.4. *Let M be an inconsistent MCS. Then $(E_1, E_2) \in E^{\pm}(M)$ iff there exists an answer set I of $P_p^E(M)$ where $E_1 = \{r \mid e1(r) \in I\}$ and $E_2 = \{r \mid e2(r) \in I\}$.*

Recall the concept of a saturated ("spoiled") interpretation. An interpretation is saturated, if it is a superset of I_{spoil}, which is defined as follows:

$$I_{spoil} = \{r1(r), nr1(r), r2(r), nr2(r), brbody(r) \mid r \in br(M)\} \cup \{in_i(b) \mid r \in br(M) \wedge C_h(r) = i \wedge \varphi(r) = b\} \cup \{spoil\} \cup \bigcup_{a \in OUT_i} \{pres_i(a), abs_i(a)\} \cup \bigcup_{b \in IN_i} \{in_i(b)\}.$$

Soundness ($\Leftarrow$). Let I be an answer set of $P_p^E(M)$. Then by Lemma 3.9 I consistently encodes an explanation candidate $E = (E_1, E_2)$ where $E_1 = \{r \in br(M) \mid e1(r) \in I\}$ and $E_2 = \{r \in br(M) \mid e2(r) \in I\}$. We show that E is an explanation of M.

Since I is an answer set of $P_p^E(M)$, it is a minimal model of $fP_p^E(M)^I$ and by Lemma 3.10 all rules of $TR_{3.9}(M) \cup \ldots \cup TR_{3.26}(M)$ are in $fP_p^E(M)^I$, so I must be a minimal model of those rules. By rule (3.27) it follows that $spoil \in I$, therefore for each $tr \in TR_{3.23}(M) \cup \ldots \cup TR_{3.26}(M)$ it holds that $H_{\text{HEX}}(tr) \in I$ since $I \models B_{\text{HEX}}(tr)$. Therefore, $I_{spoil} \subseteq I$.

Towards a contradiction, assume that E is not an explanation. Then, there exists $(R_1, R_2) \in Rg(E)$ such that $M' \not\models \bot$ holds for $M' = M[R_1 \cup cf(R_2)]$, i.e., M' has an equilibrium $S = (S_1, \ldots, S_n)$.

Consider the interpretation $I_{S,(R_1,R_2)}$ corresponding to S and (R_1, R_2), i.e., I' is a contradiction-free interpretation regarding $r1, nr1, r2, nr2, pres_i, abs_i$ that consistently encodes S and (R_1, R_2). Let $I_{S,(R_1,R_2),E} = I_{S,(R_1,R_2)} \cup \{e1(r) \in I\} \cup \{ne1(r) \in I\} \cup \{e2(r) \in I\} \cup \{ne2(r) \in I\}$ be the interpretation consistently encoding E, S, and (R_1, R_2). Finally, let $I_{app} = \{in_i(b) \mid b \in app(br_i(M'), S)\} \cup \{brbody(r) \mid r \in R_1 \wedge S \models r\}$ correspond to the set of bridge rule heads and bodies applicable under S. Combining them, we obtain an interpretation $I' = I_{S,(R_1,R_2),E} \cup I_{app}$. Note that $I' \subset I$, since I is saturated and both I and I' consistently encode E.

As we show in the following, it holds that $I' \models fP_p^E(M)^I$:

- For every $tr \in TR_{3.3}(M) \cup TR_{3.4}(M)$ it holds that $I' \models tr$ since $I \models tr$ and I agrees with I' on atoms $e1, ne1, e2$, and $ne2$.
- For every $tr \in TR_{3.5}(M) \cup \ldots \cup TR_{3.8}(M)$ it holds that $I' \models tr$ since $(R_1, R_2) \in Rg(E)$ and I' consistently encodes (R_1, R_2).
- For every $tr \in TR_{3.9}(M)$ it holds that $I' \models tr$ since I' consistently encodes S.
- For every $r \in br(M)$ it holds that $I' \models tr_{3.10}(r)$ since $I_{app} \subseteq I'$ and I_{app} is defined such that $S \models\!\!\rightsquigarrow r$ and $r \in R_1$ implies that $brbody(r) \in I'$.
- For every $r \in br(M)$ it holds that $I' \models tr_{3.11}(r)$ since $brbody(r) \in I'$ implies $r \in R_1$, hence by I' encoding (R_1, R_2) it follows that $r1(r) \in I'$.
- For every $r \in br(M)$ it holds that $I' \models tr_{3.12}(r), \ldots I' \models tr_{3.15}(r)$, because $brbody(r) \in I'$ only if $S \models r$, hence by I' encoding S the following hold: $H_{\text{HEX}}(tr_{3.12}(r)) \in I', \ldots, H_{\text{HEX}}(tr_{3.15}(r)) \in I'$.
- For every $r \in br(M)$ it holds that $I' \models tr_{3.16}(r)$, respectively $I' \models tr_{3.17}(r)$ since $S \models r$ and $r \in R_1$, respectively $r \in R_2$, implies that $r \in app(br_i(M'), S)$, hence $in_i(b) \in I'$ where $i \in C(M)$ and $\varphi(r) = b$.
- For every head $(i : b)$ of a bridge rule it holds that $I' \models tr_{3.18}(i, b)$, because if $in_i(b) \in I'$ for some $i \in C(M)$, then by definition of I' there exists $r \in app(br_i(M'), S)$ such that one of the following holds:
  - $S \models r$ and $r \in R_1$, which implies that $brbody(r) \in I'$.

- $r \in R_2$ and therefore $r2(r) \in I'$.

- $I' \models tr_{3.19}(i)$ holds for all $1 \leq i \leq n$: By definition of I_{app}, it holds that $\{b \mid in_i(b) \in I'\} = app(br_i(M'), S)$ and since I' encodes S, it also it holds that $\{a \mid pres_i(a)\} = S_i$. By assumption S is an equilibrium of M', hence $S_i \in \mathbf{ACC}_i(app(br_i(M'), S))$. Therefore, $f_{\&con\_out'_i}(I', pres_i, in_i) = 1$ and $I' \not\models B_{\text{HEX}}(tr_{3.19}(i))$.

- For every $tr \in TR_{3.20}(M) \cup \ldots \cup TR_{3.22}(M)$ it holds that $I' \models tr$ since I' is conflict-free and $I' \not\models B_{\text{HEX}}(tr)$.

- For every $tr \in TR_{3.23}(M) \cup \ldots \cup TR_{3.26}(M)$ it holds that $I' \models tr$ since $spoil \notin I'$.

- Rule (3.27): is not in the reduct $fP_p^E(M)^I$, hence it needs not be satisfied by I'.

Therefore, all rules of $fP_p^E(M)^I$ are satisfied and it follows that I' is a model of $fP_p^E(M)^I$. Since $I' \subset I$, I is not a minimal model of $fP_p^E(M)^I$, which contradicts that I is an answer set of $P_p^E(M)$. This proves that $E \in E^{\pm}(M)$. □

Completeness ($\Rightarrow$). Let $E = (E_1, E_2) \in E^{\pm}(M)$. Then for every $(R_1, R_2) \in Rg(E)$ it holds that $M' \models \bot$ where $M' = M[R_1 \cup cf(R_2)]$, i.e., for every belief state $S = (S_1, \ldots, S_n)$ some $1 \leq i \leq n$ exists such that $S_i \notin \mathbf{ACC}_i(app(br_i(M'), S))$.

We show that

$$\begin{aligned} I_E = &\{e1(r) \mid r \in E_1\} \cup \{ne1(r) \mid r \in br(M) \setminus E_1\} \\ &\cup \{e2(r) \mid r \in E_2\} \cup \{ne2(r) \mid r \in br(M) \setminus E_2\} \\ &\cup I_{spoil} \end{aligned}$$

is an answer set of $P_p^E(M)$.

Since I_E contains respective instances for $e1$, $ne1$, $e2$, and $ne2$, $fP_p^E(M)^{I_E}$ contains the following rules: $tr_{3.5}(r)$ such that $r \in E_1$; $tr_{3.6}(r)$ such that $r \in br(M) \setminus E_1$; $tr_{3.7}(r)$ such that $r \in E_2$; and $tr_{3.8}(r)$ such that $r \in br(M) \setminus E_2$. Furthermore, because I_E contains I_{spoil}, $fP_p^E(M)^{I_E}$ contains all rules in $TR_{3.3}(M) \cup TR_{3.4}(M) \cup TR_{3.9}(M) \cup \ldots \cup TR_{3.26}(M)$. Given that $I_{spoil} \subset I_E$, it is easy to see that I_E is a model of $fP_p^E(M)^{I_E}$. It remains to show that I_E is a $\subseteq$-minimal model of $fP_p^E(M)^{I_E}$.

Assume for contradiction that some $I' \subset I_E$ is a model of $fP_p^E(M)^{I_E}$. Observe that I_E consistently encodes E by definition. Since it must hold that $I' \models tr$ where $tr \in TR_{3.3}(M) \cup TR_{3.4}(M)$ and $I' \subset I_E$, it follows that I' also consistently encodes E.

Since $fP_p^E(M)^{I_E}$ contains rules $TR_{3.23}(M) \cup \ldots \cup TR_{3.26}(M)$ which must be satisfied by I', either $spoil \notin I'$ or all respective heads are in I', which means that I' is saturated. The latter implies that $I' = I_E$, which contradicts the assumption $I' \subset I_E$. Hence it follows that $spoil \notin I'$. This requires that $I' \not\models B_{\text{HEX}}(tr)$ where $tr \in TR_{3.19}(M) \cup TR_{3.20}(M) \cup TR_{3.21}(M) \cup TR_{3.22}(M)$.

Since it holds that $I \not\models B_{\text{HEX}}(tr_{3.19}(i))$ for all $1 \leq i \leq n$, there exists a contradiction-free guess regarding $r1$, $nr1$, $r2$, $nr2$, $pres_i$, abs_i such that $f_{\&con\_out'_i}(I', pres_i, in_i) = 1$. Let $S = (S_1, \ldots, S_n)$ be the belief state consistently encoded by I' and let (R_1, R_2) be the pair of

master.hex:	`#context(1,"dlv_asp_context_acc", "kb1.dlv").`
	`#context(2,"dlv_asp_context_acc", "kb2.dlv").`
	`#context(3,"ontology_context3_acc", "").`
	`#context(4,"dlv_asp_context_acc", "kb4.dlv").`
	`r1: (3:pneum) :- (2:xraypneum).`
	`r2: (3:marker) :- (2:marker).`
	`r3: (4:need_ab) :- (3:pneum).`
	`r4: (4:need_strong) :- (3:atyppneum).`
	`r5: (4:allow_strong_ab) :- not (1:allergystrong).`

kb1.dlv:	`allergystrong.`
kb2.dlv:	`marker. xraypneum.`

kb4.dlv:	`give_strong v give_weak :- need_ab.`
	`give_strong :- need_strong.`
	`give_nothing :- not need_ab, not need_strong.`
	`:- give_strong, not allow_strong_ab.`

Figure 3.9: Examples for MCS topology and knowledge base input files of the MCS-IE tool. These files encode most parts of our running example.

sets of bridge rules consistently encoded by I'. It holds that $(R_1, R_2) \in Rg(E)$, because $TR_{3.5}(M)$ and $TR_{3.7}(M)$ together with the fact that I' is contradiction-free ensure: $e1(r) \in I'$ implies $r1(r) \in I'$ and $r2(r) \in I'$ implies that $ne2(r) \in I'$. In other words, $R_1 \subseteq E_1$ and $R_2 \subseteq br(M) \setminus E_2$, hence $(R_1, R_2) \in Rg(E)$.

By Lemma 3.11, $\{b \in IN_i \mid in_i(b) \in I'\} = \{\varphi(r) \mid r \in app(br_i(M[R_1 \cup cf(R_2)]), S)\}$ for every $1 \leq i \leq n$, which implies that $S_i \in \mathbf{ACC}_i(\{\varphi(r) \mid r \in app(br_i(M[R_1 \cup cf(R_2)]), S)\})$; i.e., S is an equilibrium of $M[R_1 \cup cf(R_2)]$. Since $(R_1, R_2) \in Rg(E)$, this contradicts that E is an explanation of M. It follows that no $I' \subset I_E$ is a model of $fP_p^E(M)^{I_E}$. Hence I_E is an answer set of $P_p^E(M)$. $\square$

Implementation and Evaluation

The above encoding in HEX for explanation computation is implemented in the MCS-IE[3] tool, the MCS Inconsistency Explainer [22], which is an experimental prototype based on the dlvhex solver. MCS-IE solves the reasoning tasks of enumerating output-projected equilibria, diagnoses, minimal diagnoses, explanations, and minimal explanations of a given MCS. To do so, it uses the encoding presented here and further, straightforward encodings for diagnosis computation (cf. [54] for a complete list). Note that the author of this thesis is only involved in the development and implementation of a MCS-IE plug-in realizing the $P_p^E(M)$ rewriting; the author is not involved in the development of the MCS-IE system itself.

Contexts can be realised as ASP programs, or by writing a context reasoning module using a C++ interface which allows for implementing arbitrary formalisms that can be captured by MCS

[3] `http://www.kr.tuwien.ac.at/research/systems/mcsie/`

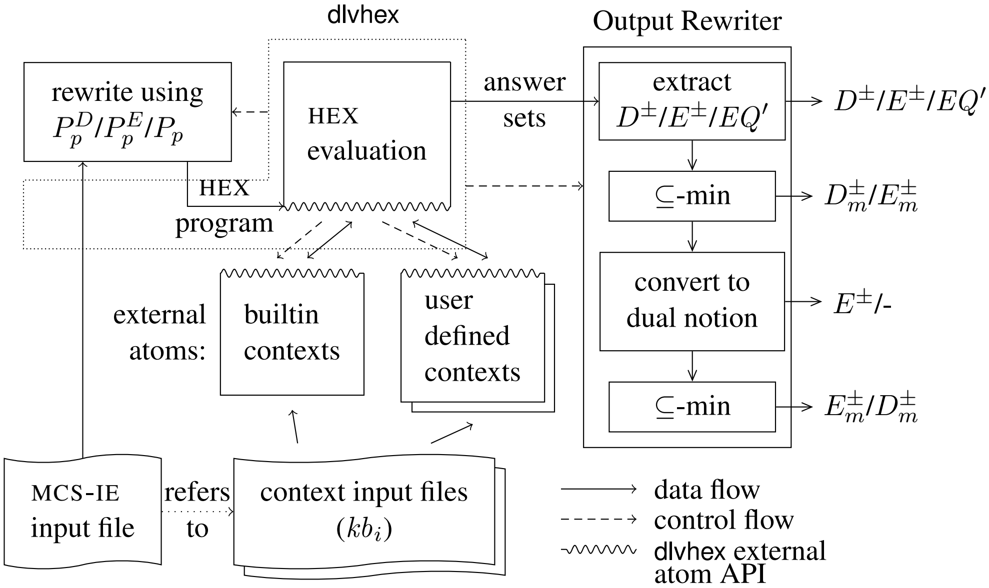


Figure 3.10: Architecture of the MCS-IE system.

contexts.

An online version of MCS-IE is available[4], which is a useful research tool for quick analysis of inconsistency in small-scale MCS. It requires no installation of additional software on the user side and allows direct editing of bridge rules and context knowledge-bases. A list of showcase MCS allows to directly compute (minimal) diagnoses and (minimal) explanations also for the MCS of Example 3.1.

Example 3.25 (ctd)**.** *Figure 3.9 shows files which encode our running example MCS in the* MCS-IE *input format. Contexts C_1, C_2, and C_4 are formalised in ASP, with knowledge bases* `kb1.dlv`*,* `kb2.dlv`*, and* `kb4.dlv`*, these contexts are evaluated through a* HEX*-plugin for external atoms, which in turn uses the* dlv *solver. On the other hand, ontology reasoning C_3 is implemented in* C++*. For more details about the format and the interface we refer to [22].*

Figure 3.10 shows the architecture of the MCS-IE system, which is implemented as a plugin to the dlvhex solver. The MCS M at hand is described by the user in a master input file, which specifies all bridge rules and contexts (it may refer to context knowledge base files). Depending on the configuration of MCS-IE, the desired reasoning tasks are solved using one of the three rewritings $P_p(M)$, $P_p^D(M)$, resp. $P_p^E(M)$, on the input MCS M. Note that this thesis only presents $P_p^E(M)$ while the other rewritings are also given in [54]. MCS-IE enumerates answer sets of the rewritten program, and potentially uses a ⊆-minimisation module, and a module which realises the conversions between diagnosis and explanation notions as described in Theorem 3.2

[4] `http://www.kr.tuwien.ac.at/research/systems/mcsie/tut/`

and Corollary 3.1. Explanations can be computed by MCS-IE using the direct encoding given in Section 3.5 or through the conversion from diagnoses.

As expected, MCS-IE shows the following behaviour wrt. efficiency: the rewriting $P_p^D(M)$, which uses guess-and-check, shows better performance than the rewriting $P_p^E(M)$, which expresses the $\mathbf{coNP}$ task of recognising explanations in the $\mathbf{\Sigma_2^P}$ formalism of full-fledged disjunctive HEX programs.

Nevertheless, it appeared that also $P_p^D(M)$ does not scale well. This lead to the development of a better HEX evaluation framework, which divides and conquers the guessing space more efficiently [49]. Further improvements on the algorithmic aspects are clearly necessary, but outside the scope of this thesis.

3.6 Summary and Outlook

We have considered the problem of inconsistency analysis in nonmonotonic Multi-Context Systems (MCS), which are a flexible, abstract formalism to interlink heterogeneous knowledge sources for information exchange. We have presented a consistency-based and an entailment-based notion of inconsistency explanation, called diagnosis and explanation, which are in a duality relation that can be exploited for computational purposes. We furthermore investigated possible refinements and showed that our basic notions are sufficient to cover the refined notions. We proved several properties regarding the relationship of diagnoses and explanations as well as modularity based on splitting-sets, product-based combinations of MCS, and decompositions of contexts. These properties include the facts that:

- diagnoses and explanations cover the same bridge rules.
- if the information exchange of an MCS exhibits a modular structure, then diagnoses and explanations of certain parts of the MCS can be extended to diagnoses and explanations of the full MCS. Under stronger conditions, diagnoses and explanations of the whole MCS are combinations of those parts of the system.
- diagnoses are neutral to shuffling the order of contexts and corresponding bridge rules of an MCS. We also show that in some cases, a context may be decomposed into two contexts such that diagnoses of the decomposed MCS correspond one-to-one to those of the original MCS.

These results all can aid in a more efficient calculation of diagnoses and explanations. We recall the characterisation of the computational complexity of the two notions, which establish generic results for a range of context complexities. They show that in many cases, explaining inconsistency does not lead to a jump in complexity compared to inconsistency testing, although (unsurprisingly) depending on the interlinking intractability might arise. We have furthermore shown how explanations can be computed by a transformation to HEX programs, which has been implemented in the experimental software tool MCS-IE.

Our results provide a basis for building advanced systems of interlinked knowledge sources, in which the natural need for inconsistency management is supported, by taking specifically the information linkage as a source of inconsistency into account, in contrast to traditional works (cf. Chapter 6) on inconsistency management that focus on the contents of the knowledge sources; however, in loosely connected systems, control over autonomous knowledge sources is elusive and modifying the information exchange may be the only resort to remove inconsistency.

Further Work. The work presented here has been continued in several directions. One of them is to impose different kinds of preferences on the notions of diagnosis and explanation. In Chapter 4 we present these approaches in detail; they allow for filtering and comparing diagnoses; using meta-programming techniques, the most-preferred diagnoses can be selected from all diagnoses and unwanted diagnoses can be filtered-out.

Another direction concerns incomplete information about contexts. The setting considered in here assumes complete information about the behaviour of the contexts in information exchange, i.e., for each 'input' of relevant beliefs from other contexts accessed via bridge rules, the 'output' in terms of firing bridge rules is fully known. In real-world applications, however, this information may be only available for specific (classes of) inputs, and querying a context arbitrarily often to gain this knowledge might be infeasible. In such scenarios the notions introduced in [51] allow to obtain reasonable approximations for diagnoses and explanations of inconsistency.

Finally, another implementation is available in which diagnoses and explanations can be computed by distributed algorithms, exploiting the distributed MCS evaluation framework of [5,6,43].

Another issue is to combine inconsistency management of contents and of context interlinking. Many approaches exist to repair inconsistency stemming from an inconsistent theory or arising from the merging of data from different sources (cf. Chapter 6). Some of these approaches like maximal consistent subsets of a knowledge base (which are ubiquitous in content-based inconsistency management) might be simulated using bridge rules. However, an emerging combination—although in a uniform formalism—would be inflexible and less amenable to refinement. More promising is to combine the notions in this chapter and in [32], which generalised MCS with a management component for each context and operations to be performed on the knowledge base when a bridge rule fires; this allows for a more sophisticated content-change than simple addition of formulas. Nevertheless, consistency can not be guaranteed in general with such content-based approaches, as inconsistency caused by cyclic information flow can not be resolved. Since the latter can be dealt with by modifying the interlinking, as for instance by our notion of diagnosis, a combination of techniques is advantageous. In Chapter 5 we present such local inconsistency management components and further investigate their impact on consistency of MCS.

CHAPTER 4

Preferences

4.1 Introduction

Given an inconsistent MCS the notion of diagnosis yields all possible ways to remove inconsistency. Since this notion is purely technical, it is not able to further distinguish unwanted diagnoses from preferred ones. Although the set of minimal diagnoses yields minimal modifications in order to ensure the existence of an equilibrium, it cannot identify diagnoses whose modifications yield serious consequences like wrongfully considering an ill patient as healthy and not giving her any medication.

Example 4.1. *Let M be an MCS handling patient treatments and billing in a hospital; it contains the following contexts: a patient database C_1, a logic program C_2 suggesting proper medication, and a logic program C_3 handling the billing. Context C_1 uses the abstract logic L^{pl}_{Σ}, while both C_2 and C_3 use L^{asp}_{Σ}. We restrict our example to a single patient with the following knowledge bases for contexts:*

$$\begin{aligned}
kb_1 = \{&hyperglycemia, allergic_animal_insulin, insurance_B\},\\
kb_2 = \{&give_human_insulin \vee give_animal_insulin \leftarrow hyperglycemia.\\
&\bot \leftarrow give_animal_insulin, not\ allow_animal_insulin\},\\
kb_3 = \{&bill \leftarrow bill_animal_insulin.\\
&bill_more \leftarrow bill_human_insulin.\\
&\bot \leftarrow insurance_B, bill_more.\}
\end{aligned}$$

Context C_1 provides information that the patient has severe hyperglycemia, that she is allergic to animal insulin, and that her health insurance is from company B. Context C_2 suggests to apply either human or animal insulin if the patient has hyperglycemia and requires that the applied insulin does not cause an allergic reaction. Context C_3 does the billing and encodes that

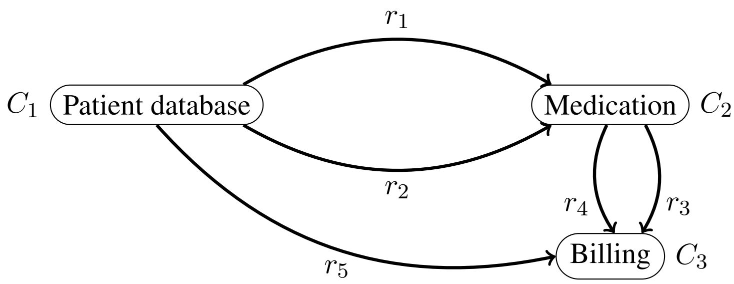


Figure 4.1: Contexts and bridge rules of the hospital MCS $M = (C_1, C_2, C_3)$ of Example 4.1.

insurance B only pays animal insulin. The bridge rules of M are:

$$\begin{array}{lll} r_1: & (2 : \mathit{hyperglycemia}) & \leftarrow (1 : \mathit{hyperglycemia}). \\ r_2: & (2 : \mathit{allow\_animal\_insulin}) & \leftarrow \mathbf{not}\ (1 : \mathit{allergic\_animal\_insulin}). \\ r_3: & (3 : \mathit{bill\_animal\_insulin}) & \leftarrow (2 : \mathit{give\_animal\_insulin}). \\ r_4: & (3 : \mathit{bill\_human\_insulin}) & \leftarrow (2 : \mathit{give\_human\_insulin}). \\ r_5: & (3 : \mathit{insurance\_B}) & \leftarrow (1 : \mathit{insurance\_B}). \end{array}$$

The MCS and its bridge rules are shown in Figure 4.1. Since the patient has hyperglycemia and is allergic to animal insulin, the belief set containing $\mathit{give\_human\_insulin}$ is the only one acceptable at C_2, i.e., the human insulin must be given. Since the insurance company does not cover human insulin, the billing context C_3 admits no acceptable belief set and the MCS M therefore is inconsistent.

The minimal diagnoses of M are as follows:

$$D^{\pm}_{m}(M) = \{(\{r_1\}, \emptyset), (\{r_4\}, \emptyset), (\{r_5\}, \emptyset), (\emptyset, \{r_2\})\}$$

Applying one of these diagnoses, i.e., considering for $(D_1, D_2) \in D^{\pm}_{m}(M)$ the MCS $M[br(M) \setminus D_1 \cup cf(D_2)]$, yields that the illness of the patient is ignored, that the medication is not billed, that the insurance receives a bill it will not pay, and that the patient is given a medication she is allergic to, respectively.

It is not easy to identify the best minimal diagnosis among those available. If the health of the patient is most important, then those diagnoses only causing a wrong billing are preferred. On the other hand, if economic values are paramount, one might consider any diagnosis leading to a wrong billing as unacceptable.

In this chapter we therefore address the problem of distinguishing and selecting the most preferred diagnoses, respectively the filtering of unwanted diagnoses.

A related approach to guaranteeing an equilibrium in MCS is elaborated in [16, 17, 19]; it is based on trust among contexts and provenance information. In contrast to that, we do not focus on a single formalism for preference, and in the spirit of MCS we aim for a solution which is general and open to a wide variety of preference formalisms. As a first step, we therefore introduce three general notions for selecting preferred diagnoses: first, protecting bridge rules unconditionally from any modification; second, removing diagnoses from any further consideration (filtering); third, selecting diagnoses that are most preferred with respect to an arbitrary preference relation.

Having these basic notions, we then show how they can be realised within the established MCS framework. The core idea to realising the selection of preferred diagnoses without confinement to a certain formalism is to use a context of an MCS for preference specification. This requires the ability of introspection or meta-reasoning regarding possible diagnoses of the MCS. Finding techniques that enable an MCS to achieve capabilities for meta-reasoning about the diagnoses of itself therefore is an important task. Our contributions here are the definition of a variety of meta-reasoning techniques as well as an investigation of some of their properties.

Finally, we focus on computational complexity to show that protecting bridge rules from modifications is easy to achieve. The complexity of filtering depends on the complexity of deciding whether a diagnosis is filtered out, but our approach comes with no additional cost. Regarding the selection of a most-preferred diagnosis, one the one hand it depends on the computational complexity of deciding whether a diagnosis is preferred over another, but on the other hand it also depends on the choice of the meta-reasoning approach. We show that one incurs exponential cost, while another approach is worst-case optimal.

The remainder of this chapter is structured as follows. In Section 4.2 we introduce some general types of preferences, i.e., filters and preference orders. In Section 4.3 we investigate two approaches to achieve meta-reasoning in MCS and we realise filters, and preference orders. In Section 4.4 we derive the computational complexity of the developed approaches to filter and select most-preferred diagnoses.

4.2 Preferences and Filters

Clearly, not all diagnoses are equally "good", since the application of some of them might have serious consequences, e.g., in Example 4.1 if the patient is treated as being all healthy. In the literature two basic ways occur frequently: one is to separately consider each outcome (i.e., diagnosis) and discard it whenever it fails some preference condition; the other is to compare outcomes with each other and decide which is the most appealing. We call the former a filter, since it filters unwanted diagnoses, and the other a preference.

Many formalisms have been developed for specifying preference and in order to capture as many as possible, we use preferences in their most general form, i.e., we use mathematical order relations. We also consider two sample instantiations, namely CP-nets (cf. [25]) where preference is specified by statements like "if bride rules r_1 and r_2 are removed, I prefer bridge rule r_3 to be condition-free" and an approach based on units of modified bridge rules.

Since preferences allow to rank diagnoses, but they do not allow the exclusion of diagnoses from being considered, preferences alone are not sufficient. If one wants to ensure that certain diagnoses are excluded from being considered valid, the need arises for a way to filter out certain diagnoses. For specifying a filter, we again use the most general approach, which is a Boolean function on diagnoses.

In this section we introduce the definitions of filters and preference orders in general, as well as some specific preference formalisms. The following sections then show how they can be realised in MCS in such a way that any formalism used to define the preference order or filter can be incorporated thanks to using the abstract logic of an MCS context. Furthermore, our approach preserves core properties of MCS like information hiding and decentralised evaluation.

Filters on Diagnoses

Filters allow a designer of an MCS to apply sanity checks on diagnoses, thus they can be seen as hard constraints: diagnoses that fail to satisfy the conditions are filtered out and not considered for consistency restoration.

Protecting Bridge Rules

In a first attempt, we may consider protecting some bridge rules from being modified altogether, i.e., we disallow a diagnosis to contain protected bridge rules. The adapted notion of diagnosis such that certain bridge rules, tagged as protected, are never part of it is as follows.

Definition 4.1. *Let M be an MCS with* protected rules $br_P \subseteq br(M)$. *A* diagnosis excluding protected rules br_P *is a diagnosis $(D_1, D_2) \in D^{\pm}(M)$, where $D_1, D_2 \subseteq br(M) \setminus br_P$. We denote the set of all minimal such diagnoses by $D^{\pm}_m(M, br_P)$.*

Example 4.2. *Consider the hospital MCS M of Example 4.1 again. One might decide that bridge rules for health-related information-flow are protected, i.e., $br_P = \{r_1, r_2\}$.*

The set of minimal protected diagnoses then is:

$$D^{\pm}_m(M, br_P) = \{(\{r_4\}, \emptyset), (\{r_5\}, \emptyset)\}$$

In the following we also write diagnosis with protected bridge rules meaning a diagnosis excluding protected rules. It follows directly from the definition that every diagnosis with protected rules also is a regular diagnosis. Furthermore, every minimal diagnosis with protected bridge rules also is a regular minimal diagnosis.

Proposition 4.1. *Let M be an inconsistent MCS with protected rules br_P. Then $D^{\pm}_{(m)}(M, br_P) \subseteq D^{\pm}_{(m)}(M)$, i.e., every (minimal) diagnosis excluding protected rules is a (minimal) diagnosis.*

Proof. Let $D \in D^{\pm}(M, br_P)$, then by definition $D \in D^{\pm}(M)$.

Given $D = (D_1, D_2) \in D^{\pm}_m(M, br_P)$, assume towards contradiction that there exists $(D'_1, D'_2) \in D^{\pm}_m(M)$ such that $(D'_1, D'_2) \subset (D_1, D_2)$. Observe that $D'_1, D'_2 \subseteq br(M) \setminus br_P$, hence $(D'_1, D'_2) \in D^{\pm}(M, br_P)$. This contradicts that $D \in D^{\pm}_m(M, br_P)$, thus it follows that $D \in D^{\pm}_m(M)$. □

Observe that $D^{\pm}_m(M, br_P)$ not necessarily contains cardinality-minimal diagnoses, consider for example an MCS M' with two diagnoses $D = (\{r_1\}, \emptyset)$ and $D' = (\{r_2, r_3\}, \emptyset)$ and $br_P = \{r_1\}$, then D is cardinality-minimal but it holds that $D \notin D^{\pm}_m(M', br_P)$ and $D' \in D^{\pm}_m(M', br_P)$.

In Section 4.4 it is shown that the computational complexity of deciding whether $D \in D^{\pm}(M, br_P)$ holds is the same as deciding whether $D \in D^{\pm}(M)$ holds, i.e., the computational complexity of diagnosis with protected bridge rules is the same as for regular diagnosis.

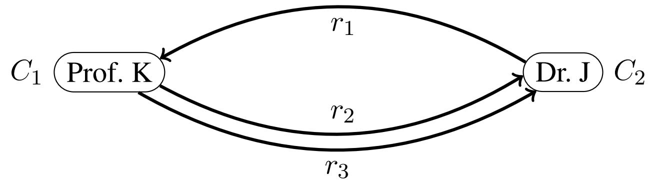


Figure 4.2: Contexts and bridge rules of the MCS $M = (C_1, C_2)$ from Example 4.3.

Filters in General

We now introduce filters in general, which allow a designer of an MCS to apply sanity checks on diagnoses. A whole diagnosis candidate (D_1, D_2) is considered whether it fails some conditions and if it does not satisfy the conditions, it is filtered out and not considered for consistency restoration; thus a filter can be seen as hard constraints on diagnoses. We here want our approach to be open to a large variety of formalisms and therefore we characterise a filter by a Boolean function (or characteristic function) on diagnosis candidates.

Example 4.3. *Consider two scientists, Prof. K and Dr. J, planning to write a paper. We formalise their reasoning in an MCS M using two contexts C_1 and C_2, each employing L_Σ^{asp} for answer set semantics. Dr. J will write most of the paper and Prof. Kwill participate if she finds time or if Dr. J thinks the paper needs improvement (bridge rule r_1). Dr. J knows that the participation of Prof. K results in a good paper (r_2 and kb_1) and he will name Prof. K as author if she participates (r_3). The knowledge bases of the contexts are:*

$$\begin{aligned} kb_1 = \{\, & \mathit{contribute} \leftarrow \mathit{improve}. \\ & \mathit{contribute} \leftarrow \mathit{has\_time}.\} \\ kb_2 = \{\, & \mathit{good} \leftarrow \mathit{coauthored}.\} \end{aligned}$$

The bridge rules of M are:

$$\begin{aligned} r_1 :& \quad (1 : \mathit{improve}) \leftarrow \mathbf{not}\ (2 : \mathit{good}). \\ r_2 :& \quad (2 : \mathit{coauthored}) \leftarrow (1 : \mathit{contribute}). \\ r_3 :& \quad (2 : \mathit{name\_K}) \leftarrow (1 : \mathit{contribute}). \end{aligned}$$

Figure 4.2 depicts the contexts and bridge rules of M. It appears that M is inconsistent, intuitively because the cycle through bridge rules r_1 and r_2 has an odd number of negations.

The set of minimal diagnoses of M is:

$$D_m^{\pm}(M) = \{\ (\{r_1\}, \emptyset),\ (\{r_2\}, \emptyset),\ (\emptyset, \{r_2\}),\ (\emptyset, \{r_1\})\ \}$$

The first two diagnoses break the cycle by removing a rule, the last two "stabilise" it.

The definition of a filter on diagnosis candidates then is as follows.

Definition 4.2. *Let M be an MCS with bridge rules $br(M)$. A diagnosis* filter *for M is a function $f{:}2^{br(M)} \times 2^{br(M)} \rightarrow \{0,1\}$ and the set of* filtered diagnoses *is $D^{\pm}_{f}(M) = \{(D_1, D_2) \in D^{\pm}(M) \mid f(D_1, D_2) = 1\}$. By $D^{\pm}_{m,f}(M)$ we denote the set of all subset-minimal such diagnoses.*

Given a diagnosis candidate $D = (D_1, D_2) \in 2^{br(M)} \times 2^{br(M)}$, we also write $f(D)$ to denote $f(D_1, D_2)$. Writing the set $D^{\pm}_{f,m}(M)$ explicitly, we obtain:

$$D^{\pm}_{m,f}(M) = \big\{D \mid D \in D^{\pm}(M) \text{ and } f(D) = 1 \text{ and } \nexists D' \in D^{\pm}(M) : \big(D' \subset D \text{ and } f(D') = 1\big)\big\} \quad (4.1)$$

Example 4.4. *Consider the MCS of Example 4.3 and the diagnoses $D = (\{r_2\}, \emptyset)$ and $D' = (\emptyset, \{r_2\})$, where the contribution of Prof. K is either enforced or forbidden. For both cases, the authorship information conveyed by r_3 is wrong. Using a filter, we can declare diagnoses undesired if they modify r_2 without modifying r_3 accordingly as follows:*

$$f(D_1, D_2) = \begin{cases} 0 & \textit{if } r_3 \in D_1, r_2 \notin D_1 \textit{ or } r_3 \notin D_1, r_2 \in D_1 \\ 0 & \textit{if } r_3 \in D_2, r_2 \notin D_2 \textit{ or } r_3 \notin D_2, r_2 \in D_2 \\ 1 & \textit{otherwise} \end{cases}$$

In particular it holds that $f(D) = 0 = f(D')$.

Note that filters are a generalisation of diagnoses with protected bridge rules, since every diagnosis with protected bridge rules corresponds to a filter as follows. Let M be an MCS with protected bridge rules br_P, we construct a filter f^{br_P} in the following way:

$$f^{br_P}(D_1, D_2) = \begin{cases} 0 & \text{if } \exists r \in br_P : r \in (D_1 \cup D_2) \\ 1 & \text{otherwise} \end{cases}$$

It is easy to see that $D \in D^{\pm}(M, br_P)$ holds iff $f^{br_P}(D) = 1$. From the definition of f^{br_P} one can also see that diagnoses with protected bridge rules are some kind of modular filter, where each bridge rule of a diagnosis D can be checked independently of the other bridge rules.

It also holds that every filtered diagnosis is a regular diagnosis, but minimal filtered diagnoses are not necessarily regular minimal diagnoses. So an analogous property as in Proposition 4.1 does not hold with respect to minimal filtered diagnoses. The following example shows why.

Example 4.5. *Consider the MCS M and the filter f of Example 4.4 again. The set of minimal filtered diagnoses is as follows:*

$$D^{\pm}_{m,f}(M) = \big\{(\{r_1\}, \emptyset),\ (\emptyset, \{r_1\}),\ (\{r_2, r_3\}, \emptyset),\ (\emptyset, \{r_2, r_3\})\big\}$$

It holds that $(\{r_2, r_3\}, \emptyset) \notin D^{\pm}_{m}(M)$, but nevertheless $(\{r_2, r_3\}, \emptyset)$ is a subset-minimal diagnosis respecting the condition expressed by the filter f. Intuitively, the latter diagnoses modify the authorship information in a consistent way and are minimal in the sense that no unnecessary modification is applied.

One could argue whether minimal filtered diagnoses should select from the set of regular minimal diagnoses only those which pass the filter, i.e., select the set $\{D \in D^{\pm}_{m}(M) \mid f(D) = 1\}$. Although such a definition looks appealing, it may be the case that no minimal diagnosis passes the filter while there are (not-minimal) diagnoses of M passing the filter. The resulting set of filtered minimal diagnoses then is empty while there are useful diagnoses satisfying the filter and these diagnoses do not incur unnecessary modifications other than those that satisfy the filter condition and make the resulting MCS consistent. Since the notion $D^{\pm}_{m,f}$ contains exactly the latter set of diagnoses, this notion seems to be more suitable.

As it is a key strength of MCS to integrate different knowledge bases in a decentralised manner, users of MCS will want to specify their constraints on diagnoses in a logic of their choice, decentralised, and under the provision that they do not have to disclose information considered private. In Section 4.3 we realise filters within the MCS formalism, such that these properties are retained.

Preferences on Diagnoses

To compare diagnoses and select the most appealing one(s), we use preferences. In the spirit of MCS we also want this approach to be open to any kind of formalism for specifying preference. In general, preference is just an order relation on diagnoses. To avoid counter-intuitive results like A being preferred over B and B being preferred over C, but A not being preferred over C, we only require that preferences are transitive. Since virtually every other preference formalism yields an order relation, we first introduce the general formalisation and later show how two specific formalisms fit into our general approach.

Definition 4.3. *A* preference order *over diagnoses for an MCS M is a transitive binary relation $\preceq$ on $2^{br(M)} \times 2^{br(M)}$; we say that D is preferred to D' if $D \preceq D'$.*

Given a preference order $\preceq$, we denote by $\prec$ the irreflexive version of $\preceq$, i.e., $D \prec D'$ holds iff $D \preceq D'$ and $D \neq D'$ hold. Using a preference order $\preceq$, we can now define what constitutes a most preferred diagnosis. Again the intuition is that such a most preferred diagnosis is one which incurs a minimal set of modifications and there exists no other diagnosis that is strictly more preferred. To do so, we first introduce $\preceq$-preferred diagnoses, which are those diagnoses such that no other diagnosis is strictly more preferred. The most preferred diagnoses then are the subset-minimal ones from the set of $\preceq$-preferred diagnoses.

Definition 4.4. *Let M be an inconsistent MCS. A diagnosis $D \in D^{\pm}(M)$ of M is called* $\preceq$-preferred *iff for all $D' \in 2^{br(M)} \times 2^{br(M)}$ with $D' \prec D \wedge D \not\preceq D'$ it holds that $D' \notin D^{\pm}(M)$. A diagnosis $D \in D^{\pm}(M)$ is* minimal $\preceq$-preferred *iff D is subset-minimal among all $\preceq$-preferred diagnoses. The set of $\preceq$-preferred diagnoses is denoted by $D^{\pm}_{\preceq}(M)$ and the set of minimal $\preceq$-preferred is denoted by $D^{\pm}_{m,\preceq}(M)$.*

Observe that we do not require that $\preceq$ is acyclic and therefore we consider all diagnoses in a cycle to be equally preferred; this justifies the condition of $D' \prec D \wedge D \not\preceq D'$ for defining $D^{\pm}_{\preceq}(M)$.

Example 4.6. *Consider the hospital MCS M of Example 4.1 again, where bridge rules r_1 and r_2 transport information regarding the patient's health and bridge rules r_3, r_4, and r_5 cover the information flow for billing. If we consider it most important that information flow regarding health information is changed as little as possible, a preference order $\preceq$ as follows might be used:*

$$(D_1, D_2) \preceq (D_1', D_2') \text{ iff } \{r_1, r_2\} \cap (D_1 \cup D_2) \subseteq (D_1' \cup D_2') \cap \{r_1, r_2\}$$

We observe that following this definition, the following preferences (and several more) hold:

$$\begin{array}{ccc} (\{r_4, r_5\}, \emptyset) \preceq (\{r_1\}, \emptyset) & (\{r_4\}, \emptyset) \preceq (\{r_1\}, \emptyset) & (\{r_5\}, \emptyset) \preceq (\{r_1\}, \emptyset) \\ (\{r_4, r_5\}, \emptyset) \preceq (\emptyset, \{r_2\}) & (\{r_4\}, \emptyset) \preceq (\emptyset, \{r_2\}) & (\{r_5\}, \emptyset) \preceq (\emptyset, \{r_2\}) \\ (\{r_4\}, \emptyset) \preceq (\{r_5\}, \emptyset) & (\{r_5\}, \emptyset) \preceq (\{r_4\}, \emptyset) & \end{array}$$

Note that $\preceq$ indeed yields cyclic preferences among those diagnosis candidates that are incomparable, especially it holds that $(\{r_4\}, \emptyset) \prec (\{r_5\}, \emptyset)$ and $(\{r_5\}, \emptyset) \prec (\{r_4\}, \emptyset)$. The set of $\preceq$-preferred diagnoses of M then is:

$$D^{\pm}_{\preceq}(M) = \{(D_1, D_2) \mid D_1, D_2 \subseteq \{r_3, r_4, r_5\} \text{ and } r_4 \in D_1 \setminus D_2 \text{ or } r_5 \in D_1 \setminus D_2\}$$

Note that $(\{r_5\}, \emptyset) \in D^{\pm}_{\preceq}(M)$, $(\{r_4\}, \emptyset) \in D^{\pm}_{\preceq}(M)$, and $(\{r_4, r_5\}, \emptyset) \in D^{\pm}_{\preceq}(M)$ all hold. Selecting from $D^{\pm}_{\preceq}(M)$ the subset-minimal ones, we obtain $D^{\pm}_{m,\preceq}(M)$, which is:

$$D^{\pm}_{m,\preceq}(M) = \{(\{r_5\}, \emptyset), (\{r_4\}, \emptyset)\}$$

This agrees with our intuition that a minimal set of modifications should be applied and we favour to modify bridge rules for billing information rather than modifying health-related bridge rules.

For use in the following sections, we also state the sets $D^{\pm}_{\preceq}(M)$ and $D^{\pm}_{m,\preceq}(M)$ explicitly.

$$\begin{aligned} D^{\pm}_{\preceq}(M) &= \{D \in D^{\pm}(M) \mid \forall D' \in D^{\pm}(M) : \neg(D' \prec D \wedge D \not\preceq D')\} \\ &= \{D \in D^{\pm}(M) \mid \forall D' \in D^{\pm}(M) : \neg(D' \preceq D \wedge D' \neq D \wedge D \not\preceq D')\} \\ &= \{D \in D^{\pm}(M) \mid \forall D' \in D^{\pm}(M) : (D' \preceq D \wedge D \not\preceq D') \Rightarrow D' = D\} \end{aligned} \tag{4.2}$$

Based on this, we can define $D^{\pm}_{m,\preceq}(M)$ in terms of $D^{\pm}_{\preceq}(M)$ as follows.

$$\begin{aligned} D^{\pm}_{m,\preceq}(M) &= \{D \in D^{\pm}_{\preceq}(M) \mid \forall D' \in D^{\pm}_{\preceq}(M) : D' \subseteq D \Rightarrow D' = D\} \\ &= \{D \in D^{\pm}(M) \mid \forall D' \in D^{\pm}(M) : (D' \preceq D \wedge D \not\preceq D' \Rightarrow D' = D) \\ &\qquad \wedge \forall D' \in D^{\pm}_{\preceq}(M) : D' \subseteq D \Rightarrow D' = D\} \\ &= \{D \in D^{\pm}(M) \mid \forall D' \in D^{\pm}(M) : (D' \preceq D \wedge D \not\preceq D' \Rightarrow D' = D) \\ &\qquad \wedge \forall D' \in D^{\pm}(M) : \big[(\forall D'' \in D^{\pm}(M) : D'' \preceq D' \\ &\qquad \wedge D' \not\preceq D'' \Rightarrow D'' = D') \wedge D' \subseteq D\big] \Rightarrow D' = D\} \end{aligned}$$

In Section 4.3 we show how preferences can be realised in general.

Sample Instantiations of Preference Orders

CP-nets One preference formalism which exhibits appealing features of locality and privacy is called conditional preference networks (CP-nets) [25]. CP-nets capture a natural class of preference statements like "If my new car is from Japan, I prefer hybrid over diesel engine, assuming all else is equal". Given that MCS are decentralised systems, users may want to express preferences on diagnoses solely based on a local set of bridge rules, assuming all other things equal. Since CP-nets allow to model such local preference and have successfully been used for preference elicitation (e.g. [46]), we consider them in more detail here.

We first recall the formalism of CP-nets (cf. [25]). In terms of CP-nets, the fact that "my new car is from Japan" is an assignment of a value to a variable; here the value is "from Japan" and the variable, call it v_O, is the origin of the car. A *variable* in the terminology of CP-nets is some attribute or feature that may take one of several values. Given a set $V = \{v_1, \dots v_n\}$ of variables, each variable $v_i \in V$ is associated with a set of its possible values, denoted by $Dom(v_i)$. An *assignment* $\mathbf{x}$ of values to a set $X \subseteq V$ of variables is a function that maps each variable $v_i \in X$ to an element of its domain $Dom(v_i)$. An assignment is partial if $X \subset V$ and complete if $X = V$; the set of all assignments to $X \subseteq V$ is denoted by $Assgt(X)$; an assignment is also called an *outcome*. Given an outcome $\mathbf{o} \in Assgt(\{x, y, z\})$, we write $\mathbf{o} = a_x b_y c_z$ to denote that $\mathbf{o}$ maps $x, y, z \in V$ to $a \in Dom(x)$, $b \in Dom(y)$, and $c \in Dom(z)$, respectively; we furthermore identify $\mathbf{o}$ with the set containing these assignments, e.g., $\mathbf{o} = \{a_x, b_y\} \cup \{c_z\}$. Given two sets of variables $X, Y \subseteq V$ with $X \cap Y = \emptyset$ and assignments $\mathbf{x} \in Assgt(X), \mathbf{y} \in Assgt(Y)$ to X and Y, then $\mathbf{xy}$ is the combination of both assignments, hence $\mathbf{xy} \in Assgt(X \cup Y)$; similarly if $d \in Dom(v)$ and $\mathbf{y} \in Assgt(Y)$ such that $v \notin Y$, then we write $\mathbf{y}d$ to denote the assignment from $Assgt(Y \cup \{v\})$ where v is assigned the value d and every $v' \in Y$ is assigned the same value as in $\mathbf{y}$.[1]

In the above sentence, we can identify two variables v_O the origin of the car and v_E the engine type, where $Dom(v_O)$ contains (at least) "from Japan" and $Dom(v_E)$ contains "hybrid" and "diesel". Here, the preference over the variable engine type depends on the value of the variable origin, i.e. v_O is a *parent* variable of v_E, denoted by $Pa(v_O) = \{v_E\}$. Note that a variable v can have multiple parent variables, hence $Pa(v)$ is a set. The dependency among variables is what comprises the network part of a CP-net. Another building block are so-called conditional-preference tables (CPTs), which formalise the preference statements for every variable. Intuitively, a conditional preference table for some variable v associates to each complete assignment for $Pa(v)$, a total preorder over the possible values of v, i.e., over $Dom(v)$. A relation $\precsim$ over a set O of outcomes is a *total preorder* iff it is transitive, reflexive, and for any two elements $o, o' \in O$ it holds that either $o \precsim o'$ or $o' \precsim o$ (both may be the case, hence a total preorder allows indifference)[2].

Formally, a *CP-net* is a directed graph $N = (V, E)$ where $V = \{v_1, \dots, v_n\}$ is a finite set of variables (or features or attributes) and $E \subseteq V \times V$ is the conditional dependency between variables. For $v \in V$ we denote the set of parents of v by $Pa(v) = \{v' \in V \mid (v', v) \in E\}$.

[1]Following the notation from [25], we note that $v \in V$ with $d \in Dom(v)$ is clear from the context whenever $\mathbf{y}d$ is used.

[2]Different from [25], but in line with the remainder of this thesis we write $o \precsim o'$ to denote that o is preferred over o'.

Each variable $v \in V$ of a CP-net $G = (V, E)$ is associated (or labelled) with a *conditional preference table* $CPT(v)$ that maps each outcome $\mathbf{u} \in Assgt(Pa(v))$ of the parents of v to a total preorder $\precsim^v_{\mathbf{u}}$ over $Dom(v)$. Note that $CPT(v)$ contains one order over $Dom(v)$ for every possible assignment of its parent variables; for $Pa(v) = \{p_1, \ldots, p_k\}$ therefore $CPT(v)$ contains $|Dom(p_1)| \times \ldots \times |Dom(p_k)|$ orders over $Dom(v)$.

Example 4.7. *We formalise the sentence "If my new car is from Japan, I prefer hybrid over diesel engine, and if my new car is from Germany, I prefer diesel over hybrid" by the CP-net $N = (V, E)$ with variables $V = \{Origin, Engine\}$ and dependency $E = \{(Origin, Engine)\}$, i.e., there is only one edge in the graph N. We consider only small domains with two countries of origin $Dom(Origin) = \{Japan, Germany\}$ and two engine types $Dom(Engine) = \{hybrid, diesel\}$.*

The CPT of the variable $Engine$ states the above sentence in terms of total preorders for every outcome of its parent $Pa(Origin)$:

$$CPT(Engine): \quad Japan_{Origin}: \quad hybrid \precsim diesel$$
$$Germany_{Origin}: \quad diesel \precsim hybrid$$

The CPT regarding the variable $Origin$ with $Pa(Origin) = \emptyset$ states that a car from Japan is preferred

$$CPT(Origin): \quad Japan \precsim Germany$$

For readability, the above listing only contains $\precsim$, while in fact these are three different total preorders, namely $\precsim^{Engine}_{Japan_{Origin}}$, $\precsim^{Engine}_{Germany_{Origin}}$, and $\precsim^{Origin}_{\emptyset}$ from top to bottom. In this CP-net the outcome that is preferred over all others is $Japan_{Origin} hybrid_{Engine}$.

There are several ways to give the semantics of a CP-net $N = (V, E)$ (cf. [25]), one is by so-called flipping sequences, another is by preference graphs, but the most general one is based on total preorders over all possible outcomes that satisfy the CP-net.

Given a total preorder $\precsim$ over the outcomes of variables $V = \{v_1, \ldots, v_n\}$, i.e. $\precsim$ is a total preorder over $Dom(v_1) \times \ldots \times Dom(v_n)$, then $\precsim$ is said to *satisfy* the CP-net G iff it satisfies the conditional preference table $CPT(v)$ of every $v \in V$; $\precsim$ satisfies $CPT(v)$ iff it satisfies every total preorder $\precsim^v_{\mathbf{u}}$ of $CPT(v)$ with $\mathbf{u} \in Assgt(Pa(v))$. Finally, $\precsim$ satisfies $\precsim^v_{\mathbf{u}}$ iff for all $x, x' \in Dom(v)$ holds that whenever $x \precsim^v_{\mathbf{u}} x'$ holds then it holds for all $\mathbf{y} \in Assgt(V \setminus (\{v\} \cup Pa(v)))$ that $\mathbf{y}x\mathbf{u} \precsim \mathbf{y}x'\mathbf{u}$. Intuitively, $\precsim$ satisfies the CP-net G iff it agrees on all entries of every conditional preference table of G. Observe that not for every CP-net a total preorder exists which satisfies the CP-net. But if a CP-net N is acyclic and indifference in CPTs is not allowed, then some $\precsim$ exists which satisfies N; hence CP-nets are often restricted to be acyclic with their CPTs not containing indifference. In the following, we only consider satisfiable CP-nets, i.e., CP-nets such that there exists at least one total preorder $\precsim$ that satisfies the CP-net.

Given a CP-net $N = (V, E)$ and two outcomes $\mathbf{o}, \mathbf{o}' \in Assgt(V)$, one says N *entails* $\mathbf{o} \precsim \mathbf{o}'$ (i.e. outcome $\mathbf{o}$ is preferred over $\mathbf{o}'$), denoted by $N \models \mathbf{o} \precsim \mathbf{o}'$, iff $\mathbf{o} \precsim \mathbf{o}'$ holds in every total preorder $\precsim$ that satisfies N. The question whether $N \models \mathbf{o} \precsim \mathbf{o}'$ holds is also called a dominance query, since it answers whether the outcome $\mathbf{o}$ dominates the outcome $\mathbf{o}'$, i.e., whether $\mathbf{o}$ is always preferred to $\mathbf{o}'$. Also note that this entailment is transitive, i.e., if $N \models \mathbf{o} \precsim \mathbf{o}'$ and $N \models \mathbf{o}' \precsim \mathbf{o}''$ both hold, then it holds that $N \models \mathbf{o} \precsim \mathbf{o}''$. Given a CP-net N, we can thus readily define a preference order $\preceq$ that is transitive, by taking entailment on N. Note however, that the resulting $\preceq$ is over outcomes of the variables of N and not over pairs of sets of diagnoses, hence we need to decide how a CP-net should represent diagnosis candidates over an MCS M.

It is rather natural that each variable $v \in V$ of a CP-net $N = (V, E)$ represents a bridge rule $r \in br(M)$ and the domain $Dom(v)$ represents the possible modifications expressed in a diagnosis. Since a bridge rule may be removed, condition-free, or unmodified, a first attempt to capture preferences over diagnosis using CP-nets is the following (cf. [55]).

Definition 4.5. *A CP-net $N = (V, E)$ is called* 3-compatible *with an MCS M if the following holds: there exists a bijective mapping $CP_V : br(M) \to V$ mapping bridge rules of M to variables V, and for every $v \in V$ the domain is $Dom(v) = \{unmodified, removed, condition\text{-}free\}$.*

Note that a 3-compatible CP-net can not represent a diagnosis candidate $D = (D_1, D_2) = (\{r\}, \{r\})$ with $r \in br(M)$, since the domain of $CP_V(r)$ can not indicate that r is both removed and condition-free. If D is applied to M, however, $\varphi(r)$ is added to the respective context of $M[D_1, D_2]$, since D makes r condition-free. Now consider the diagnosis $D' = (D'_1, D'_2) = (\emptyset, \{r\})$ and observe that $\varphi(r)$ is added to the respective context of $M[D'_1, D'_2]$. Therefore both diagnoses admit the same equilibria, i.e., $\mathrm{EQ}(M[D_1, D_2]) = \mathrm{EQ}(M[D'_1, D'_2])$. One might argue that this is sufficient and both D and D' should be represented by an outcome of N that maps the variable $CP_V(r)$ to *condition-free*.

On the other hand, distinguishing between both diagnoses might be desired to capture all possible diagnoses of an MCS, so the CP-net needs to represent the possibility that a bridge rule is both removed and condition-free. One could simply extend the notion of 3-compatible to make $Dom(CP_V(r)))$ contain four elements. There are arguments against this, which stem from the analysis of the computational complexity of CP-nets. The computational complexity of binary-valued CP-nets, i.e., CP-nets where for all $v \in V$ holds that $|Dom(v)| = 2$, has been studied more extensively and it is known that dominance queries in certain classes of multi-valued CP-nets are not in **NP** (cf. [25]). Therefore, we suggest to represent a diagnosis candidate (D_1, D_2) such that each bridge rule $r \in br(M)$ is represented by two variables, one indicating whether $r \in D_1$ and another one indicating whether $r \in D_2$.

Definition 4.6. *A CP-net $N = (V, E)$ is* fully compatible *with an MCS M if there exists a partitioning V_1, V_2 of V such that there exists a bijective function $CP^1_V : br(M) \to V_1$ and a bijective function $CP^2_V : br(M) \to V_2$; furthermore, the domains are such that for all $v \in V_1$ it holds that $Dom(v_1) = \{inD1, not\_inD1\}$ and for all $v \in V_2$ it holds that $Dom(v) = \{inD2, not\_inD2\}$.*

Having those two notions of compatibility, it is clear how a diagnosis candidate relates to a global outcome of a compatible CP-net. For an MCS M and a fully compatible CP-net $N = (V, E)$ it is as follows: given a diagnosis candidate $(D_1, D_2) \in 2^{br(M)} \times 2^{br(M)}$ the corresponding outcome is $\mathbf{o} \in Assgt(V)$ such that $\mathbf{o} = \{inD1_r \mid r \in D_1\} \cup \{not\_inD1_r \mid r \notin D_1\} \cup \{inD2_r \mid r \in D_2\} \cup \{not\_inD2_r \mid r \notin D_2\}$. For a 3-compatible CP-net $N = (V, E)$ and given diagnosis candidate (D_1, D_2) the corresponding outcome is $\mathbf{o} = \{unmodified_r \mid r \notin$

$D_1 \cup D_2\} \cup \{removed_r \mid r \in D_1 \setminus D_2\} \cup \{condition\text{-}free_r \mid r \in D_2\}$. The relation is one-to-one in the case of a fully compatible CP-net, while in the case of a 3-compatible CP-net, there are several diagnoses mapping to the same global outcome of the CP-net since a bridge rule that is condition-free and removed is considered as *condition-free* in the CP-net. In the following we then write $N \models D \precsim D'$ to denote that there are outcomes o and o' of N such that o corresponds to D, o' corresponds to D', and $N \models o \precsim o'$.

Example 4.8. *Assume an MCS M where several corporations make contracts using bridge rules. Contract details, such as when a contract will start, how long it is valid, who owes whom money, etc. are encoded with bridge rules. For instance, C_1 is leasing a car from C_2 with the following properties encoded as bridge rules:*

$$r_1: \quad (1: pay(car, 500)) \leftarrow (2: price(car, 500))$$
$$r_2: \quad (1: due(car, monthly)) \leftarrow (2: due(car, monthly))$$

If r_2 is removed to restore consistency, r_1 becomes meaningless and possibly confuses further reasoning. Removing both rules is then preferred to removing only r_2, i.e., if r_2 is removed, r_1 is preferred to be removed, too.

We can represent such a preference with a 3-compatible CP-net $N = (V, E)$ with $V = \{v_1, v_2\}$ and $E = \{(v_2, v_1)\}$, i.e., $Pa(v_1) = \{v_2\}$, $Pa(v_2) = \emptyset$, and $CP_V(r_i) = v_i$ for $i \in \{1, 2\}$. Assuming that adding rules unconditionally is always considered to be the worst option, the conditional preference table of v_1 is:

$$\begin{array}{lll} CPT(v_1): & unmodified_{v_2}: & unmodified_{v_1} \precsim removed_{v_1} \precsim condition\text{-}free_{v_1} \\ & removed_{v_2}: & removed_{v_1} \precsim unmodified_{v_1} \precsim condition\text{-}free_{v_1} \\ & condition\text{-}free_{v_2}: & unmodified_{v_1} \precsim removed_{v_1} \precsim condition\text{-}free_{v_1} \end{array}$$

The CPT for v_2 is:

$$CPT(v_2): \quad unmodified_{v_2} \precsim removed_{v_2} \precsim condition\text{-}free_{v_2}$$

Note that in the first, second, third, and fourth line we write $\precsim$ to denote $\precsim^{v_1}_{unchanged_{v_2}}$, $\precsim^{v_1}_{removed_{v_2}}$, $\precsim^{v_1}_{condition\text{-}free_{v_2}}$, and $\precsim^{v_1}_{\emptyset}$, respectively.

Observe that if the converse preference for r_2 depending on the status of r_1 is desired in addition, the resulting CP-net becomes cyclic, which requires special care to guarantee that the CP-net is satisfiable by some order (cf. [25] for some details on this issue).

Definition 4.7. *Given an MCS M and a CP-net N that is either 3-compatible to M or fully compatible to M, we say a diagnosis $D \in D^{\pm}(M)$ is N-*preferred *iff there exists no $D' \in D^{\pm}(M)$ such that $N \models D' \precsim D$ and it does not hold that $N \models D \precsim D'$. Let $D^N(M)$ denote the set of all N-preferred diagnoses of M then the set of optimal diagnoses preferred according to N are the diagnoses of $D^N(M)$ that are subset-minimal; we denote these by $D^{\pm}_{opt}(M, N)$. Formally,*

$$D^{\pm}_{opt}(M, N) = \{D \in D^N(M) \mid \forall D' \in D^N(M) : D' \subseteq D \Rightarrow D = D'\}.$$

Observe that given a CP-net N that is compatible to the MCS M, we can readily define a preference order $\preceq^N$ that is equivalent to N as follows: for all $D, D' \subseteq 2^{br(M)} \times 2^{br(M)}$ it holds that $N \models D \precsim D'$ iff $D \preceq D'$. Since the entailment of the CP-net is transitive, $\preceq^N$ also is transitive, hence it is a preference relation in the sense of Definition 4.3. A consequence of that is the following:

Proposition 4.2. *Given a CP-net N compatible to an MCS M, let $D \preceq^N D'$ hold iff $N \models D \precsim D'$ holds. Then $D^N(M) = D^{\pm}_{\preceq^N}(M)$ and $D^{\pm}_{m,\preceq^N}(M) = D^{\pm}_{opt}(M, N)$.*

Proof. We first show that $D^N(M) = D^{\pm}_{\preceq^N}(M)$. We write down $D^N(M)$ in set-notation and obtain:

$$\begin{aligned} D^N(M) &= \{D \in D^{\pm}(M) \mid \nexists D' \in D^{\pm}(M) : N \models D' \precsim D \wedge \neg(N \models D \precsim D')\} \\ &= \{D \in D^{\pm}(M) \mid \forall D' \in D^{\pm}(M) : \neg N \models D' \precsim D \vee N \models D \precsim D'\} \end{aligned}$$

Regarding $D^{\pm}_{\preceq^N}(M)$ we have that:

$$\begin{aligned} D^{\pm}_{\preceq^N}(M) &= \{D \in D^{\pm}(M) \mid \forall D' \in D^{\pm}(M) : \neg(D' \precsim^N D \wedge D \not\precsim^N D' \wedge D' \neq D)\} \\ &= \{D \in D^{\pm}(M) \mid \forall D' \in D^{\pm}(M) : \neg(N \models D' \precsim D \wedge \neg N \models D \precsim D' \wedge D' \neq D)\} \\ &= \{D \in D^{\pm}(M) \mid \forall D' \in D^{\pm}(M) : \neg N \models D' \precsim D \vee N \models D \precsim D' \vee D' = D\} \end{aligned}$$

It remains to show that given any $D, D' \in D^{\pm}(M)$, the following two formulas are equivalent:

$$\neg N \models D' \precsim D \vee N \models D \precsim D' \tag{4.3}$$

$$\neg N \models D' \precsim D \vee N \models D \precsim D' \vee D' = D \tag{4.4}$$

Clearly, (4.3) implies (4.4), it thus remains to show that (4.4) implies (4.3). The latter clearly holds if $\neg N \models D' \precsim D$ holds or $N \models D \precsim D'$ holds. Therefore, it only remains to show that in the case where both do not hold, (4.3) is implied by (4.4): from $N \models D' \precsim D$ and $\neg N \models D \precsim D'$ follows $D' = D$, hence by $N \models D' \precsim D$ it then follows that $N \models D \precsim D'$, i.e., (4.3) is satisfied in this case. Consequently, (4.4) implies (4.3) and thus, both conditions are equivalent. Therefore, it holds that $D^N(M) = D^{\pm}_{\preceq^N}(M)$.

It then follows trivially from the definitions of $D^{\pm}_{opt}(M, N)$ and $D^{\pm}_{m,\preceq^N}(M)$ that they are the same, because $D^{\pm}_{opt}(M, N)$ is the set of $\subseteq$-minimal diagnoses of $D^{\pm}_{\preceq^N}(M)$ while $D^{\pm}_{m,\preceq^N}(M)$ is the set of $\subseteq$-minimal diagnoses of $D^N(M)$. □

Another way of defining the semantics of a CP-net $N = (V, E)$ is via so-called flipping sequences. The idea is that given an outcome $\mathbf{o}$ of N, we can find a more preferred outcome $\mathbf{o}'$ by finding one entry $\precsim^v_{\mathbf{u}}$ in the $CPT(v)$ of some variable v that matches with $\mathbf{o}$ but whose outcome may be improved, i.e., it holds that $d' \precsim^v_{\mathbf{u}} d$, $\mathbf{u} \subset \mathbf{o}$, and $d_v \in \mathbf{o}$. This means that flipping the value assigned to v from d to d' yields a complete outcome $\mathbf{o}'$ which is more preferred than $\mathbf{o}$. Formally, let $\mathbf{o}$ and $\mathbf{o}'$ be two complete outcomes of N such that there exists $v \in V$ with $d, d' \in Dom(v)$, $d \neq d'$, $d_v \in \mathbf{o}$, $d'_v \in \mathbf{o}'$, $\mathbf{o} \setminus \{d_v\} = \mathbf{o}' \setminus \{d'_v\}$, $d' \precsim^v_{\mathbf{u}} d$, $\mathbf{u} \subset \mathbf{o}$, and $\mathbf{u} \subset \mathbf{o}'$, then the flip from $\mathbf{o}$ to $\mathbf{o}'$ is an *improving flip*. A sequence $\mathbf{o}_1, \ldots, \mathbf{o}_k$ is a *flipping sequence* of improving flips iff for all $1 \leq i < k$ holds that the flip from $\mathbf{o}_i$ to $\mathbf{o}_{i+1}$ is improving. Intuitively, a complete outcome is optimal, if no improving flips are possible. Indeed, flipping sequences agree with entailment for acyclic CP-nets without indifference (cf. [25]).

In [55] these flipping sequences are used to define the optimal diagnoses of an MCS M given a 3-compatible CP-net N. Let $\mathit{iflips}(D, D')$ denote the set of sequences of improving flips of diagnosis candidates D and D', i.e., $\mathit{iflips}(D, D')$ is nonempty if there exists a flipping sequence from the outcome corresponding to D to the outcome corresponding to D', which holds iff $N \models D' \precsim D$. In terms of flips, the most preferred diagnoses $D^{\pm}_{opt}(M, N)$ of an MCS then are $D^{\pm}_{opt}(M, N) = \min_{\subseteq}\{D \in D^{\pm}(M) \mid \forall D' \in D^{\pm}(M) : \mathit{iflips}(D, D') = \emptyset\}$.

The computational complexity of deciding whether a global outcome is preferred over another by a given CP-net is very much depending on properties of the given CP-net (cf. [25, 75]). It can be decided in quadratic time if the CP-net is binary-valued and tree-structured; for binary-valued directed-path singly connected CP-nets, the same decision problem, however, is $\mathbf{NP}$-complete. If a CP-net is multi-valued with partially specified preferences, then the problem is not in $\mathbf{NP}$.[3] This is important for realising CP-net based preferences on diagnoses of an MCS, because either the realisation must be open to being adapted to the specific computational needs, or only a restricted variant of CP-nets may be used. Since the former is more general and more appealing, we realise CP-nets in a general manner in the following sections.

One further way to give the semantics of CP-nets is via the corresponding outcome graph. A CP-net N induces a preference graph G_N over complete outcomes, where each complete outcome is a vertex in the preference graph. An edge from outcome o_i to o_j indicates that a preference for o_j over o_i can be determined directly from one conditional preference table of the CP-net (cf. [25]). The transitive closure G^+_N of a preference graph induces a preference order on global outcomes; conceptually this transitive closure is similar to flipping sequences, i.e., there is an edge from outcome o to o' iff the corresponding diagnosis candidates D and D' are such that $\mathit{iflips}(D, D')$ is non-empty. Furthermore, for a CP-net compatible with an MCS, every global outcome represents a potential diagnosis.

Proposition 4.3. *Let M be an inconsistent MCS, and let N be a CP-net associated with M. Then G^+_N induces a preference order $\prec$ over diagnoses of M.*

Proof. G^+_N is the transitive closure of G_N, hence the relation induced by its edge-relation is also transitive, i.e., it is a preference order. □

Unit-based Groups of Bridge Rules In this section we introduce a simple and practical approach to identify preferred diagnoses based on units of bridge rules which together convey information about an entity.[4] Intuitively, a unit of bridge rules is a non-empty set of bridge rules which together ensure that the information flow about some entity is correct. For example, the bridge rules r_1 and r_2 of Example 4.1 convey necessary information about the patients condition, i.e., her illness and her allergy. Information about only one of these two leads to wrong

[3] The proof of Theorem 20 in [25] shows that exponentially long flipping sequences may occur in such CP-nets. We conjecture that the same construction and proof is applicable to those CP-nets for preferences on diagnoses as written in [55], since indifference can be used to the same effect as partially specified preferences.

[4] In [126] a unit of bridge rules is called a "category". To avoid confusion, we call it unit of bridge rules here.

conclusions and might be dangerous, e.g., if a diagnosis makes r_2 unconditional then the patient is given animal insulin, risking an allergic reaction. It is better to not conclude anything about the patient than making wrong and dangerous conclusions. Such preference is justified whenever it is the case that a group of bridge rules only makes sense if none of them is modified.

In logic programming often a single rule by itself is not useful, but only several rules together form a specific behaviour and cover an intended meaning. As syntax and semantics of bridge rules is inspired by logic programming rules, we predict that the same also holds for bridge rules. Names for units of bridge rules in general are arbitrary, including the possibility of a syntactic derivation from the MCS, e.g., by a combination of involved beliefs and knowledge-base formulas.

Definition 4.8. *Let $\mathcal{U}$ be the set of unit names, M an MCS, and for each $r \in br(M)$ let $unit(r) \subseteq \mathcal{U}$ be an association of bridge rules to (one or more) unit names. By $U_M = \bigcup_{r \in br(M)} unit(r)$ we denote the set of units of bridge rules of M.*

Note that unit names only serve the purpose of naming the units of bridge rules explicitly; from the formal perspective, each unit name could be substituted by the set of bridge rules that are associated with it. In the following we also identify a unit name $u \in U_M$ with the set of bridge rules $\{r \in br(M) \mid unit(r) = u\}$ that belong to u.

Example 4.9. *In Example 4.1 rules r_1 and r_2 carry the information of how to treat the patient correctly, while rules $r_3, r_4,$ and r_5 carry information for accounting and billing. We can identify two units of bridge rules, e.g., "treatment" for bridge rules r_1, r_2 and "billing" for r_3, r_4, r_5.*

We formalise this using the set of unit names $U_M = \{treatment, billing\}$ and associating bridge rules to units as follows:

$$unit(r_1) = unit(r_2) = \{treatment\}$$
$$unit(r_3) = unit(r_4) = unit(r_5) = \{billing\}$$

This grouping naturally follows from what the bridge rules are intended to do.

The identification of bridge rules that work together is usually easy at design time of an MCS, since the person(s) specifying bridge rules know what their intended meaning is, i.e., they know which bridge rules form a unit and what it should do. In traditional programming such information is often explicitly expressed in programmers' comments in the source code.

If a bridge rule is modified by a diagnosis, it is likely that the behaviour of all units the bridge rule is part of is modified and possibly corrupted. Furthermore, if the result of unit of bridge rules A depends on another unit of bridge rules B, then A gives wrong or unexpected results if B is modified, although A was not modified directly. Therefore units may depend on each other and modifications of rules of one unit also changes the result of units that depend upon the former unit. We therefore also consider dependencies among units.

Definition 4.9. *Let U_M be the units of an MCS M. Each $u \in U_M$ is associated with a set of units $P_u \subseteq U_M$ it depends on. We write $dep(u, u')$ iff $u' \in P_u$.*

Example 4.10. *In Example 4.9, if r_2 is modified, the patient not only is given a different treatment, but also the billing gives other results than expected, since there is a patient that is not billed at* *all. If this behaviour is correct or not may depend on whether this case was expected to occur for the knowledge base of C_3. Since this case potentially should not happen, we consider the unit "billing" to depend on the unit "treatment", formally $dep(billing, treatment)$.*

Note that the dependency of units as well as their names and associations are semantic information, so for an MCS several categorisations into units may be adequate. If we assume that each bridge rule of an MCS was added by the creator for some reason, then the creator intuitively knows the unit this bridge rule belongs to, i.e., the reason(s) for a bridge rule to exist corresponds to the unit it belongs to. Therefore we assume that units are supplied by the creator of the MCS since they are (at least implicitly) known at the time of creation.

For dependencies among units we also assume them to be specified explicitly by the creator of the MCS. Although, under certain restrictions, it is possible to derive them automatically. For example, if all contexts of an MCS consist of logic programs and those programs are openly known, then one could take the dependency graph G of the whole MCS to derive dependencies among categories. Here G could be the dependency graph over all bridge rules combined with the rules of all contexts (suitably renamed, if necessary). Then a unit u_1 depends on u_2, if there exist bridge rules r_1, r_2 with $u_1 \in unit(r_1)$ and $u_2 \in unit(r_2)$ such that there is a path in G from the node representing the head of r_1 to the node representing the head of r_2. In the case that the unit of a constraint rule depends on two other units, it is, however, not immediately clear if those two units then mutually depend on each other. Therefore an automatic derivation of units has to address further details which are beyond the scope of this thesis.

Different grouping of bridge rules into units and different dependencies may lead to other diagnoses being preferred. Therefore we assume in the following that a grouping of bridge rules into units deemed correct for the given MCS is applied. Whether this can be derived automatically (at least to some extent) is an issue for future work.

Using the dependency information, we can now state which units are influenced by a diagnosis and possibly lead to wrong information.

Definition 4.10. *Let M be an MCS with unit names U_M and dependencies dep. For a diagnosis $D = (D_1, D_2)$ of M, the set of* possibly corrupted units *wrt. D is the smallest set $U_D \subseteq U_M$ such that for all $r \in D_1 \cup D_2$ holds $unit(r) \subseteq U_D$ and whenever $u_1 \in U_D$ and $dep(u_2, u_1)$ then $u_2 \in U_D$.*

A diagnosis which modifies a smaller set of units is always desirable, as it ensures that more parts of the diagnosed system still yield reliable results. This induces a preference order such that preferred diagnoses modify only a minimal set of units.

Definition 4.11. *Let $D, D' \in D^{\pm}(M)$ be diagnoses of an MCS M. D is at least as preferred as D' iff $U_D \subseteq U_{D'}$. We denote this preference order by $D \preceq_U D'$.*

Assuming that all categories are of equal importance, one can strengthen the above notion by requiring that a preferred diagnosis modifies only the least amount of categories, i.e., prefer diagnoses which modify cardinality minimal sets of units. Cardinality-based preference can drastically reduce the number of diagnoses to be considered. So it may be easier for a human operator, responsible for restoring consistency, to select the best diagnosis.

Definition 4.12. *Let $D, D' \in D^{\pm}(M)$ be diagnoses of an MCS M. D is preferred over D' iff $|U_D| \leq |U_{D'}|$. This is denoted by $D \preceq_{|U|} D'$.*

Notice that $\preceq_{|U|}$ minimizes the number of modified units, which not necessarily is related to the number of modified bridge rules. Further refinements are possible, e.g., such that $\subseteq$-minimal diagnoses among the $\preceq_{|U|}$-minimal ones are preferred.

Example 4.11. *In Example 4.10 we have $U_M = \{treatment, billing\}$, $unit(r_1) = unit(r_2) = \{treatment\}$, $unit(r_3) = unit(r_4) = unit(r_5) = \{billing\}$, and dependency is given by $dep(billing, treatment)$. We obtain for the diagnoses $D = (\{r_1\}, \emptyset)$ and $D' = (\{r_4\}, \emptyset)$ that $U_D = \{treatment, billing\}$ and $U_{D'} = \{billing\}$, hence $D' \preceq_U D$ holds as well as $D' \preceq_{|U|} D$.*

The most-preferred diagnoses given $\preceq_U$ or $\preceq_{|U|}$ are

$$D^{\pm}_{m,\preceq_U}(M) = D^{\pm}_{m,\preceq_{|U|}}(M) = \{(\{r_4\}, \emptyset), (\{r_5\}, \emptyset)\}.$$

4.3 MCS-Realisation

We now present ways to realise filters and preference orders in general, including CP-nets and preferences on units of bridge rules. All realisations use a rewriting technique transforming an MCS M into an extended MCS M', where certain new contexts can do meta-reasoning on diagnoses of the original MCS M. The underlying idea here is that a diagnosis D applied to M' has the same effects as if D would be applied to M, but in M' there are additional contexts which observe the behaviour of the bridge rules taken from M. Hence these observation contexts are enabled to reason on the observed diagnosis D. One significant advantage of this approach is that the observation context may use any abstract logic to reason on the observed diagnosis, hence any formalism that can be captured by an abstract logic may be used for implementing the filter or preference order. Thus our approach can capture a wide range of formalism to specify preferences and it allows the creator of an MCS to use whichever formalism she or he sees to fit best.

We introduce two different transformations, where the idea of the first is to only add bridge rules and contexts to observe the information exchange between contexts of M. The disadvantage of this transformation is that there are MCS where the observation is not able to identify each diagnosis correctly. The second transformation is more general and allows correct identification of diagnoses, but it requires the rewriting of all bridge rules. This rewriting is not intrusive, since it only requires that each rule is duplicated and one additional positive literal added in it.

Both transformation approaches realise filters in general by using diagnoses with protected bridge rules. Since the realisation of preferences is more involved, it is only shown using the second transformation. Preferences also require some additional notions of diagnoses, which allow to prioritise some bridge rules. This notion of prioritised bridge rules in principle establishes a lexicographic order on diagnosis candidates. We furthermore present two possible ways to realise preferences in general using the second transformation. The first one incurs the addition of exponentially many bridge rules, while the second one requires only linearly many additional bridge rules, but comes at the cost of duplicating the original MCS, i.e., each context of the original MCS occurs twice in the resulting MCS.

As we show in Section 4.4, the computational complexity of identifying diagnoses with protected bridge rules is the same as the complexity of identifying regular diagnoses, hence selecting minimal filtered diagnoses is computationally not more expensive than selecting regular minimal diagnoses. Section 4.4 furthermore shows that the complexity of selecting minimal $\preceq$-preferred diagnoses (using the second transformation duplicating the original MCS) is higher than selecting minimal diagnoses, but the complexity of doing so is still worst-case optimal. Hence we show that the complexity of selecting minimal $\preceq$-preferred diagnoses is costly, but our approach to do so is optimal from the perspective of computational complexity.

Furthermore, for preference orders and filters that are not inherently centralised, the realisation allows that preferred solutions are found in a decentralised, localised manner, maintaining privacy and information hiding. Thus we preserve key properties of MCS also for inconsistency assessment and selection of preferred diagnoses.

Meta-Reasoning Transformation

We now present the first transformation to enable meta-reasoning about diagnoses in an MCS. This approach is called the *meta-reasoning transformation*. The objective is to enable the observation of bridge rule applicability, i.e., to have some observation contexts which know whether certain bridge rules are applicable in a belief state. The idea behind this is a follows: given a minimal diagnosis (D_1, D_2) of an inconsistent MCS M, $r \in D_1$ implies that the body of r is satisfied in $M[br(M) \setminus D_1 \cup cf(D_2)]$ while $\varphi(r)$ is not added to the context C_k with $k = C_h(r)$, since r is removed and (D_1, D_2) is a minimal diagnosis. Similarly for $r \in D_2$ it holds that $\varphi(r)$ is added to context C_k with $k = C_h(r)$ while the body of r is not satisfied. Therefore, observing the body and head of a bridge rule is sufficient to detect whether it has been modified by a diagnosis, given that the diagnosis is minimal.

Observing the body of a bridge rule r is possible by use of a protected bridge rule whose body is the same as of r. The observation of the addition of the head formula, however, is not always possible, since the resulting belief set not necessarily exposes information about the (input) knowledge base. The observation of the presence of the head of $\varphi(r)$ requires that there is a belief of C_k with $k = C_h(r)$ which is present in every acceptable belief set of C_i if and only if $\varphi(r)$ is added to the knowledge base of C_i. Note that such a behaviour occurs naturally in many logics, e.g. every context using the logic L_{Σ}^{asp} for Answer-set programs shows this behaviour for all atoms which occur only in the head of a single bridge rule.

To observe all logics, the approach here is a two-step transformation. First, a given MCS M is enlarged with so-called relay contexts to allow the observation of bridge rules. Second, the enlarged/relayed MCS is enhanced with observation contexts which are able to detect the applicability of bridge rules. Furthermore, these contexts can also detect whether and how a bridge rule occurs in a minimal diagnosis.

Relayed Multi-Context Systems

We now present how an MCS can be extended by relay contexts that allow the observation of heads of applicable bridge rules. We first introduce the notion of a *relayed MCS* and then show that its belief states and applicable bridge rules correspond one-to-one to belief states and applicable bridge rules of the original MCS. Furthermore, we show that the same also holds if both systems are modified according to a diagnosis candidate of the original system and the corresponding diagnosis candidate of the relayed system.

All relay contexts are based on a simple abstract logic which behaves similar to an identity function. Formally, given an MCS M, the *relay logic* $L^{\circledR}$ wrt. M is the logic $L^{\circledR} = (2^H, 2^H, \mathbf{ACC}^{\circledR})$ where $H = \{head_r \mid r \in br(M)\}$ contains a new symbol $head_r$ for every bridge rule $r \in br(M)$ and $\mathbf{ACC}^{\circledR}(kb) = \{kb\}$ for any $kb \subseteq H$. Hence a context employing a relay logic exhibits its input knowledge-base formulas as the only acceptable belief set and all bridge rules are identifiable by a separate symbol.

Definition 4.13. *Given an MCS $M = (C_1, \ldots, C_n)$, the corresponding* relayed MCS *is the MCS $M' = (C'_1, \ldots, C'_n, C'_{n+1}, \ldots, C'_{2n})$ where it holds for every $1 \leq i \leq n$ that C'_{n+i} is the relay context of C_i, and C'_i is the relayed context of C_i.*

Formally, for $C_i = (L_i, kb_i, br_i)$, the relay context *of C_i is $C_{n+i} = (L^{\circledR}, \emptyset, br_{n+i})$ and the* relayed context *is $C'_i = (L_i, kb_i, br'_i)$ where $L^{\circledR}$ is the relay logic for M. Furthermore, for every $r \in br_i$ of form* (2.1)*:*

- *$br_{n+i}(M')$ contains the* relayed bridge rule*:*

$$(n+i\colon head_r) \leftarrow (c_1\colon p_1), \ldots, (c_j\colon p_j), \mathbf{not}\ (c_{j+1}\colon p_{j+1}), \ldots, \mathbf{not}\ (c_m\colon p_m). \quad (4.5)$$

- *$br_i(M')$ contains:*

$$(i\colon s) \leftarrow (n+i\colon head_r). \quad (4.6)$$

For convenience, we write $M^{\circledR}$ to denote the corresponding relayed MCS of M. Furthermore, for a bridge rule $r \in br(M)$, we write $r^{\circledR}$ to denote the corresponding relayed bridge rule of form (4.5), which belongs to the relay context, i.e., for $r \in br_i(M)$ it holds that $r^{\circledR} \in br_{n+i}(M^{\circledR})$. We extend this notion also to sets of bridge rules $R \subseteq br(M)$ where $R^{\circledR} = \{r^{\circledR} \mid r \in R\}$. Since $^{\circledR}$ is a bijective mapping, we also use it to map (some of) the bridge rules of $M^{\circledR}$ to bridge rules of M. Hence, for $R \subseteq br(M)$ it holds that $(R^{\circledR})^{\circledR} = R$.

Example 4.12. *Consider the MCS $M = (C_1, C_2)$ of Example 4.3 where $C_1 = (L^{asp}_{\Sigma}, kb_1, \{r_1\})$, $C_2 = (L^{asp}_{\Sigma}, kb_2, \{r_2, r_3\})$, and the bridge rules $br(M)$ of M are:*

$$\begin{aligned}
r_1 &: & (1: improve) &\leftarrow \mathbf{not}\ (2: good). \\
r_2 &: & (2: coauthored) &\leftarrow (1: contribute). \\
r_3 &: & (2: name_K) &\leftarrow (1: contribute).
\end{aligned}$$

The relay version of M is $M^{\circledR} = (C'_1, C'_2, C'_3, C'_4)$ as follows:

$$\begin{aligned}
C'_1 &= (L^{asp}_{\Sigma}, kb_1, \{r_1^{\circledR}\}) & C'_2 &= (L^{asp}_{\Sigma}, kb_2, \{r_2^{\circledR}, r_3^{\circledR}\} \\
C'_3 &= (L^{\circledR}, \emptyset, \{r'_1\}) & C'_4 &= (L^{\circledR}, \emptyset, \{r'_2, r'_3\})
\end{aligned}$$

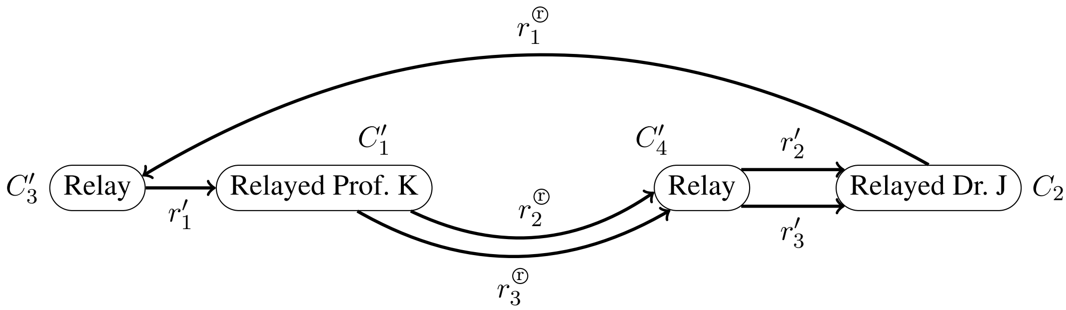


Figure 4.3: The relayed version $M^{\textcircled{r}} = (C'_1, C'_2, C'_3, C'_4)$ of the MCS $M = (C_1, C_2)$ of Example 4.3, which is depicted in Figure 4.2.

The bridge rules of $M^{\textcircled{r}}$ *are as follows:*

$$
\begin{aligned}
r_1^{\textcircled{r}} :&\quad (3 : head_{r_1}) \leftarrow \mathbf{not}\ (2 : good).\\
r'_1 :&\quad (1 : improve) \leftarrow (3 : head_{r_1}).\\
r_2^{\textcircled{r}} :&\quad (4 : head_{r_2}) \leftarrow (1 : contribute).\\
r'_2 :&\quad (2 : coauthored) \leftarrow (4 : head_{r_2}).\\
r_3^{\textcircled{r}} :&\quad (4 : head_{r_3}) \leftarrow (1 : contribute).\\
r'_3 :&\quad (2 : name\_K) \leftarrow (4 : head_{r_3}).
\end{aligned}
$$

Figure 4.3 shows the bridge rules and contexts of $M^{\textcircled{r}}$.

In the following, the MCS resulting from the application of a diagnosis candidate to an MCS occurs many times. Recall that given an MCS M and a diagnosis candidate $(D_1, D_2) \subseteq 2^{br(M)} \times 2^{br(M)}$, $M[D_1, D_2]$ denotes the MCS resulting from the application of (D_1, D_2) to M, i.e., $M[D_1, D_2] = M[br(M) \setminus D_1 \cup cf(D_2)]$.

For applying diagnoses, we extend the notion of relayed bridge rules to unconditional bridge rules as follows: given a bridge rule $r \in br_i(M)$, then $cf(r)^{\textcircled{r}} = cf(r^{\textcircled{r}})$, i.e., $cf(r)^{\textcircled{r}} = (n + i : head_r) \leftarrow .$ where $i = C_h(r)$, and n is the number of contexts of M. Therefore, for $D_1, D_2 \subseteq br(M)$ we obtain that $M^{\textcircled{r}}[D_1^{\textcircled{r}}, D_2^{\textcircled{r}}] = M^{\textcircled{r}}[br_{M^{\textcircled{r}}} \setminus D_1^{\textcircled{r}} \cup \{cf(r^{\textcircled{r}}) \mid r \in D_2\}]$.

Intuitively, the bridge rules of a relay context behave just like the bridge rules of the original context while the bridge rules of the relayed context simply import the information from the relay context. Since every relay has an empty knowledge-base and its acceptance function behaves like the identity function (modulo wrapping the knowledge base into a singleton set), the relayed MCS behaves exactly like the original MCS.

It follows that the accepted belief sets of C_i in M are related one-to-one to the accepted belief sets of C'_i and C'_{n+i} in $M^{\textcircled{r}}$. This even holds under application of a diagnosis (D_1, D_2) to M and application of its relayed variant $(D_1^{\textcircled{r}}, D_2^{\textcircled{r}})$ to $M^{\textcircled{r}}$.

The formal proof of this intuition relies on a one-to-one mapping of accepted belief states $S = (S_1, \ldots, S_n)$ of $M[D_1, D_2]$ to accepted belief states $S' = (S_1, \ldots, S_n, S_{n+1}, \ldots, S_{2n})$ of

$M^{\textcircled{T}}$ such that for every $1 \leq i \leq n$ it holds that $S_{n+i} = \{head_r \mid r \in app(br_i(M[D_1, D_2]), S)\}$. In the following we write $S^{\textcircled{T}}$ to denote the belief state for $M^{\textcircled{T}}$ resulting from the belief state $S = (S_1, \ldots S_n)$ of $M[D_1, D_2]$ as above, i.e.,

$$S^{\textcircled{T}} = (S_1, \ldots, S_n, \{head_r \mid r \in app(br_1(M[D_1, D_2]), S)\}, \ldots, \\ \{head_r \mid r \in app(br_n(M[D_1, D_2]), S)\}).$$

We next show that applying a modification to M and the corresponding modification to $M^{\textcircled{T}}$ results in the same bridge rules being applicable in both MCS.

Lemma 4.1. *Let $M(C_1, \ldots, C_n)$ be an MCS, $R_1, R_2 \subseteq br(M)$ be sets of bridge rules of M, and $S = (S_1, \ldots, S_n)$ be a belief state of M. Then, for every $1 \leq i \leq n$ holds that:*

i. $app(br_i(M[R_1, R_2]), S)^{\textcircled{T}} = app(br_{n+i}(M^{\textcircled{T}}[R_1^{\textcircled{T}}, R_2^{\textcircled{T}}]), S^{\textcircled{T}})$*, and*

ii. $\{\varphi(r) \mid r \in app(br_i(M[R_1, R_2]), S)\} = \{\varphi(r') \mid r' \in app(br_i(M^{\textcircled{T}}[R_1^{\textcircled{T}}, R_2^{\textcircled{T}}]), S^{\textcircled{T}})\}$.

Proof. i. "$\subseteq$": Let $r \in app(br_i(M[br(M) \setminus R_1 \cup cf(R_2)]), S)$. We have to show that $r^{\textcircled{T}} \in app(br_{n+i}(M^{\textcircled{T}}[br_{M^{\textcircled{T}}} \setminus R_1^{\textcircled{T}} \cup cf(R_2^{\textcircled{T}})]), S^{\textcircled{T}})$. Since it holds that $br_i(M[br(M) \setminus R_1 \cup cf(R_2)]) \subseteq br(M) \setminus R_1 \cup cf(R_2)$, we know that $r \in br(M) \setminus R_1$ or $r \in cf(R_2)$.

- Case $r \in br(M) \setminus R_1$: then, $r \in app(br_i(M[br(M) \setminus R_1 \cup cf(R_2)]), S)$, i.e., it holds that $S \mid\!\rightsquigarrow r$. Consider $r^{\textcircled{T}}$ and observe that $(c : p) \in body^+(r^{\textcircled{T}})$, respectively $(c : p) \in body^-(r^{\textcircled{T}})$, implies that $(c : p) \in body^+(r)$, respectively $(c : p) \in body^-(r)$, holds. Since $S \mid\!\rightsquigarrow r$, it follows that $p \in S_c$ for all $(c : p) \in body^+(r)$ and $p' \notin S_{c'}$ for all $(c' : p') \in body^-(r)$. By definition of $S^{\textcircled{T}}$, it follows that $p \in S_c$ for all $(c : p) \in body^+(r^{\textcircled{T}})$ and $p' \notin S_{c'}$ for all $(c' : p') \in body^-(r^{\textcircled{T}})$. Consequently, it holds that $S' \mid\!\rightsquigarrow r^{\textcircled{T}}$, i.e., $r^{\textcircled{T}} \in app(br_{n+i}(M^{\textcircled{T}}[R_1^{\textcircled{T}}, R_2^{\textcircled{T}}]), S^{\textcircled{T}})$.
- Case $r \in cf(R_2)$: then, there exists $r' \in br(M)$ such that $r = cf(r')$ and $r' \in R_2$, therefore it holds that $r'^{\textcircled{T}} \in R_2^{\textcircled{T}}$. By definition of cf, it holds that $cf(r'^{\textcircled{T}}) \in cf(R_2^{\textcircled{T}})$, where $cf(r'^{\textcircled{T}}) = (n + i : \varphi(r')) \leftarrow$. Since $body(cf(r'^{\textcircled{T}})) = \emptyset$, it holds that $cf(r'^{\textcircled{T}}) \in app(br_{n+i}(M^{\textcircled{T}}[R_1^{\textcircled{T}}, R_2^{\textcircled{T}}]), S^{\textcircled{T}})$. Because $cf(r'^{\textcircled{T}}) = (cf(r'))^{\textcircled{T}}$ and $cf(r') = r$, it holds that $r^{\textcircled{T}} \in app(br_{n+i}(M^{\textcircled{T}}[R_1^{\textcircled{T}}, R_2^{\textcircled{T}}]), S^{\textcircled{T}})$.

In summary, if it holds that $r \in app(br_i(M[br(M) \setminus R_1 \cup cf(R_2)]), S)$, then it holds that $r^{\textcircled{T}} \in app(br_{n+i}(M^{\textcircled{T}}[br_{M^{\textcircled{T}}} \setminus R_1^{\textcircled{T}} \cup cf(R_2^{\textcircled{T}})]), S^{\textcircled{T}})$.

"$\supseteq$": Let $r' \in app(br_{n+i}(M^{\textcircled{T}}[br_{M^{\textcircled{T}}} \setminus R_1^{\textcircled{T}} \cup cf(R_2^{\textcircled{T}})]), S^{\textcircled{T}})$, we have to show that there exists $r \in app(br_i(M[br(M) \setminus R_1 \cup cf(R_2)]), S)$ such that $r^{\textcircled{T}} = r'$. Again, we distinguish whether $r' \in br_{M^{\textcircled{T}}} \setminus R_1^{\textcircled{T}}$ or $r' \in cf(R_2^{\textcircled{T}})$.

- Case $r' \in br_{M^{\textcircled{T}}} \setminus R_1^{\textcircled{T}}$: then, $r' \in app(br_{n+i}(M^{\textcircled{T}}), S^{\textcircled{T}})$, i.e., $S^{\textcircled{T}} \mid\!\rightsquigarrow r'$. Observe that $C_b(r') \subseteq \{1, \ldots, n\}$. Since $S^{\textcircled{T}} \mid\!\rightsquigarrow r'$, $(c : p) \in body^+(r')$ implies $p \in S_c$ and

$(c : p) \in \mathit{body}^-(r')$ implies $p \notin S_c$. Consider $r \in \mathit{br}(M)$ such that $r^{\textcircled{T}} = r'$, then $S \models r$ since S and $S^{\textcircled{T}}$ agree on each S_i for $1 \leq i \leq n$ and $(c : p) \in \mathit{body}^+(r)$, respectively $(c : p) \in \mathit{body}^-(r)$, implies $(c : p) \in \mathit{body}^+(r')$, respectively $(c : p) \in \mathit{body}^-(r')$. Consequently, $r \in \mathit{app}(\mathit{br}_i(M), S)$, $r^{\textcircled{T}} \in \mathit{app}(\mathit{br}_i(M), S)^{\textcircled{T}}$, and $r' \in \mathit{app}(\mathit{br}_i(M), S)^{\textcircled{T}}$.

- Case $r' \in \mathit{cf}(R_2^{\textcircled{T}})$: then, there exists $r'' \in R_2^{\textcircled{T}}$ such that $\mathit{cf}(r'') = r'$, and there exists $r \in \mathit{br}(M)$ such that $(r)^{\textcircled{T}} = r''$. By definition of $R_2^{\textcircled{T}}$, it also holds that $r \in R_2$ and $\mathit{body}(\mathit{cf}(r)) = \emptyset$, therefore it holds that $\mathit{cf}(r) \in \mathit{app}(\mathit{br}_i(M[\mathit{br}(M) \setminus R_1 \cup \mathit{cf}(R_2)]), S)$.

  It remains to show that $(\mathit{cf}(r))^{\textcircled{T}} = r'$. From the facts that $r' = \mathit{cf}(r'')$ and $r'' = (r)^{\textcircled{T}}$ it follows that $r' = \mathit{cf}((r)^{\textcircled{T}})$, which is equivalent to $r' = (\mathit{cf}(r))^{\textcircled{T}}$.

This shows that $(\mathit{app}(\mathit{br}_{n+i}(M^{\textcircled{T}}[\mathit{br}_{M^{\textcircled{T}}} \setminus R_1^{\textcircled{T}} \cup \mathit{cf}(R_2^{\textcircled{T}})]), S^{\textcircled{T}})) \subseteq$
$(\mathit{app}(\mathit{br}_i(M[\mathit{br}(M) \setminus R_1 \cup \mathit{cf}(R_2)]), S))^{\textcircled{T}}$ and finally proves item i.

ii. "$\subseteq$": Let $s \in \{\varphi(r) \mid r \in \mathit{app}(\mathit{br}_i(M[R_1, R_2]), S)\}$, i.e., there exists a bridge rule $t \in \mathit{br}_i(M[R_1, R_2])$ such that $t \in \mathit{app}(\mathit{br}_i(M[R_1, R_2]), S)$ and $\varphi(t) = s$. By construction of $M^{\textcircled{T}}$, it then follows that there exists a bridge rule $t' \in \mathit{br}_i(M^{\textcircled{T}}[R_1^{\textcircled{T}}, R_2^{\textcircled{T}}])$ of form (4.6), i.e., $t' = (i : s) \leftarrow (C_{n+i} : \mathit{head}_t)$. Since it holds by definition of $S^{\textcircled{T}} = (S_1, \ldots, S_{2n})$ that $S_{n+i} = \{\mathit{head}_r \mid r \in \mathit{app}(\mathit{br}_i(M[R_1, R_2]), S)\}$, it follows that $\mathit{head}_t \in S_{n+i}$, because $t \in \mathit{app}(\mathit{br}_i(M[R_1, R_2]), S)$. Therefore, $t' \in \mathit{app}(\mathit{br}_i(M^{\textcircled{T}}[R_1^{\textcircled{T}}, R_2^{\textcircled{T}}]), S^{\textcircled{T}})$ and consequently $s \in \{\varphi(r') \mid r' \in \mathit{app}(\mathit{br}_i(M^{\textcircled{T}}[R_1^{\textcircled{T}}, R_2^{\textcircled{T}}]), S^{\textcircled{T}})\}$.

"$\supseteq$": Let $s \in \{\varphi(r') \mid r' \in \mathit{app}(\mathit{br}_i(M^{\textcircled{T}}[R_1^{\textcircled{T}}, R_2^{\textcircled{T}}]), S^{\textcircled{T}})\}$. Hence there exists $t' \in \mathit{br}_i(M^{\textcircled{T}}[R_1^{\textcircled{T}}, R_2^{\textcircled{T}}])$ such that $\varphi(t') = s$ and $S^{\textcircled{T}} \models t'$. By construction of $M^{\textcircled{T}}$, all bridge rules in $\mathit{br}_i(M^{\textcircled{T}}[R_1^{\textcircled{T}}, R_2^{\textcircled{T}}])$ are of form (4.6). Hence without loss of generality, let $t' = (i : s) \leftarrow (n + i : \mathit{head}_t)$ such that $t \in \mathit{br}_i(M)$ and $\varphi(t) = s$. Since $S^{\textcircled{T}} \models t'$, i.e., $\mathit{head}_t \in S_{n+i}$, it follows by definition of S_{n+i} that $t \in \mathit{app}(\mathit{br}_i(M[R_1, R_2]), S)$. Since $\varphi(t) = s$, it follows that $s \in \{\varphi(r) \mid r \in \mathit{app}(\mathit{br}_i(M[R_1, R_2]), S)\}$.

□

We now show that S is an equilibrium of M iff $S^{\textcircled{T}}$ is an equilibrium of $M^{\textcircled{T}}$, and that this relation also holds under bridge rule modifications, i.e., it holds for $M[D_1, D_2]$ and $M^{\textcircled{T}}[D_1^{\textcircled{T}}, D_2^{\textcircled{T}}]$ given that $(D_1, D_2) \in 2^{\mathit{br}(M)} \times 2^{\mathit{br}(M)}$.

Proposition 4.4. *Let M be an MCS, $D_1, D_2 \subseteq \mathit{br}(M)$, and S be a belief state of M. Then S is an equilibrium of $M[D_1, D_2]$ iff $S^{\textcircled{T}}$ is an equilibrium of $M^{\textcircled{T}}[D_1^{\textcircled{T}}, D_2^{\textcircled{T}}]$.*

Proof. Let $M = (C_1, \ldots, C_n)$ and $M^{\textcircled{T}} = (C'_1, \ldots, C'_n, C'_{n+1}, \ldots, C'_{2n})$ where it holds for $1 \leq i \leq n$ that $C'_i = (\Sigma_i, kb_i, br'_i)$ and $C_i = (\Sigma_i, kb_i, br_i)$. Furthermore, let $S = (S_1, \ldots, S_n)$ and let $S^{\textcircled{T}} = (S'_1, \ldots, S'_n, S'_{n+1}, \ldots, S'_{2n})$. Note that $S_i = S'_i$ for $1 \leq i \leq n$.

"$\Rightarrow$": Let $S = (S_1, \ldots, S_n)$ be an equilibrium of $M[D_1, D_2]$, i.e., for all $1 \leq i \leq n$ it holds that $S_i \in \mathbf{ACC}_i(kb_i \cup \{\varphi(r) \mid r \in \mathit{app}(\mathit{br}_i(M[D_1, D_2]), S)\})$. It follows from Lemma 4.1 item i. that $\{\varphi(r) \mid r \in app(br_i(M[D_1, D_2]), S)\} = \{\varphi(r') \mid r' \in app(br_i(M^{\textcircled{T}}[D_1^{\textcircled{T}}, D_2^{\textcircled{T}}]), S^{\textcircled{T}})\}$, hence it holds that context C_i' of $M^{\textcircled{T}}$ accepts $S_i' = S_i$, i.e., $S_i' \in \mathbf{ACC}_i(kb_i \cup \{\varphi(r') \mid r' \in app(br_i(M^{\textcircled{T}}[D_1^{\textcircled{T}}, D_2^{\textcircled{T}}]), S^{\textcircled{T}})\}$.

It remains to show that for $n+1 \leq j \leq 2n$ it holds that $S_j' \in \mathbf{ACC}_j(kb_j \cup \{\varphi(r) \mid r \in app(br_j(M^{\textcircled{T}}[R_1^{\textcircled{T}}, R_2^{\textcircled{T}}]), S^{\textcircled{T}})\})$. By definition of $M^{\textcircled{T}}$, it holds that $C_j = (\Sigma^{\textcircled{T}}, \emptyset, br_j)$, i.e., $S_j \in \mathbf{ACC}_j(kb_j \cup \{\varphi(r) \mid r \in app(br_j(M^{\textcircled{T}}[R_1^{\textcircled{T}}, R_2^{\textcircled{T}}]), S^{\textcircled{T}})\})$ iff $S_j = \{\varphi(r) \mid r \in app(br_j(M^{\textcircled{T}}[R_1^{\textcircled{T}}, R_2^{\textcircled{T}}]), S^{\textcircled{T}})\}$. Therefore, we have to show that

$$\{\varphi(r) \mid r \in app(br_j(M^{\textcircled{T}}[R_1^{\textcircled{T}}, R_2^{\textcircled{T}}]), S^{\textcircled{T}})\} = \{head_r \mid r \in app(br_{j-n}(M[R_1, R_2]), S)\}.$$

Since every $r \in br_j(M^{\textcircled{T}}[R_1^{\textcircled{T}}, R_2^{\textcircled{T}}])$ is of form (4.6), i.e., it holds that $\varphi(r) = head_t$ for some $t \in br_{j-n}(M)$, it is sufficient to show that

$$app(br_j(M^{\textcircled{T}}[R_1^{\textcircled{T}}, R_2^{\textcircled{T}}]), S^{\textcircled{T}}) = app(br_{j-n}(M[R_1, R_2]), S).$$

But this is already shown in Lemma 4.1 i. thus it holds that $S_j' \in \mathbf{ACC}_j(kb_j \cup \{\varphi(r) \mid r \in app(br_j(M^{\textcircled{T}}[R_1^{\textcircled{T}}, R_2^{\textcircled{T}}]), S^{\textcircled{T}})\})$. Therefore, it holds that $S^{\textcircled{T}}$ is an equilibrium of $M^{\textcircled{T}}[R_1^{\textcircled{T}}, R_2^{\textcircled{T}}]$.

"$\Leftarrow$": Let $S^{\textcircled{T}}$ be an equilibrium of $M^{\textcircled{T}}[R_1^{\textcircled{T}}, R_2^{\textcircled{T}}]$. Then it holds for any $1 \leq i \leq n$ that $S_i' \in \mathbf{ACC}_i(kb_i \cup \{\varphi(r) \mid r \in app(br_i(M^{\textcircled{T}}[R_1^{\textcircled{T}}, R_2^{\textcircled{T}}]), S^{\textcircled{T}})$. By Lemma 4.1 ii. it holds that $\{\varphi(r) \mid r \in app(br_i(M^{\textcircled{T}}[R_1^{\textcircled{T}}, R_2^{\textcircled{T}}]), S^{\textcircled{T}})\} = \{\varphi(r) \mid r \in app(br_i(M[R_1, R_2]), S)\}$ and since $S_i' = S_i$, it follows that $S_i \in \mathbf{ACC}_i(kb_i \cup \{\varphi(r) \mid r \in app(br_i(M[R_1, R_2]), S)\})$, i.e., for $1 \leq i \leq n$ it holds that context C_i of $M[R_1, R_2]$ accepts S_i. Consequently, S is an equilibrium of $M[R_1, R_2]$. □

It also holds that there is a one-to-one mapping between equilibria of $M^{\textcircled{T}}$ and equilibria of M.

Corollary 4.1. *Given an MCS M and sets of bridge rules $R_1, R_2 \subseteq br(M)$. If $S' = (S_1', \ldots, S_{2n}')$ is an equilibrium of $M^{\textcircled{T}}[R_1^{\textcircled{T}}, R_2^{\textcircled{T}}]$ then $S = (S_1', \ldots, S_n')$ is an equilibrium of $M[R_1, R_2]$ and $S^{\textcircled{T}} = S'$.*

Proof. For reference, let $S = (S_1, \ldots, S_n) = (S_1', \ldots, S_n')$ and let $M^{\textcircled{T}} = (C_1', \ldots, C_{2n}')$. Let $1 \leq i \leq n$; since C_{n+i}' is a relay context employing the logic $L^{\textcircled{T}}$ and S' is an equilibrium of $M^{\textcircled{T}}[R_1^{\textcircled{T}}, R_2^{\textcircled{T}}]$, it holds that $S_{n+i}' = \{\varphi(r) \mid r \in app(br_{n+i}(M^{\textcircled{T}}[R_1^{\textcircled{T}}, R_2^{\textcircled{T}}]), S')\}$. Furthermore, for all $r \in br_{n+i}(M^{\textcircled{T}}[R_1^{\textcircled{T}}, R_2^{\textcircled{T}}])$ and for all $(c\!:\!p) \in body^{\pm}(r)$ it holds that $1 \leq c \leq n$. Because $S' =_{1,\ldots,n} S$, it follows that $app(br_{n+i}(M^{\textcircled{T}}[R_1^{\textcircled{T}}, R_2^{\textcircled{T}}]), S)$. By the construction of $M^{\textcircled{T}}[R_1^{\textcircled{T}}, R_2^{\textcircled{T}}]$ an analogous reasoning as in the proof of Lemma 4.1 item i. can be applied to show that indeed $app(br_{n+i}(M^{\textcircled{T}}[R_1^{\textcircled{T}}, R_2^{\textcircled{T}}]), S) = app(br_i(M[R_1, R_2]), S)^{\textcircled{T}}$. In summary, it holds for all $1 \leq i \leq n$ that $S_{n+i}' = \{head_r \mid r \in app(br_i(M[R_1, R_2]), S)\}$, i.e., $S' = S^{\textcircled{T}}$.

Since S' is an equilibrium of $M^{\textcircled{T}}[R_1^{\textcircled{T}}, R_2^{\textcircled{T}}]$ it follows directly from Proposition 4.4 that S is an equilibrium of $M[R_1, R_2]$.

□

It follows that there is a one-to-one correspondence of diagnoses of M and $M^{\circledR}$, given that only such pairs of sets of bridge rules of $M^{\circledR}$ are considered where there exists a corresponding pair of sets of bridge rules of M, formally:

Corollary 4.2. *Given an MCS M and $D_1, D_2 \subseteq br(M)$, then $(D_1, D_2) \in D^{\pm}(M)$ holds iff $(D_1^{\circledR}, D_2^{\circledR}) \in D^{\pm}(M^{\circledR})$ holds.*

Proof. Observe that $(D_1, D_2) \in D^{\pm}(M)$ implies that there exists an equilibrium S of $M[D_1, D_2]$. Analogously, $(D_1^{\circledR}, D_2^{\circledR}) \in D^{\pm}(M^{\circledR})$ implies that there exists an equilibrium S' of $M^{\circledR}[D_1^{\circledR}, D_2^{\circledR}]$. By Corollary 4.1 it holds that $S' = S^{\circledR}$ and S is an equilibrium of $M[D_1, D_2]$ iff S' is an equilibrium of $M^{\circledR}[D_1^{\circledR}, D_2^{\circledR}]$, which proves the statement. □

An alternative characterisation of the same pairs is possible via the notion of a diagnosis with protected bridge rules. Let br_P be the set of bridge rules of $M^{\circledR}$ which are of form (4.6), i.e., $br_P = br_{(M^{\circledR})} \setminus (br(M))^{\circledR}$. Then, $(D_1^{\circledR}, D_2^{\circledR}) \in D^{\pm}(M^{\circledR})$ iff there exists $(D_1', D_2') \in D^{\pm}(M, br_P)$ such that $D_1' = D_1^{\circledR}$ and $D_2' = D_2^{\circledR}$.

In the next section the relayed MCS $M^{\circledR}$ is used for meta-reasoning about diagnoses. Since the purpose of the relayed MCS is to observe whether the knowledge-base formula $\varphi(r)$ of a bridge rule r is added to the respective context, relays are necessary in general. For a large class of contexts, however, the relaying is not necessary and either the original MCS M could be used or an MCS where only some contexts are relayed. If a context $C = (L, kb, br)$ with $L = (\mathbf{KB}, \mathbf{BS}, \mathbf{ACC})$ satisfies the following condition, it does not need a relay: for all $\varphi(r)$ exists $b \in \bigcup \mathbf{BS}$ such that for all $H \subseteq \{\varphi(r) \mid r \in br\}$ and for all $S \in \mathbf{ACC}(kb \cup H)$ holds $b \in S$ iff $\varphi(r) \in H$. Informally, for every head-formula of a bridge rule r exists a special belief b matching the presence of r in the set of applicable bridge rules. Note that this only works for contexts whose bridge rule heads are unique, i.e., if it holds for all $r, r' \in br$ with $r \neq r'$ that $\varphi(r) \neq \varphi(r')$.

There is another alternative to solve the issue addressed by the relayed MCS, by changing the knowledge-base of a context instead of adding a relay context. For example, a context $C = (L_{\Sigma}^{asp}, kb, br)$ using ASP can be rewritten to $C' = (L_{\Sigma}^{asp}, kb', br')$ such that the above condition is met by adding auxiliary atoms: let $brhead$ be an new atom which neither occurs in kb nor in $\{\varphi(r) \mid r \in br\}$. Then $kb' = kb \cup \{\varphi(r) \leftarrow brhead(r).\}$ and $br' = \{(C_h(r) : brhead(r)) \leftarrow body(r). \mid r \in r\}$. So, every atom in the head of a bridge rule is changed to a unique ground atom, which can only occur in an accepted belief set of C' iff the atom is added by a bridge rule, i.e., $brhead(r)$ is the atom to identify whether the head of the rule corresponding to r has been added. The effect is that considering only output-projected beliefs, C and C' have the same sets of accepted beliefs. For simplicity of presentation, in the following, we consider relayed MCS only, since this approach also is the most general one.

Observing Diagnoses

We have shown that the relayed version $M^{\circledR}$ of an MCS M exhibits belief states such that the belief set of each relay context exactly gives the set of applicable bridge rules, even if a diagnosis and its corresponding relayed version is applied to M, respectively $M^{\textcircled{T}}$. In the next step, we add observation contexts to $M^{\textcircled{T}}$ to observe the applicability of bridge rules and draw conclusions from these observations.

The observation of a set $B \subseteq br(M)$ of bridge rules of M is possible in $M^{\textcircled{T}}$ by employing any logic which is compatible to bridge rules with head beliefs from $\bigcup_{r \in B}\{body_r, head_r\}$. Let Σ_B be such a logic, then an *observation context* is a tuple (Σ_B, kb_B, B) where kb_B is a Σ_B-knowledge base and $B \subseteq br(M)$. Next, we give the meta-reasoning transformation of M which enables a set O of observation contexts to observe the respective bridge rules, by extending $M^{\textcircled{T}}$ with a new context for each observation context $o \in O$.

Definition 4.14. *Let $M = (C_1, \dots, C_n)$ be an MCS, its relay version $M^{\textcircled{T}} = (C'_1 \dots, C'_{2n})$, and a set of observation contexts $O = \{(\Sigma_{B_1}, kb_{B_1}, B_1), \dots, (\Sigma_{B_k}, kb_{B_k}, B_k)\}$. The* meta-reasoning transformation *of M wrt. O is the MCS $M^O = (C'_1, \dots, C'_{2n}, C_{2n+1}, \dots, C_{2n+k})$ where for each $1 \leq i \leq k$ it holds that $C_{2n+i} = (\Sigma_{B_i}, kb_{B_i}, br_{B_i})$ is a context based on the observation context $(\Sigma_{B_i}, kb_{B_i}, B_i)$ such that its set of bridge rules br_{B_i} contains for every $r \in B_i$ of form* (2.1) *with $C_h(r) = \ell$ the following two bridge rules:*

$$(2n + i : body_r) \leftarrow (c_1 : p_1), \dots, (c_j : p_j), \mathbf{not}\ (c_{j+1} : p_{j+1}), \dots, \mathbf{not}\ (c_m : p_m). \tag{4.7}$$

$$(2n + i : head_r) \leftarrow (n + \ell : head_r). \tag{4.8}$$

We call an observation context $(\Sigma = (\mathbf{KB}, \mathbf{BS}, \mathbf{ACC}), kb, B)$ *conservative* if it accepts every set of applicable bridge rules, i.e., if $\mathbf{ACC}(kb \cup H) \neq \emptyset$ for every $H \subseteq \{head_r, body_r \mid r \in B\}$.

Example 4.13. *Consider the relayed MCS $M^{\textcircled{T}} = (C'_1, C'_2, C'_3, C'_4)$ of Example 4.12 and the observation context $Obs = (L^{asp}_\Sigma, \emptyset, \{r_2, r_3\})$. Note that by choice, the bridge rule r_1 is not observed. The corresponding MCS M^O for $O = \{Obs\}$ is as follows: $M^O = (C'_1, C'_2, C'_3, C'_4, C_5)$ with $C_5 = (L^{asp}_\Sigma, \emptyset, br_5)$ and bridge rules br_5 as follows:*

$$\begin{aligned} br_5 = \{\ & (5 : body_{r_2}) \leftarrow (1 : contribute). \\ & (5 : head_{r_2}) \leftarrow (4 : head_{r_2}). \\ & (5 : body_{r_3}) \leftarrow (1 : contribute). \\ & (5 : head_{r_3}) \leftarrow (4 : head_{r_3}).\} \end{aligned}$$

Figure 4.4 shows the bridge rules and contexts of M^O.

The following theorem shows that equilibria of M and equilibria of the meta-reasoning transformation M^O correspond to each other.

Theorem 4.1. *Given an MCS M, a set of conservative observation contexts O, a pair of sets of bridge rules $R_1, R_2 \subseteq br(M)$, and a belief state $S = (S_1, \dots, S_n)$ for M. Then, S is an equilibrium of $M[R_1, R_2]$ iff there exist belief sets $S_{2n+1}, \dots, S_{2n+|O|}$ such that $S' = (S_1, \dots, S_n, S_{n+1}, \dots, S_{2n+|O|})$ is an equilibrium of $M^O[R^{\textcircled{T}}_1, R^{\textcircled{T}}_2]$.*

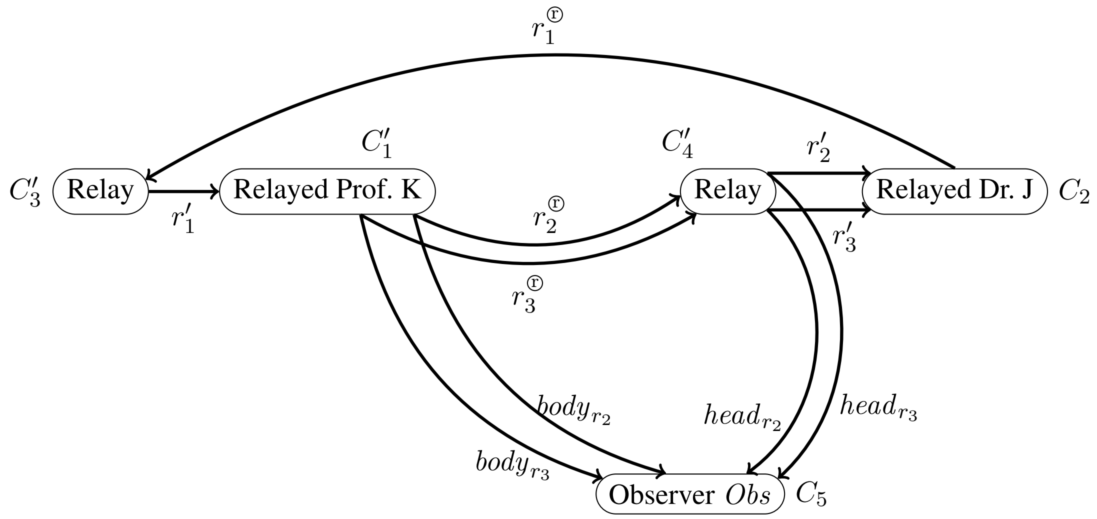


Figure 4.4: The meta-reasoning transformation of the MCS $M = (C_1, C_2)$ of Example 4.3 where bridge rules r_2 and r_3 are observed by C_5.

Proof. "$\Rightarrow$": Let $S = (S_1, \ldots, S_n)$ be an equilibrium of $M[R_1, R_2]$, then $S^{\textcircled{r}}$ is an equilibrium of $M^{\textcircled{r}}[R_1^{\textcircled{r}}, R_2^{\textcircled{r}}]$ by Proposition 4.4. Observe that $U = \{1, \ldots, 2n\} \subseteq C\left(M^O\right)$ is a splitting set of M^O, because for every bridge rule $r \in br_{M^O}$ it holds that $C_b\left(r\right) \cap \{2n+1, \ldots, 2n+|O|\} = \emptyset$. Now consider the MCS $M^O[br_{M^{\textcircled{r}}}]$ where all bridge rules of observation contexts are removed. Obviously, it holds for every $1 \leq i \leq 2n$ that $br_i(M^O) = br_i(M^{\textcircled{r}})$. Consider $S^{\textcircled{r}} = (S_1, \ldots, S_{2n})$ and observe that by Proposition 4.4, it holds that $S^{\textcircled{r}}$ is an equilibrium of $M^{\textcircled{r}}$. Hence, we can extend $S^{\textcircled{r}}$ to a belief state $S'' = (S_1, \ldots, S_{2n}, S_{2n+1}, \ldots S_{2n+|O|})$ where for every $1 \leq j \leq |O|$ it holds that $S_{2n+j} \in \mathbf{ACC}_{2n+j}(kb_{2n+j})$. Since every observation context is conservative, it holds that every such S_{2n+j} exists. By Proposition 4.4 then every context C_i for $1 \leq i \leq 2n$ of $M^O[(br_{M^O} \setminus R_1^{\textcircled{r}} \cup cf(R_2^{\textcircled{r}})) \cap br_{M^{\textcircled{r}}}]$ accepts S_i, which implies that S'' is an equilibrium of $M^O[(br_{M^O} \setminus R_1^{\textcircled{r}} \cup cf(R_2^{\textcircled{r}})) \cap br_{M^{\textcircled{r}}}]$.

By Lemma 3.3 we then conclude that every context C_i for $1 \leq i \leq 2n$ of the MCS M^O accepts S_i. Since observation contexts are conservative, there exists some S' such that $S' = (S_1, \ldots, S_{2n]}, S'_{2n+1}, \ldots, S''_{2n+|O|}$ with $S' =_U S''$ and for every $1 \leq j \leq |O|$ it holds that $S_{2n+j} \in \mathbf{ACC}_{2n+j}(kb_{2n+j} \cup app(br_{2n+j}(M^O[br_{M^O} \setminus R_1^{\textcircled{r}} \cup cf(R_2^{\textcircled{r}})]), S'))$. By Lemma 3.4 (instantiated with $B = br_{M^O} \setminus R_1^{\textcircled{r}} \cup cf(R_2^{\textcircled{r}})$ and $R = (br_{M^O} \setminus R_1^{\textcircled{r}} \cup cf(R_2^{\textcircled{r}})) \cap br_{M^{\textcircled{r}}})$ it then follows that S' is accepted by every context C_i for $1 \leq i \leq 2n$, i.e.,S' is an equilibrium of $M^O[R_1^{\textcircled{r}}, R_2^{\textcircled{r}}]$.

"$\Leftarrow$": Let $S' = (S_1, \ldots, S_{2n+|O|})$ be an equilibrium of $M^O[R_1^{\textcircled{r}}, R_2^{\textcircled{r}}]$. Since $U = \{1, \ldots, 2n\}$ is a splitting set of $M^O[R_1^{\textcircled{r}}, R_2^{\textcircled{r}}]$, it holds by Lemma 3.3 for all $1 \leq i \leq 2n$ that

$$S_i \in \mathbf{ACC}_i(kb_i \cup app(br_i(M^O[(br_{M^O} \setminus R_1^{\textcircled{r}} \cup cf(R_2^{\textcircled{r}})) \cap b_U]), S'))$$

$$\text{iff } S_i \in \mathbf{ACC}_i(kb_i \cup app(br_i(M^O[(br_{M^O} \setminus R_1^{\textcircled{T}} \cup cf(R_2^{\textcircled{T}})], S')).$$

Consider the MCS $M^{\textcircled{T}}[R_1^{\textcircled{T}}, R_2^{\textcircled{T}}]$ and observe that $br_i(M^{\textcircled{T}}[R_1^{\textcircled{T}}, R_2^{\textcircled{T}}]) = br_i(M^O[(br_{M^O} \setminus R_1^{\textcircled{T}} \cup cf(R_2^{\textcircled{T}})) \cap b_U])$ for every $1 \leq i \leq 2n$. For the belief state $S'' = (S_1, \ldots, S_{2n})$ it then follows that $app(br_i(M^{\textcircled{T}}[R_1^{\textcircled{T}}, R_2^{\textcircled{T}}]), S'') = app(br_i(M^O[(br_{M^O} \setminus R_1^{\textcircled{T}} \cup cf(R_2^{\textcircled{T}})) \cap b_U]), S')$ for $1 \leq i \leq 2n$.

Since for every such i, the context C_i is the same in $M^{\textcircled{T}}$ and M^O, it therefore follows that $S_i \in \mathbf{ACC}_i(kb_i \cup \{\varphi(r) \mid r \in app(br_i(M^{\textcircled{T}}[R_1^{\textcircled{T}}, R_2^{\textcircled{T}}]), S'')\})$. Hence, S'' is an equilibrium of $M^{\textcircled{T}}[R_1^{\textcircled{T}}, R_2^{\textcircled{T}}]$, thus by Corollary 4.1 $S'' = S^{\textcircled{T}}$ for $S = (S_1, \ldots, S_n)$, and by Proposition 4.4 it finally holds that S is an equilibrium of $M[R_1, R_2]$. □

The next lemma shows that M^O indeed allows to observe corresponding modifications of bridge rules of M.

Lemma 4.2. *Let M be an MCS, $R_1, R_2 \subseteq br(M)$, and $O = \{O_1, \ldots, O_k\}$ be a set of observation contexts such that $O_i = (\Sigma_o, kb_o, br_o)$ is an observation context with $1 \leq i \leq k$. Let M^O be the meta-reasoning transformation of M wrt. O, let S be a belief state of M and $S^O = (S_1, \ldots, S_{2n+k})$ be a belief state of M^O where $(S_1, \ldots, S_{2n}) = S^{\textcircled{T}}$, and let $H_i = \{\varphi(r) \mid r \in app(br_{2n+i}(M^O[R_1^{\textcircled{T}}, R_2^{\textcircled{T}}]), S^O)$ be the beliefs added to observation context O_i. Then, for every bridge rule $r \in br_o$ observed by O_i with $C_h(r) = j$ holds that:*

- *$S \models\!\!\rightsquigarrow r$ iff $body_r \in H_i$ and*
- *$r \in app(br_j(M[R_1, R_2]), S)$ or $cf(r) \in app(br_j(M[R_1, R_2]), S)$ iff $head_r \in H_i$.*

Proof. Let $K = \{1, \ldots, n\}$ be the index set of contexts of M.

- By definition of M^O, it holds that $C_b\left(r^{\textcircled{T}}\right) = C_b(r) \subseteq \{1, \ldots, n\}$, and by definition of S^O, it holds that $S^O =_K S^{\textcircled{T}} =_K S^O$, i.e., they agree on belief sets $S_1, \ldots, S_n$. Therefore $S \models\!\!\rightsquigarrow r$ iff $S^O \models\!\!\rightsquigarrow r^{\textcircled{T}}$. Since there is exactly one bridge rule r' of form (4.7) wrt. r and O_i, it holds that $\varphi(r') = body_r$ and $body(r') = body(r^{\textcircled{T}})$. Therefore, $S^{\textcircled{T}} \models\!\!\rightsquigarrow r^{\textcircled{T}}$ iff $S^{\textcircled{T}} \models\!\!\rightsquigarrow r'$ iff $body_r \in H_i$. Thus, it holds that $S \models\!\!\rightsquigarrow r$ iff $body_r \in H_i$.
- By Lemma 4.1 it holds that

$$(app(br_j(M[R_1, R_2]), S))^{\textcircled{T}} = app(br_{n+j}(M^{\textcircled{T}}[R_1^{\textcircled{T}}, R_2^{\textcircled{T}}]), S^{\textcircled{T}}).$$

  Since $r \in br_O$ and $br_O \subseteq br(M)$, it holds that $r \in app(br_j(M[R_1, R_2]), S)$ implies that $r \in app(br_j(M[R_1, R_2 \setminus \{r\}]), S)$.

  "⇒": Let $r \in app(br_j(M[R_1, R_2]), S)$. Then $S \models\!\!\rightsquigarrow r$ and by analogous reasoning as above, it holds that $S^O \models\!\!\rightsquigarrow r^{\textcircled{T}}$, hence $head_r \in S_{n+j}$ by definition of S^O and $S^{\textcircled{T}}$. Therefore, by the bridge rule of form (4.8) wrt. r and O_i, it follows that $head_r \in H_i$. Let $cf(r) \in app(br_j(M[R_1, R_2]), S)$, then $r \in R_2$, hence $r^{\textcircled{T}} \in R_2^{\textcircled{T}}$ and consequently $cf(r^{\textcircled{T}}) \in app(br_{2n+i}(M^O[R_1^{\textcircled{T}}, R_2^{\textcircled{T}}]), S^O)$, which implies that $head_r \in H_i$.

"⇐": Let $head_r \in H_i$, then $head_r \in S_{n+j}$ since there is only one bridge rule r' with $\varphi(r') = head_r$, i.e., r' is of form (4.8) wrt. r and O_i. Since $head_r \in S_{n+j}$, it holds by the definition of S^O and $S^{Ⓣ}$, that either $r \in app(br_j(M[R_1, R_2]), S)$ or $cf(r) \in app(br_j(M[R_1, R_2]), S)$.

□

We now introduce a restriction on filters to correctly observe all minimal-filtered diagnoses of those restricted filters. This restriction ensures that the filter does not enforce diagnoses where unnecessary bridge rules or bridge rule bodies are removed, i.e. the filter is deletion parsimonious. Intuitively, a filter is parsimonious if for every diagnosis candidate (D_1, D_2) it holds that either (D_1, D_2) is not a diagnosis that passes the filter, or the resulting MCS admits an equilibrium where all $r \in D_1$ are applicable and all $r \in D_2$ are not applicable.

Definition 4.15. *Let M be an MCS and let f be a filter for M. A pair of bridge rules $(D_1, D_2) \in 2^{br(M)} \times 2^{br(M)}$ is* deletion-parsimonious *iff $f(D_1, D_2) = 1$ and there exists $S \in \mathrm{EQ}(M[D_1, D_2])$ such that $\forall r \in D_1 : S \models\!\!\leadsto r$ and $\forall r \in D_2 : S \not\models\!\!\leadsto r$ both hold.*

The filter f is a deletion-parsimonious filter *if for every $(D_1, D_2) \in D^{\pm}(M)$ it holds that: either (D_1, D_2) is deletion-parsimonious or there exists $(D_1', D_2') \subset (D_1, D_2)$ which is deletion-parsimonious.*

A direct consequence of this notion is the following for every subset-minimal filtered diagnosis.

Corollary 4.3. *Let f be a deletion-parsimonious filter and $(D_1, D_2) \in D^{\pm}_{m,f}(M)$. Then there exists a belief state $S \in \mathrm{EQ}(M[D_1, D_2])$ such that:*

- *every bridge rule $r \in D_1$ is applicable in S, i.e., $S \models\!\!\leadsto r$, and*
- *every bridge rule $r \in D_2$ is not applicable in S, i.e., $S \not\models\!\!\leadsto r$.*

Proof. Let $(D_1, D_2) \in D^{\pm}_{m,f}(M)$ hold. Towards contradiction, assume that for all belief states $S \in \mathrm{EQ}(M[D_1, D_2])$ either there exists some $r \in D_1$ with $S \not\models\!\!\leadsto r$ or there exists some $r \in D_2$ with $S \models\!\!\leadsto r$. Since $(D_1, D_2) \in D^{\pm}_{m,f}(M)$, it holds that $f(D_1, D_2) = 1$ and by the above condition it then holds that (D_1, D_2) is not deletion-parsimonious. Since f is a deletion-parsimonious filter, it then follows that there exists some deletion-parsimonious $(D_1', D_2') \subset (D_1, D_2)$, i.e., $f(D_1', D_2') = 1$ and there exists $S \in \mathrm{EQ}(M[D_1', D_2'])$ such that for all $r \in D_1$ holds $S \models\!\!\leadsto r$ and for all $r \in D_2$ holds $S \not\models\!\!\leadsto r$. Note that S is a witnessing equilibrium of (D_1', D_2'), i.e., $(D_1', D_2') \in D^{\pm}(M)$ holds. $f(D_1', D_2') = 1$, and $(D_1', D_2') \subset (D_1, D_2)$ thus contradicts that $(D_1, D_2) \in D^{\pm}_{m,f}(M)$ also holds. Consequently, it holds for some belief state $S \in \mathrm{EQ}(M[D_1, D_2])$ that for all $r \in D_1$ holds $S \models\!\!\leadsto r$ and for all $r \in D_2$ holds $S \not\models\!\!\leadsto r$. □

Since the notion of a deletion-parsimonious filter not only depends on the filter f, but also on the behaviour of the MCS M under modifications, it is no easy task to check whether a given filter f is deletion-parsimonious. Indeed, restricting to deletion-parsimonious filters excludes many useful filters. Nevertheless, if f is such that $f(D_1, D_2) = 0$ whenever $(D_1, D_2) \notin D^{\pm}_m(M)$

holds, then f is a deletion-parsimonious filter. This holds because only minimal diagnoses pass the filter and by Proposition 3.6 it holds for every belief state $S \in \text{EQ}(M[D_1, D_2])$ that $r \in D_1$ implies $S \models\!\!\leadsto r$ and $r \in D_2$ implies that $S \not\models\!\!\leadsto r$. Intuitively, the filter only selects a subset of the set of $\subseteq$-minimal diagnoses. Another example of a deletion-parsimonious filter is given in Example 4.4.

We now show how deletion-parsimonious filters can be realised and we prove the correctness of that approach. In general, however, filters are not deletion-parsimonious, therefore we later present ways to realise filters in general at the expense of more complex observation contexts.

Definition 4.16. *Let M be an MCS and f be a deletion-parsimonious filter for M. A* parsimonious-filter transformation *of M wrt. f is a meta-reasoning transformation M^O with $O = \{(\Sigma_f, kb_f, br(M))\}$ where the logic $\Sigma_f = (\mathbf{KB}_f, \mathbf{BS}_f, \mathbf{ACC}_f)$ of the observation context is such that*

- *if $kb \in \mathbf{KB}_f$ and $H \subseteq \{body_r, head_r \mid r \in br(M)\}$, then $(kb \cup H) \in \mathbf{KB}_f$, and*
- *for any $H \subseteq \{body_r, head_r \mid r \in br(M)\}$ with $D_1 = \{r \in br(M) \mid body_r \in H \wedge head_r \notin H\}$ and $D_2 = \{r \in br(M) \mid body_r \notin H \wedge head_r \in H\}$, it holds that*

$$\mathbf{ACC}_f\,(kb_f \cup H) = \emptyset \text{ iff } f(D_1, D_2) = 0.$$

By M_f we denote the parsimonious-filter transformation of M wrt. f.

An example of an observation context of a parsimonious filter-transformation is the following based on ASP. Intuitively, the encoding first derives whether the observed bridge rules are removed or made condition-free. Then it checks whether the set of modified bridge rules corresponds to any pair (D_1, D_2) with $f(D_1, D_2) = 0$. A respective ASP program P of an observation context might be as follows:

$$\begin{aligned}
inD1(r) &\leftarrow body_r, not\ head_r. && \forall r \in br(M)\\
inD2(r) &\leftarrow head_r, not\ body_r. && \forall r \in br(M)\\
\bot &\leftarrow filter(D), not\ passes(D). &&\\
passes(D) &\leftarrow filter(D), inD1(r), not\ rfiltD1(D, r). && \forall r \in br(M)\\
passes(D) &\leftarrow filter(D), not\ inD1(r), rfiltD1(D, r). && \forall r \in br(M)\\
passes(D) &\leftarrow filter(D), inD2(r), not\ rfiltD2(D, r). && \forall r \in br(M)\\
passes(D) &\leftarrow filter(D), not\ inD2(r), rfiltD2(D, r). && \forall r \in br(M)
\end{aligned}$$

This assumes that the filter f also is specified explicitly in P. To do so, we assume that "(D_1, D_2)" denotes a unique name for (D_1, D_2), and P then also contains the following facts:

$$\begin{aligned}
&filter(\text{“}(D_1, D_2)\text{”}). && \forall D_1, D_2 \in br(M) \text{ with } f(D_1, D_2) = 0\\
&rfiltD1(\text{“}(D_1, D_2)\text{”}, r). && \forall r \in D_1\\
&rfiltD2(\text{“}(D_1, D_2)\text{”}, r). && \forall r \in D_2
\end{aligned}$$

This encoding is exponential in the size of $br(M)$ in the worst case, since then all diagnosis candidates are specified explicitly by facts. In practice a filter can be represented much more succinctly in ASP by using rules to describe the filter conditions.

The following theorem shows that a parsimonious filter-transformation allows to select the minimal-filtered diagnoses of M wrt. a deletion-parsimonious filter f by selecting minimal protected diagnoses of the transformed MCS M_f.

Theorem 4.2. *Given an inconsistent MCS $M = (C_1, \dots, C_n)$, let f be a deletion-parsimonious filter on diagnoses and let M_f be a parsimonious filter-transformation of M wrt. f, i.e., the protected rules of M_f are $br_P = br(M^O) \setminus (br(M)^{\textcircled{T}})$. Then, $D = (D_1, D_2) \in D^{\pm}_{m,f}(M)$ iff $(D_1^{\textcircled{T}}, D_2^{\textcircled{T}}) \in D^{\pm}_{m}(M_f, br_P)$.*

Proof. Let $M_f = (C_1, \dots, C_{2n}, C_f)$ where C_f is the filter context that observes all bridge rules of $br(M)$.

"$\Rightarrow$": Let $D = (D_1, D_2) \in D^{\pm}_{m,f}(M)$, i.e., $f(D) = 1$. Then by Corollary 4.3 there exists an equilibrium $S = (S_1, \dots, S_n) \in \mathrm{EQ}(M[D_1, D_2])$ such that for every bridge rule $r \in D_1$ it holds that $S \,|\!\!\rightsquigarrow r$ and for every bridge rule $r \in D_2$ it holds that $S \not|\!\!\rightsquigarrow r$.

We first show that $(D_1^{\textcircled{T}}, D_2^{\textcircled{T}}) \in D^{\pm}_{f}(M_f, br_P)$. By Proposition 4.4 it follows that $S^{\textcircled{T}} \in \mathrm{EQ}(M^{\textcircled{T}}[D_1^{\textcircled{T}}, D_2^{\textcircled{T}}])$. Since $M^{\textcircled{T}}[D_1^{\textcircled{T}}, D_2^{\textcircled{T}}]$ induces a splitting set U in $M^O[D_1^{\textcircled{T}}, D_2^{\textcircled{T}}]$ with $U = \{1, \dots, 2n\}$, it follows by Proposition 3.7 that there exists some $S' = (S_1, \dots, S_{2n}, S_f) \in \mathrm{EQ}(M^O[D_1^{\textcircled{T}}, D_2^{\textcircled{T}}]$ with $(S_1, \dots, S_{2n}) = S^{\textcircled{T}}$, if S_f is accepted by C_f. Hence, it remains to show that $S_f \in \mathbf{ACC}_f(kb_f \cup H)$ where $H = \{\varphi(r) \mid r \in app(br_{2n+1}(M^O[D_1^{\textcircled{T}}, D_2^{\textcircled{T}}]), S')\}$. By definition of C_f, it remains to show that: for every $r \in D_1$ it holds that $body_r \in H \land head_r \notin H$ and for every $r \in D_2$ it holds that $body_r \notin H \land head_r \in H$.

Let $r \in D_1$. Then by Corollary 4.3, it holds that $S \,|\!\!\rightsquigarrow r$ which implies by Lemma 4.2 that $body_r \in H$. Since $D_1 \cap D_2 = \emptyset$, it does not hold that $cf(r) \in app(br_j(M[D_1, D_2]), S)$ where $j = C_h(r)$ and since $r \in D_1$, it also does not hold that $r \in app(br_j(M[D_1, D_2]), S)$, hence by Lemma 4.2 it follows that $head_r \notin H$. Let $r \in D_2$. Then $cf(r) \in app(br_j(M[D_1, D_2]), S)$ where $j = C_h(r)$ and by Lemma 4.2 it follows that $head_r \in H$. Furthermore, by Corollary 4.3, it follows that $S \not|\!\!\rightsquigarrow r$, hence by Lemma 4.2 it follows that $body_r \notin H$. In summary, it holds for every $r \in D_1$ that $body_r \in H \land head_r \notin H$ while for every $r \in D_2$ it holds that $body_r \notin H \land head_r \in H$. Therefore, by definition of C_f, it follows that $S_f \in \mathbf{ACC}_f(kb_f \cup H)$ and consequently $S' \in \mathrm{EQ}(M_f[D_1^{\textcircled{T}}, D_2^{\textcircled{T}}])$, i.e., $(D_1^{\textcircled{T}}, D_2^{\textcircled{T}}) \in D^{\pm}_{f}(M_f, br_P)$.

Second, we show that $(D_1^{\textcircled{T}}, D_2^{\textcircled{T}})$ is minimal, i.e., we show that there is no $({D'_1}^{\textcircled{T}}, {D'_2}^{\textcircled{T}}) \subset (D_1^{\textcircled{T}}, D_2^{\textcircled{T}})$ which is in $D^{\pm}_{m}(M_f, br_P)$. Towards contradiction, assume that such $({D'_1}^{\textcircled{T}}, {D'_2}^{\textcircled{T}})$ exists. Then it follows from Corollary 4.2 that $(D'_1, D'_2) \in D^{\pm}(M)$. Since $({D'_1}^{\textcircled{T}}, {D'_2}^{\textcircled{T}}) \in D^{\pm}_{m}(M_f, br_P)$, it holds that there exists an equilibrium S' such that $S' = (S'_1, \dots, S'_{2n}, S'_f) \in \mathrm{EQ}(M_f[{D'_1}^{\textcircled{T}}, {D'_2}^{\textcircled{T}}])$, specifically $S'_f \in \mathbf{ACC}_f(kb_f \cup H)$ where it holds that $H = \{\varphi(r) \mid r \in app(br_f(M_f[{D'_1}^{\textcircled{T}}, {D'_2}^{\textcircled{T}}]), S')\}$. Note that by Corollary 4.1 follows that there exists a belief state S of M such that $S = (S'_1, \dots, S'_n)$. If H correctly "encodes" the diagnosis (D'_1, D'_2), then it follows that $f(D'_1, D'_2) = 1$ and consequently, that $(D'_1, D'_2) \in D^{\pm}_{m,f}(M)$ which then contradicts the assumption that $(D_1, D_2) \in D^{\pm}_{m,f}(M)$ since $(D'_1, D'_2) \subset (D_1, D_2)$.

It therefore remains to show that (i) $\mathit{body}_r \in H \wedge \mathit{head}_r \notin H$ implies $r \in D'_1$ and (ii) $\mathit{body}_r \notin H \wedge \mathit{head}_r \in H$ implies $r \in D'_2$. Let for (i) $\mathit{body}_r \in H$, $\mathit{head}_r \notin H$ be the case. Then $S' \mathrel{|\!\!\leadsto} r'$ where r' is the rule of form (4.7) wrt. r and C_f and $\mathit{body}(r') = \mathit{body}(r^{\textcircled{T}})$. Since $\mathit{head}_r \notin H$, this implies that $\mathit{head}_r \notin S_{n+j}$ where $j = C_h(r)$. Since S' is an equilibrium, it holds by Corollary 4.2 that $\mathit{head}_r \notin \{\varphi(r') \mid r' \in \mathit{app}(\mathit{br}_{n+j}(M_f[D_1'^{\textcircled{T}}, D_2'^{\textcircled{T}}]), S')\}$, which only holds if $r^{\textcircled{T}} \in D_1'^{\textcircled{T}}$. Hence it holds that $r \in D'_1$. Now consider case (ii) and let it hold that $\mathit{body}_r \notin H \wedge \mathit{head}_r \in H$. By Lemma 4.2 then follows that $S \mathrel{|\!\!\not\leadsto} r$, hence $S' \mathrel{|\!\!\not\leadsto} r^{\textcircled{T}}$, because S and S' agree on all belief sets S'_i where $i \in C_b(r) = C_b\left(r^{\textcircled{T}}\right)$. Since $\mathit{head}_r \in H$, it follows by the bridge rule of form (4.8) wrt. r that $\mathit{head}_r \in S_{n+j}$ where $j = C_h(r)$. Since S' is an equilibrium, it holds by Corollary 4.2 that $\mathit{head}_r \in \{\varphi(r') \mid r' \in \mathit{app}(\mathit{br}_{n+j}(M_f[D_1'^{\textcircled{T}}, D_2'^{\textcircled{T}}]), S')\}$. Since $S' \mathrel{|\!\!\not\leadsto} r^{\textcircled{T}}$, this implies that $\mathit{cf}(r^{\textcircled{T}}) \in \mathit{app}(\mathit{br}_{n+j}(M_f[D_1'^{\textcircled{T}}, D_2'^{\textcircled{T}}]), S')$, i.e., $r^{\textcircled{T}} \in D_2'^{\textcircled{T}}$, thus $r \in D'_2$. Since $S_f \in \mathbf{ACC}_f(\mathit{kb}_f \cup H)$ and by the definition of $\mathbf{ACC}_f$, it therefore holds that $f(D'_1, D'_2) = 1$, which implies that $(D'_1, D'_2) \in D^{\pm}_{m,f}(M)$; thus contradicts that $(D_1, D_2) \in D^{\pm}_{m,f}(M)$. Therefore, there exists no such $(D_1'^{\textcircled{T}}, D_2'^{\textcircled{T}}) \in D^{\pm}_m(M_f, \mathit{br}_P)$ and it follows that $(D_1^{\textcircled{T}}, D_2^{\textcircled{T}}) \in D^{\pm}_m(M_f, \mathit{br}_P)$.

"$\Leftarrow$": Let $(D_1^{\textcircled{T}}, D_2^{\textcircled{T}}) \in D^{\pm}_m(M_f, \mathit{br}_P)$. Then there exists an equilibrium S' such that $S' = (S_1, \ldots, S_{2n}, S_f) \in \mathrm{EQ}(M_f[D_1^{\textcircled{T}}, D_2^{\textcircled{T}}])$. We first show that $(D_1, D_2) \in D^{\pm}_f(M)$. Corollary 4.2 implies that $(D_1, D_2) \in D^{\pm}(M)$, therefore it only remains to show that $f(D_1, D_2) = 1$. Consider the input $H = \{\varphi(r) \mid r \in \mathit{app}(\mathit{br}_f(M_f[D_1^{\textcircled{T}}, D_2^{\textcircled{T}}]), S')\}$ of C_f under S'. Note that $S^{\textcircled{T}} = (S_1, \ldots, S_{2n})$ and $S = (S_1, \ldots, S_n)$. We have to show that (i) $\mathit{body}_r \in H \wedge \mathit{head}_r \notin H$ implies $r \in D_1$ and (ii) $\mathit{body}_r \notin H \wedge \mathit{head}_r \in H$ implies $r \in D_2$. For (i) observe that by Lemma 4.2, $S \mathrel{|\!\!\leadsto} r$ and $r \notin \mathit{app}(\mathit{br}_j(M[D_1, D_2]), S)$ where $j = C_h(r)$. Therefore, it holds that $r \in D_1$. For case (ii) it holds by Lemma 4.2 that $S \mathrel{|\!\!\not\leadsto} r$ and either $r \in \mathit{app}(\mathit{br}_j(M[D_1, D_2]), S)$ or $\mathit{cf}(r) \in \mathit{app}(\mathit{br}_j(M[D_1, D_2]), S)$. Since $S \mathrel{|\!\!\not\leadsto} r$, the latter must be the case, i.e., $r \in D_2$. Therefore, H "encodes" (D_1, D_2) correctly and by the definition of $\mathbf{ACC}_f$ it holds that $f((D_1, D_2)) = 1$, hence $(D_1, D_2) \in D^{\pm}_f(M)$.

Second, we show that there exists no $(D'_1, D'_2) \subset (D_1, D_2)$ such that $(D'_1, D'_2) \in D^{\pm}_f(M)$. Towards contradiction, assume that $(D'_1, D'_2) \in D^{\pm}_{m,f}(M)$ with $(D'_1, D'_2) \subset (D_1, D_2)$ exists. In the "$\Rightarrow$"-direction above, it is already proven that $(D'_1, D'_2) \in D^{\pm}_{m,f}(M)$ implies that $(D_1'^{\textcircled{T}}, D_2'^{\textcircled{T}}) \in D^{\pm}_m(M_f, \mathit{br}_P)$. Since $(D_1'^{\textcircled{T}}, D_2'^{\textcircled{T}}) \subset (D_1^{\textcircled{T}}, D_2^{\textcircled{T}})$, this contradicts that $(D_1^{\textcircled{T}}, D_2^{\textcircled{T}}) \in D^{\pm}_m(M_f, \mathit{br}_P)$. Therefore no such $(D'_1, D'_2) \subset (D_1, D_2)$ exists and it holds that $(D_1, D_2) \in D^{\pm}_{m,f}(M)$. $\square$

The restriction to deletion-parsimonious filters excludes a significant amount of possible filters, but a more involved transformation allows to capture all filters. This, however, requires additional guessing on those bridge rules which can not be observed, to accommodate for the fact that the observation of a diagnosis is imperfect.

In the next section another approach at meta-reasoning in MCS is presented where no additional guessing is required. Since the other approach also allows to realise filters and preference orders in general, the correctness proof of this approach is omitted and the guessing approach to realise filters is only presented. Furthermore, we state several observations for filters in general, i.e., conjectures about the correctness of the approach presented in this section.

The application of a parsimonious-filter transformation on an MCS M and a filter f that is not deletion-parsimonious yields a system which cannot identify all minimal-filtered diagnoses, but every obtained diagnosis corresponds to a minimal-filtered diagnosis. Note that any diagnosis (D_1, D_2) obtained from M_f in such a way yields a witnessing equilibrium $S \in \mathrm{EQ}(M[D_1, D_2])$ such that $r \in D_1$ implies $S \models\!\!\leadsto r$ and $r \in D_2$ implies $S \not\models\!\!\leadsto r$.

The expected relationship for an MCS M and any filter f, is then as follows: let M_f be the parsimonious filter-transformation of M wrt. f; then $(D_1^{\circledR}, D_2^{\circledR}) \in D_m^{\pm}(M_f, br_P)$ implies $(D_1, D_2) \in D_{m,f}^{\pm}(M)$.

To identify all minimal-filtered diagnoses using our approach of observing witnessing equilibria, it suffices to guess for those bridge rules $r \in br(M)$ whose status cannot be observed definitely, whether $r \in D_1$, $r \in D_2$, or neither holds. If a minimal such guess yields a diagnosis such that $f(D_1, D_2) = 1$, then this diagnosis is a minimal-filtered one. To guess on these "relatively unknown" bridge rules, we can either introduce additional bridge rules realising the guessing, or we can leave this guessing to the logic employed by the filter context as follows.

Definition 4.17. *Let M be an MCS, let f be a filter, and let O be a singleton set containing an observation context, $O = \{(\Sigma_f, kb_f, br(M))\}$. Then the* filter-transformation *M_f^+ of M wrt. f equals the meta-reasoning transformation M^Osuch that the following holds for the logic $\Sigma_f = (\mathbf{KB}_f, \mathbf{BS}_f, \mathbf{ACC}_f)$:*

- *if $kb \in \mathbf{KB}_f$ and $H \subseteq \{body_r, head_r \mid r \in br(M)\}$, then $(kb \cup H) \in \mathbf{KB}_f$;*
- *for every $bs \in \mathbf{BS}_f$ and for every $B \subseteq \{inD1_r, inD2_r \mid r \in br(M)\}$ there exists $bs \cup B \in \mathbf{BS}_f$;*
- *for every $(D_1, D_2) \in 2^{br(M)} \times 2^{br(M)}$, $H \subseteq \{body_r, head_r \mid r \in br(M)\}$, and $S_f \in \mathbf{BS}_f$ it holds that:*

$$
\begin{aligned}
S_f \in \mathbf{ACC}_f\,(kb_f \cup H) \text{ iff } & f(D_1, D_2) = 1, \\
& D_1 = \{r \mid inD1_r \in S_f\}, D_2 = \{r \mid inD2_r \in S_f\}, \\
& \{r \in br(M) \mid body_r \in H \wedge head_r \notin H\} \subseteq D_1, \text{ and} \\
& \{r \in br(M) \mid body_r \notin H \wedge head_r \in H\} \subseteq D_2
\end{aligned}
$$

- *if $A = \{inD1_r, inD2_r \mid r \in br(M)\}$, then for any $S_f, S'_f \in \mathbf{ACC}_f(kb_f \cup H)$ it holds that: $S_f \cap A \subseteq S'_f \cap A$ implies $S_f = S'_f$.*

The minimal-filtered diagnoses of M can then be obtained by considering the witnessing equilibria of the minimal-diagnoses of the filter-transformation M_f^+. Each such equilibrium contains for the observation context a belief set that corresponds to a minimal-filtered diagnosis. Formally, given an MCS M and a filter f, let M_f^+ be the filter-transformation of M wrt. f.

Furthermore, let $br_P = br(M_f^+) \setminus (br(M))^{\circledR}$, let $A_1 = \{inD1_r \mid r \in br(M)\}$, and let $A_2 = \{inD2_r \mid r \in br(M)\}$. Then the following relationship is conjectured (proof omitted):

$$D^{\pm}_{m,f}(M) = \{(S_f \cap A_1, S_f \cap A_2) \mid (S_1, \dots, S_f) \in \text{EQ}(M_f^+(M[D_1, D_2]), \\ (D_1, D_2) \in D^{\pm}_m(M_f^+, br_P)\}.$$

We now also give a concrete example for the logic and knowledge base of the observation context of M_f^+.

Example 4.14. *We realise the logic used for the observation context in the filter-transformation as a disjunctive ASP program P. Although the computational complexity for disjunctive ASP is in $\mathbf{\Sigma_2^P}$, this does not increase the overall complexity of finding a minimal diagnosis, because this task also is $\mathbf{\Sigma_2^P}$-hard.*

The goal of P is, given some observation, to find all minimal diagnosis candidates which satisfy the filter. Only minimal extensions of the observed diagnosis that still satisfy the filter should lead to an answer set. An extension (D_1, D_2) of the surely modified (observed) bridge rules (R_1, R_2) is minimal, if all other extensions (D'_1, D'_2) between the extension and the observed bridge rules do not satisfy the filter, i.e., for all (D'_1, D'_2) with $(R_1, R_2) \subseteq (D'_1, D'_2) \subset (D_1, D_2)$ it holds that $f(D'_1, D'_2) = 0$. Furthermore, if it holds that $(R_1, R_2) \subseteq (D_1, D_2)$ and $f(D_1, D_2) = 1$, then (D_1, D_2) is such a minimal extension of the observation that satisfies the filter.

To realise the check for all such (D'_1, D'_2) we use the saturation technique similar as used in Section 3.5 with HEX*-programs. This means that there is an atom spoil whose derivation triggers the saturation of the answer set with all atoms possibly used in the guessing of (D'_1, D'_2). If one (D'_1, D'_2) with $f(D'_1, D'_2) = 1$ is found, then this "answer set" is not saturated, since it is a counter-example to the guessed minimal diagnosis candidate (D_1, D_2). Another constraint then ensures that every answer set must be saturated, invalidating the non-minimal (D_1, D_2). If all diagnosis candidates (D'_1, D'_2) are such that $f(D'_1, D'_2) = 0$, then the minimal interpretation of P is the saturated interpretation, which makes it an answer set.*

Let P be the disjunctive ASP program, then P contains for every $r \in br(M)$ the following rules for those bridge rules (R_1, R_2) whose modification is observed, and a guess for the extension (D_1, D_2) of possibly modified bridge rules.

$$
\begin{array}{ll}
inR1_r \leftarrow body_r, not\ head_r. & inR2_r \leftarrow not\ body_r, head_r. \\
inD1_r \leftarrow inR1_r. & inD2_r \leftarrow inR2_r. \\
inD1_r \vee ninD1_r. & inD2_r \vee ninD2_r. \\
\bot \leftarrow ninD1_r, inD1_r. & \bot \leftarrow ninD2_r, inD2_r.
\end{array}
$$

To ensure that (D_1, D_2) fits the observation, the following rules are in P for every $r \in br(M)$:

$$
\begin{array}{ll}
\bot \leftarrow ninD1_r, inD2_r, not\ body_r, not\ head_r. & \bot \leftarrow inD1_r, inD2_r, not\ body_r, not\ head_r. \\
\bot \leftarrow ninD1_r, ninD2_r, body_r, not\ head_r. & \bot \leftarrow ninD1_r, inD2_r, body_r, not\ head_r. \\
\bot \leftarrow inD1_r, inD2_r, body_r, not\ head_r. & \\
\bot \leftarrow ninD1_r, ninD2_r, not\ body_r, head_r. & \bot \leftarrow inD1_r, ninD2_r, not\ body_r, head_r. \\
\bot \leftarrow inD1_r, ninD2_r, body_r, head_r. &
\end{array}
$$

The following rules of P guess bridge rules (D'_1, D'_2) between (R_1, R_2) and (D_1, D_2); for every $r \in br(M)$ there are the following rules:

$$
\begin{array}{ll}
inD'1_r \vee ninD'1_r. & inD'2_r \vee ninD'2_r. \\
inD'1_r \leftarrow inR1_r. & inD'2_r \leftarrow inR2_r. \\
ninD'1_r \leftarrow ninD1_r. & ninD'2_r \leftarrow ninD2_r.
\end{array}
$$

To ensure that (D'_1, D'_2) also fits the observation, the following rules are in P for every $r \in br(M)$:

$$
\begin{array}{ll}
\bot \leftarrow ninD'1_r, inD'2_r, not\ body_r, not\ head_r. & \bot \leftarrow inD'1_r, inD'2_r, not\ body_r, not\ head_r. \\
\bot \leftarrow ninD'1_r, ninD'2_r, body_r, not\ head_r. & \bot \leftarrow ninD'1_r, inD'2_r, body_r, not\ head_r. \\
\bot \leftarrow inD'1_r, inD'2_r, body_r, not\ head_r. & \\
\bot \leftarrow ninD'1_r, ninD'2_r, not\ body_r, head_r. & \bot \leftarrow inD'1_r, ninD'2_r, not\ body_r, head_r. \\
\bot \leftarrow inD'1_r, ninD'2_r, body_r, head_r. &
\end{array}
$$

The next rules of P ensure that $f(D_1, D_2) = 1$ and that the answer set is saturated if $f(D'_1, D'_2) = 0$. For that, we represent the filter f explicitly in such a way that every combination of bridge rules is given as one rule, i.e., we write down each line of the characteristic function that corresponds to f. Formally, P contains for all $A, B \subseteq br(M)$ with $f(A, B) = 0$, $A = \{r_1, \ldots, r_k\}$, $br(M) \setminus A = \{r_{k+1}, \ldots, r_n\}$, $B = \{r'_1, \ldots, r'_\ell\}$, and $br(M) \setminus B = \{r'_{\ell+1}, \ldots, r'_m\}$ the following two rules:

$$
\begin{aligned}
spoil \leftarrow & inD'1_{r_1}, \ldots, inD'1_{r_k}, ninD'1_{r_{k+1}}, \ldots, ninD'1_{r_n}, \\
& inD'2_{r'_1}, \ldots inD'2_{r'_\ell}, ninD'2_{r'_{\ell+1}}, \ldots, ninD'2_{r'_m}. \\
\bot \leftarrow & inD1_{r_1}, \ldots, inD1_{r_k}, ninD1_{r_{k+1}}, \ldots, ninD1_{r_n}, \\
& inD2_{r'_1}, \ldots inD2_{r'_\ell}, ninD2_{r'_{\ell+1}}, \ldots, ninD2_{r'_m}.
\end{aligned}
$$

The following rules in P for every $r \in br(M)$ saturate the interpretation if $spoil$ is derived and they derive $spoil$ if the atoms $inD'1, ninD'1, inD'2, ninD'2$ do not correspond to some

(D'_1, D'_2).

$$inD'1_r \leftarrow spoil. \qquad ninD'1_r \leftarrow spoil$$
$$inD'2_r \leftarrow spoil. \qquad ninD'2_r \leftarrow spoil$$
$$spoil \leftarrow inD'1_r, ninD'1_r \qquad spoil \leftarrow inD'2_r, ninD'2_r$$

Finally, the following constraint of P ensures that every answer set is saturated:

$$\bot \leftarrow not\ spoil.$$

Meta-reasoning Encoding

Instead of observing a (minimally) changed MCS and guessing for not definitely observable modifications, we can encode the modifications of a diagnosis directly in an MCS such that observations are perfect, but the original system is no longer just observed but actively modified instead. Conceptually, given an MCS $M = (C_1, \ldots, C_n)$ all its bridge rules are rewritten and protected such that a diagnosis is applied only to the bridge rules of an additional context C_{n+1}. This context C_{n+1} then is able to definitely observe the modifications and to exhibit this observation to all other contexts via its acceptable belief set.

The bridge rules of the original system are modified to consider the belief set of C_{n+1}. So they either behave like removed or like made unconditional, depending on what C_{n+1} believes. For these two modes of behaviour, each bridge rule $r \in br(M)$ is replaced by two bridge rules in the meta-reasoning system: one bridge rule for becoming unconditional and one that behaves like r or like being removed (i.e., it simply does not fire when C_{n+1} believes that r is removed). The form of these two bridge rules is similar to the form of bridge rules in the HEX-encoding for computing diagnoses (cf. Section 3.5, rules (3.10), (3.16), and (3.17)).

Since this meta-reasoning encoding is used as foundation for filters and preferences, we introduce a property θ that describes the additional behaviour of the context C_{n+1}. This allows to later specify the required behaviour for filters and preferences. The preference encoding requires further bridge rules for mapping preferences to bridge rules; this set of additional bridge rules is called $\mathcal{K}$, so we obtain as meta-reasoning encoding of M an MCS $M^{mr(\theta,\mathcal{K})}$. The definition of $M^{mr(\theta,\mathcal{K})}$ and the following propositions are thus more general than needed for encoding filters only. The advantage of this approach is that we have a common foundation for both encodings and several propositions hold for both encodings. As is later shown in full detail, the property θ to realise a filter f is simply stating that $\theta(D_1, D_2, \emptyset)$ holds iff $f(D_1, D_2) = 1$.

To encode (observe) diagnoses, the context C_{n+1} needs bridge rules where a diagnosis can be applied to and which can be observed reliably. To that end, for every $r \in br(M)$ we have the following two bridge rules to encode/observe whether r is removed or made unconditional.

$$d1(r): \qquad (n{+}1 : not\_removed_r) \leftarrow \top. \tag{4.9}$$
$$d2(r): \qquad (n{+}1 : uncond_r) \leftarrow \bot. \tag{4.10}$$

For a set $R \subseteq br(M)$, let $d1(R) = \{d1(r) \mid r \in R\}$ and $d2(R) = \{d2(r) \mid r \in R\}$. Furthermore, for a set of bridge rules R, we say that the heads of R are *unique*, if it holds for any $r, r' \in R$ that $\varphi(r) = \varphi(r')$ and $C_h(r) = C_h(r')$ implies that $r = r'$. The meta-reasoning encoding $M^{mr(\theta,\mathcal{K})}$ then is as follows.

Definition 4.18. *Let $M = (C_1, \dots, C_n)$ be an MCS, let $\mathcal{K}$ be a set of bridge rules such that the following holds for all $r \in \mathcal{K}$: $body(r) = \{\bot\}$, $C_h(r) = n{+}1$, and for all $r' \in br(M)$ holds $\varphi(r) \neq not\_removed_{r'}$ and $\varphi(r) \neq uncond_{r'}$. Furthermore, let θ be a ternary property over $2^{br(M)} \times 2^{br(M)} \times 2^{\mathcal{K}}$. Then, the MCS $M^{mr(\theta,\mathcal{K})} = (C'_1, \dots, C'_n, C_{n+1})$ is a* meta-reasoning encoding *if the following holds:*

(i) for every $C_i = (L_i, kb_i, br_i)$ with $1 \leq i \leq n$ it holds that $C'_i = (L_i, kb_i, br'_i)$ where br'_i contains for every $r \in br_i$ of form (2.1) *the following two bridge rules:*

$$(i : s) \leftarrow (c_1 : p_1), \dots, (c_j : p_j), \mathbf{not}\ (c_{j+1} : p_{j+1}), \dots, \mathbf{not}\ (c_m : p_m), \mathbf{not}\ (n{+}1 : removed_r). \tag{4.11}$$

$$(i : s) \leftarrow (n{+}1 : uncond_r). \tag{4.12}$$

(ii) $C_{n+1} = (L_{n+1}, kb_{n+1}, br_{n+1})$ is any context such that:

a) $br_{n+1} = d1(br(M)) \cup d2(br(M)) \cup \mathcal{K}$ and the only rules with head formulas $not\_removed_r$ and $uncond_r$ are of form (4.11) *and* (4.12).

b) the semantics $\mathbf{ACC}_{n+1}$ of L_{n+1} fulfills for every $H \subseteq \{\varphi(r) \mid r \in br_{n+1}\}$ that $S_{n+1} \in \mathbf{ACC}_{n+1}(kb_{n+1} \cup H)$ iff $\theta(R_1, R_2, R_3)$ holds where:

$$R_1 = \{r \in br(M) \mid not\_removed_r \notin H\},$$
$$R_2 = \{r \in br(M) \mid uncond_r \in H\},$$
$$R_3 = \{r \in \mathcal{K} \mid \varphi(r) \in H\}, \text{ and}$$
$$S_{n+1} = \{removed_r \mid r \in R_1\} \cup \{uncond_r \mid r \in R_2\}$$

The protected bridge rules br_P of $M^{mr(\theta,\mathcal{K})}$ are all rules of form (4.11) *and* (4.12).

Note that the heads of br_{n+1} are unique, because the bridge rules r of $\mathcal{K}$ are all of the same form except for their head formula $\varphi(r)$ and the remaining bridge rules of br_{n+1} also have unique heads. The condition about acceptable belief sets, namely that $S_{n+1} = \{removed_r \mid r \in R_1\} \cup \{uncond_r \mid r \in R_2\}$ at first seems to be a strong restriction on possible belief sets, since it disallows the occurrence of any other belief. On the other hand, however, the set of output-projected beliefs OUT_{n+1} of context C_{n+1} for every $M^{mr(\theta,\mathcal{K})}$ is such that $OUT_{n+1} = \{removed_r, uncond_r \mid r \in br(M)\}$, i.e., no other belief of C_{n+1} is used by any bridge rule of $M^{mr(\theta,\mathcal{K})}$. We can therefore safely allow that C_{n+1} exhibits other beliefs and all of the following proofs go through.

Example 4.15. *Recall the MCS $M = (C_1, C_2)$ of Example 4.3. Let $\mathcal{K} = \emptyset$ and $\theta(D_1, D_2, \emptyset)$ always hold. Then the meta-reasoning encoding $M^{mr(\theta,\mathcal{K})} = (C'_1, C'_2, C_3)$ is such that the* *context C_1, C_2, equals modulo bridge rules the context C'_1, C'_2, respectively. Recall that the bridge rules of M are:*

$$
\begin{aligned}
r_1 : &\quad (1 : \mathit{improve}) \leftarrow \mathbf{not}\ (2 : \mathit{good}).\\
r_2 : &\quad (2 : \mathit{coauthored}) \leftarrow (1 : \mathit{contribute}).\\
r_3 : &\quad (2 : \mathit{name\_K}) \leftarrow (1 : \mathit{contribute}).
\end{aligned}
$$

The bridge rules of $M^{mr(\theta,\mathcal{K})}$ then are as follows:

$$
\begin{aligned}
r'_1 : &\quad (1 : \mathit{improve}) \leftarrow \mathbf{not}\ (2 : \mathit{good}), \mathbf{not}\ (3 : \mathit{removed}_{r_1}).\\
r''_1 : &\quad (1 : \mathit{improve}) \leftarrow (3 : \mathit{uncond}_{r_1}).\\
&\quad (2 : \mathit{coauthored}) \leftarrow (1 : \mathit{contribute}), \mathbf{not}\ (3 : \mathit{removed}_{r_2}).\\
&\quad (2 : \mathit{coauthored}) \leftarrow (3 : \mathit{uncond}_{r_2}).\\
&\quad (2 : \mathit{name\_K}) \leftarrow (1 : \mathit{contribute}), \mathbf{not}\ (3 : \mathit{removed}_{r_3}).\\
&\quad (2 : \mathit{name\_K}) \leftarrow (3 : \mathit{uncond}_{r_3}).\\
d1(r_1) : &\quad (3 : \mathit{not\_removed}_{r_1}) \leftarrow \top.\\
d2(r_1) : &\quad (3 : \mathit{uncond}_{r_1}) \leftarrow \bot.\\
d1(r_2) : &\quad (3 : \mathit{not\_removed}_{r_2}) \leftarrow \top.\\
d2(r_2) : &\quad (3 : \mathit{uncond}_{r_2}) \leftarrow \bot.\\
d1(r_3) : &\quad (3 : \mathit{not\_removed}_{r_3}) \leftarrow \top.\\
d2(r_3) : &\quad (3 : \mathit{uncond}_{r_3}) \leftarrow \bot.
\end{aligned}
$$

Notice that only the latter half of the bridge rules of $M^{mr(\theta,\mathcal{K})}$ is not protected, i.e., the first six bridge rules are guaranteed to be not modified in a diagnosis with protected bridge rules. Figure 4.5 depicts the contexts and, for better visibility, only those bridge rules of $M^{mr(\theta,\mathcal{K})}$ that stem from $r_1 \in br(M)$ are shown.

The remainder of this subsection is dedicated to prove that $M^{mr(\theta,\mathcal{K})}$ allows to do meta-reasoning on diagnoses of M. The results here are used in the following two subsections to prove that $M^{mr(\theta,\mathcal{K})}$ allows to realise filters and preferences in general.

The following lemma shows that the applicable bridge rules of M under a diagnosis (D_1, D_2) add exactly those knowledge-base elements that are also added under the corresponding diagnosis $(d1(D_1), d2(D_2) \cup K)$ of $M^{mr(\theta,\mathcal{K})}$, where $K \subseteq \mathcal{K}$ is arbitrary.

Lemma 4.3. *Let M be an MCS and $M^{mr(\theta,\mathcal{K})}$ be a meta-reasoning encoding wrt. θ and $\mathcal{K}$. Furthermore, let $D_1, D_2 \subseteq br(M)$, let $K \subseteq \mathcal{K}$, let $S = (S_1, \dots, S_n)$ be a belief state of M, and let $S' = (S_1, \dots, S_n, S_{n+1})$ be a belief state of $M^{mr(\theta,\mathcal{K})}$ where $S_{n+1} = \{\mathit{uncond}_r \mid r \in D_2\} \cup \{\mathit{removed}_r \mid r \in D_1\}$. Then, for all $1 \leq i \leq n$ it holds that $\{\varphi(r) \mid r \in app(br_i(M^{mr(\theta,\mathcal{K})}[d1(D_1), d2(D_2) \cup K]), S')\} = \{\varphi(r) \mid r \in app(br_i(M[D_1, D_2]), S)\}$.*

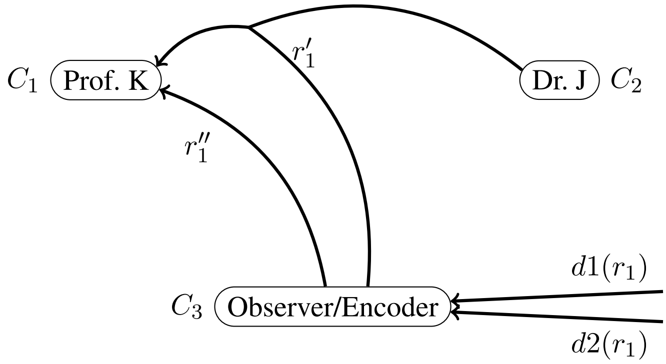


Figure 4.5: Contexts of the meta-reasoning encoding $M^{mr(\theta,\mathcal{K})} = (C_1, C_2, C_3)$ from Example 4.15. Only bridge rules $r_1', r_1'', d1(r_1), d2(r_1)$ of $M^{mr(\theta,\mathcal{K})}$ that stem from bridge rule $r_1 \in br(M)$ are shown.

Proof. Let $D_1, D_2 \subseteq br(M)$, let $K \subseteq \mathcal{K}$, let $S = (S_1, \ldots, S_n)$ be a belief state of M, and let $S' = (S_1, \ldots, S_n, S_{n+1})$ be a belief state of $M^{mr(\theta,\mathcal{K})}$ where $S_{n+1} = \{uncond_r \mid r \in D_2\} \cup \{removed_r \mid r \in D_1\}$. Furthermore, let i be arbitrary such that $1 \leq i \leq n$ holds. We show that $\{\varphi(r) \mid r \in app(br_i(M^{mr(\theta,\mathcal{K})}[d1(D_1), d2(D_2) \cup K]), S')\} = \{\varphi(r) \mid r \in app(br_i(M[D_1, D_2]), S)\}$ holds.

"$\supseteq$": Let $s \in \{\varphi(r) \mid r \in app(br_i(M[D_1, D_2]), S)\}$. Then $s = \varphi(r)$ for some bridge rule r such that either $r \in br(M) \setminus D_1$ and $S \models r$, or $r = cf(r_2)$ where $r_2 \in D_2$. In the former case, consider the bridge rule r_1 of form (4.11) wrt. r. By construction, $body(r_1) = body(r) \cup \{\mathbf{not}\ (n{+}1 : removed_r)\}$ and $\varphi(r_1) = \varphi(r)$. Since $r \notin D_1$, $removed_r \notin S_{n+1}$, and since S and S' agree on S_i for $i \in \{1, \ldots, n\}$, i.e., $S =_{\{1,\ldots,n\}} S'$, it follows that $S' \models r_1$. Therefore $\varphi(r_1) = \varphi(r) = s \in \{\varphi(r) \mid r \in app(br_i(M^{mr(\theta,\mathcal{K})}[d1(D_1), d2(D_2) \cup K]), S')\}$. In the latter case, where $r = cf(r_2)$ and $r_2 \in D_2$ hold, observe that $r_2 \in D_2$ implies that $uncond_{r_2} \in S_{n+1}$. Consider the bridge rule r_2' of form (4.12) wrt. r_2 and observe that $\varphi(r_2') = \varphi(r_2) = s$ while $body(r_2') = \{(n{+}1 : uncond_{r_2})\}$. Since $uncond_{r_2} \in S_{n+1}$, it holds that $S' \models r_2'$, hence $s \in \{\varphi(r) \mid r \in app(br_i(M^{mr(\theta,\mathcal{K})}[d1(D_1), d2(D_2) \cup K]), S')\}$. Thus it follows that $\{\varphi(r) \mid r \in app(br_i(M[D_1, D_2]), S)\} \subseteq \{\varphi(r) \mid r \in app(br_i(M^{mr(\theta,\mathcal{K})}[d1(D_1), d2(D_2) \cup K]), S')\}$.

"$\subseteq$": Let $s \in \{\varphi(r) \mid r \in app(br_i(M^{mr(\theta,\mathcal{K})}[d1(D_1), d2(D_2) \cup K]), S')\}$. Then there exists some $r \in app(br_i(M^{mr(\theta,\mathcal{K})}[d1(D_1), d2(D_2) \cup K]), S')$ such that $s = \varphi(r)$. Note that r either is of form (4.11) or of form (4.12). In the former case, it holds that $S' \models r$ and $removed_{r_1} \notin S_{n+1}$ where $r_1 \in br_i(M)$ and r is the bridge rule of form (4.11) wrt. r_1. Since S and S' agree on all belief sets from S_1 to S_n, i.e., $S =_{\{1,\ldots,n\}} S'$, and $body(r) = body(r_1) \cup \{\mathbf{not}\ (n{+}1 : removed_r)\}$, it holds that $S \models r$. Since $removed_{r_1} \notin S_{n+1}$ it furthermore holds that $r_1 \notin D_1$. This implies that $r_1 \in br_i(M[D_1, D_2])$ and consequently it holds that $r_1 \in app(br_i(M[D_1, D_2]), S)$, thus $s = \varphi(r) = \varphi(r_1) \in \{\varphi(r) \mid r \in app(br_i(M[D_1, D_2]), S)\}$. If r is of form (4.12), $body(r) = \{(n{+}1 : uncond_{r_2})\}$ where $r_2 \in br_i(M)$ and r is the bridge rule of form (4.12) wrt. r_2. Since $r \in app(br_i(M^{mr(\theta,\mathcal{K})}[d1(D_1), d2(D_2) \cup K]), S')$ and $r \notin d2(D_2) \cup K$, it follows that $S' \models r$, hence $uncond_{r_2} \in S_{n+1}$ and thus $r_2 \in D_2$. Therefore, it holds that $cf(r_2) \in app(br_i(M[D_1, D_2]), S)$ and consequently $\varphi(r_2) = \varphi(r) = s \in \{\varphi(r) \mid r \in app(br_i(M[D_1, D_2]), S)\}$. In both cases it holds that $\{\varphi(r) \mid r \in app(br_i(M^{mr(\theta,\mathcal{K})}[d1(D_1), d2(D_2) \cup K]), S')\} \subseteq \{\varphi(r) \mid r \in app(br_i(M[D_1, D_2]), S)\}$. □

The next lemma shows that every protected diagnosis of a meta-reasoning MCS is exhibited in the belief set of the observation context of every witnessing equilibrium of said diagnosis.

Lemma 4.4. *Let $M = (C_1, \ldots, C_n)$ be an MCS and $M^{mr(\theta,\mathcal{K})} = (C_1, \ldots, C_n, C_{n+1})$ be a meta-reasoning encoding. Given that $D_1, D_2 \subseteq br(M)$, $K \subseteq \mathcal{K}$, and $S = (S_1, \ldots, S_n, S_{n+1})$ is a belief state of $M^{mr(\theta,\mathcal{K})}$,*

$$S_{n+1} \in \mathbf{ACC}_{n+1}(kb_{n+1} \cup \{\varphi(r) \mid r \in app(br_{n+1}(M^{mr(\theta,\mathcal{K})}[d1(D_1), d2(D_2) \cup K]), S)\})$$

holds iff $S_{n+1} = \{uncond_r \mid r \in D_2\} \cup \{removed_r \mid r \in D_1\}$ and $\theta(D_1, D_2, K)$ holds.

Proof. By definition of $\mathbf{ACC}_{n+1}$ (cf. Definition 4.18)

$$S_{n+1} \in \mathbf{ACC}_{n+1}(kb_{n+1} \cup \{\varphi(r) \mid r \in app(br_{n+1}(M^{mr(\theta,\mathcal{K})}[d1(D_1), d2(D_2) \cup K]), S)\})$$

holds iff $S_{n+1} = \{removed_r \mid r \in R_1\} \cup \{uncond_r \mid r \in R_2\}$ and $\theta(R_1, R_2, R_3)$ is true, where

$$\begin{aligned}
R_1 &= \{r \in br(M) \mid not_removed_r \notin H\}, \\
R_2 &= \{r \in br(M) \mid uncond_r \in H\}, \\
R_3 &= \{r \in \mathcal{K} \mid \varphi(r) \in H\}, \text{ and} \\
H &= \{\varphi(r) \mid r \in app(br_{n+1}(M^{mr(\theta,\mathcal{K})}[d1(D_1), d2(D_2) \cup K]), S)\}.
\end{aligned}$$

To prove this lemma, it therefore suffices to show that $R_1 = D_1$, $R_2 = D_2$, and $R_3 = K$.

Consider the set B of bridge rules of context C_{n+1} in the MCS resulting from the application of the diagnosis:

$$\begin{aligned}
B &= br_{n+1}(M^{mr(\theta,\mathcal{K})}[d1(D_1), d2(D_2) \cup K]) \\
&= \Big(br_{n+1}(M^{mr(\theta,\mathcal{K})}) \setminus d1(D_1)\Big) \cup cf\big(d2(D_2) \cup K\big) \\
&= \Big(\big(d1(br(M)) \cup d2(br(M)) \cup \mathcal{K}\big) \setminus d1(D_1)\Big) \cup cf\big(d2(D_2) \cup K\big).
\end{aligned}$$

Observe that every bridge rule $r \in B$ is such that either $body(r) = \{\bot\}$ or $body(r) = \{\top\}$. Hence, for any belief state S the set of applicable bridge rules, call it B_{app}, is exactly the set of rules whose body is $\top$. Formally,

$$B_{app} = \{r \in B \mid body(r) = \{\top\}\} = app(br_{n+1}(M^{mr(\theta,\mathcal{K})}[d1(D_1), d2(D_2) \cup K]), S).$$

Recall that $r \in d1(br(M)) \cup d1(D_1) \cup cf\big(d2(D_2) \cup K\big)$ implies that $body(r) = \{\top\}$, while $r \in d2(br(M)) \cup \mathcal{K}$ implies that $body(r) = \{\bot\}$. Therefore,

$$B_{app} = d1(br(M)) \setminus d1(D_1) \cup cf\big(d2(D_2) \cup K\big)$$

and consequently it holds for the set H of heads of applicable bridge rules that

$$\begin{aligned}H &= \{\varphi(r) \mid r \in \mathit{app}(br_{n+1}(M^{mr(\theta,\mathcal{K})}[d1(D_1), d2(D_2) \cup K]), S)\} \\ &= \{\varphi(r) \mid r \in B_{app}\} \\ &= \{\varphi(r) \mid r \in \big(d1(br(M)) \setminus d1(D_1) \cup \mathit{cf}(d2(D_2) \cup K)\big)\} \\ &= \{\mathit{not\_removed}_r \mid r \in br(M) \setminus D_1\} \cup \{\mathit{uncond}_r \mid r \in D_2\} \cup \{\varphi(r) \mid r \in K\}.\end{aligned}$$

Since the heads of br_{n+1} are unique, it holds for any $r_K \in \mathcal{K}$ and $r \in br(M)$ that $\mathit{uncond}_r \neq \varphi(r_K) \neq \mathit{not\_removed}_r$ and it also holds for any $K' \subseteq \mathcal{K}$ that the heads of K' are unique. Consequently, it holds that

$$\begin{aligned}R_1 &= \{r \in br(M) \mid \mathit{not\_removed}_r \notin H\} = \{r \in br(M) \mid r \in D_1\} = D_1 \\ R_2 &= \{r \in br(M) \mid \mathit{uncond}_r \in H = \{r \in br(M) \mid r \in D_2\} = D_2 \\ R_3 &= \{r \in \mathcal{K} \mid \varphi(r) \in H\} = \{r \in \mathcal{K} \mid r \in K\} = K.\end{aligned}$$

Since it only remained to show that $R_1 = D_1$, $R_2 = D_2$, and $R_3 = K$, the lemma is therefore proven. □

The following proposition shows that there is a one-to-one correspondence between diagnoses of M and diagnoses of $M^{mr(\theta,\mathcal{K})}$.

Proposition 4.5. *Let M be an MCS and $M^{mr(\theta,\mathcal{K})}$ be a meta-reasoning encoding with protected bridge rules br_P, and let $D_1, D_2 \subseteq br(M)$, $K \subseteq \mathcal{K}$.*

(1) Let $S = (S_1, \ldots, S_n)$ be a belief state of M and let $S' = (S_1, \ldots, S_n, S_{n+1})$ where $S_{n+1} = \{\mathit{removed}_r \mid r \in D_1\} \cup \{\mathit{uncond}_r \mid r \in D_2\}$. Then, $S \in \mathrm{EQ}(M[D_1, D_2])$ and $\theta(D_1, D_2, K)$ holds iff $S' \in \mathrm{EQ}(M^{mr(\theta,\mathcal{K})}[d1(D_1), d2(D_2) \cup K])$ holds.

(2) $(D_1, D_2) \in D^{\pm}(M)$ and $\theta(D_1, D_2, K)$ hold iff $(d1(D_1), d2(D_2) \cup K) \in D^{\pm}(M^{mr(\theta,\mathcal{K})}, br_P)$ holds.

Proof. (1) Since $S_{n+1} = \{\mathit{uncond}_r \mid r \in D_2\} \cup \{\mathit{removed}_r \mid r \in D_1\}$ and $S' = (S_1, \ldots, S_n, S_{n+1})$, all pre-conditions of Lemma 4.4 and Lemma 4.3 are satisfied; hence we conclude the following.

By Lemma 4.4, $\theta(D_1, D_2, K)$ holds iff

$$S_{n+1} \in \mathbf{ACC}_{n+1}\big(kb_{n+1} \cup \{\varphi(r) \mid r \in \mathit{app}(br_{n+1}(M^{mr(\theta,\mathcal{K})}[d1(D_1), d2(D_2) \cup K]), S')\}\big). \quad (4.13)$$

By Lemma 4.3, for all $1 \leq i \leq n$ holds

$$\{\varphi(r) \mid r \in \mathit{app}(br_i(M^{mr(\theta,\mathcal{K})}[d1(D_1), d2(D_2) \cup K]), S')\} = \{\varphi(r) \mid r \in \mathit{app}(br_i(M[D_1, D_2]), S)\}.$$

which implies that for all $1 \leq i \leq n$ it holds that

$$\mathbf{ACC}_i(kb_i \cup \{\varphi(r) \mid r \in app(br_i(M^{mr(\theta,\mathcal{K})}[d1(D_1), d2(D_2) \cup K]), S'))\}) \\ = \mathbf{ACC}_i(kb_i \cup \{\varphi(r) \mid r \in app(br_i(M[D_1, D_2]), S)\}).$$

This in turn implies that for all $1 \leq i \leq n$, it holds that

$$S_i \in \mathbf{ACC}_i(kb_i \cup \{\varphi(r) \mid r \in app(br_i(M^{mr(\theta,\mathcal{K})}[d1(D_1), d2(D_2) \cup K]), S'))\}) \\ \text{iff } S_i \in \mathbf{ACC}_i(kb_i \cup \{\varphi(r) \mid r \in app(br_i(M[D_1, D_2]), S)\}). \quad (4.14)$$

From (4.14) and (4.13) we therefore obtain that: $\theta(D_1, D_2, K)$ holds and for all $1 \leq i \leq n$ it holds that $S_i \in \mathbf{ACC}_i(kb_i \cup \{\varphi(r) \mid r \in app(br_i(M[D_1, D_2]), S)$ if and only if for all $1 \leq j \leq n+1$ it holds that

$$S_i \in \mathbf{ACC}_i(kb_i \cup \{\varphi(r) \mid r \in app(br_i(M^{mr(\theta,\mathcal{K})}[d1(D_1), d2(D_2) \cup K]), S'))\}).$$

This is equivalent to: $\theta(D_1, D_2, K)$ and $S \in \text{EQ}(M[D_1, D_2])$ hold iff it holds that $S' \in \text{EQ}(M^{mr(\theta,\mathcal{K})}[d1(D_1), d2(D_2) \cup K])$.

(2) This is a direct consequence of (1) and the fact that a diagnosis implies the existence of a witnessing equilibrium and vice versa, i.e., $(D_1, D_2) \in D^{\pm}(M)$ iff there exists a belief state $S \in \text{EQ}(M[D_1, D_2])$, for any M, D_1, D_2, and S. Thus

	$(D_1, D_2) \in D^{\pm}(M)$ and $\theta(D_1, D_2, K)$ hold
iff	$\theta(D_1, D_2, K)$ and $(S_1, \ldots, S_n) \in \text{EQ}(M[D_1, D_2])$ hold
iff (by (1))	$(S_1, \ldots, S_n, S_{n+1}) \in \text{EQ}(M^{mr(\theta,\mathcal{K})}[d1(D_1), d2(D_2) \cup K])$ holds
iff	$(d1(D_1), d2(D_1) \cup K) \in D^{\pm}(M^{mr(\theta,\mathcal{K})})$ holds.

It remains to show that $(d1(D_1), d2(D_1) \cup K) \in D^{\pm}(M^{mr(\theta,\mathcal{K})})$ iff $(d1(D_1), d2(D_1) \cup K) \in D^{\pm}(M^{mr(\theta,\mathcal{K})}, br_P)$. This follows from $(d1(D_1) \cup d2(D_2) \cup K) \cap br_P = \emptyset$ (see Definition 4.18) and Proposition 4.1, which shows that $D^{\pm}(M^{mr(\theta,\mathcal{K})}, br_P) \subseteq D^{\pm}(M^{mr(\theta,\mathcal{K})})$, i.e, every diagnosis with protected bridge rules also is a diagnosis.

□

The following lemma shows that the bridge rules of context C_{n+1} in the MCS $M^{mr(\theta,\mathcal{K})}$ are such that for a minimal diagnosis $(D_1, D_2) \in D^{\pm}_m(M^{mr(\theta,\mathcal{K})}, br_P)$, a bridge rule r with $body(r) = \{\top\}$ is only contained in D_1 (or not modified at all), and a bridge rule r with $body(r) = \{\bot\}$ is only contained in D_2 (or not modified at all).

Lemma 4.5. *Let $M^{mr(\theta,\mathcal{K})}$ be a meta-reasoning encoding with protected bridge rules br_P, and let $(D_1, D_2) \in D^{\pm}_m(M^{mr(\theta,\mathcal{K})}, br_P)$. Then, for every $r \in br(M^{mr(\theta,\mathcal{K})}) \setminus br_P$ holds that:*

(i) $body(r) = \{\top\}$ implies $r \notin D_2$ and

(ii) $body(r) = \{\bot\}$ implies $r \notin D_1$.

Proof. Since $(D_1, D_2) \in D^{\pm}_m(M^{mr(\theta,\mathcal{K})}, br_P)$, there exists a witnessing equilibrium $S \in$ $\text{EQ}(M^{mr(\theta,\mathcal{K})}[D_1, D_2])$ of (D_1, D_2). Since (D_1, D_2) is a diagnosis with protected bridge rules, it holds that $(D_1 \cup D_2) \cap br_P = \emptyset$, which by construction of $M^{mr(\theta,\mathcal{K})}$ implies that $r \in br_{n+1}$.

For a proof by contradiction, we now show the following:

(i) if $body(r) = \{\top\}$ and $r \in D_2$ then $(D_1 \setminus \{r\}, D_2 \setminus \{r\}) \in D^{\pm}(M^{mr(\theta,\mathcal{K})}, br_P)$;

(ii) if $body(r) = \{\bot\}$ and $r \in D_1$ then $(D_1 \setminus \{r\}, D_2) \in D^{\pm}(M^{mr(\theta,\mathcal{K})}, br_P)$.

To show that the respective smaller diagnosis admits a witnessing equilibrium it suffices in the following to consider only applicable bridge rules of C_{n+1}, because it is the only context of $M^{mr(\theta,\mathcal{K})}$ with bridge rules that are not protected.

(i) Case $body(r) = \{\top\}$ and $r \in D_2$. Then

$$\varphi(r) \in \{\varphi(r) \mid r \in app(br_{n+1}(M^{mr(\theta,\mathcal{K})}[D_1, D_2]), S)\}$$

since $cf(r) \in app(br_{n+1}(M^{mr(\theta,\mathcal{K})}[D_1, D_2]), S)$. Now consider $(D_1 \setminus \{r\}, D_2 \setminus \{r\}) \subset (D_1, D_2)$ and observe that $r \in app(br_{n+1}(M^{mr(\theta,\mathcal{K})}[D_1 \setminus \{r\}, D_2 \setminus \{r\}]), S)$ since r is a bridge rule of the modified system and $body(r) = \{\top\}$. Consequently, $S \in \text{EQ}(M^{mr(\theta,\mathcal{K})}[D_1 \setminus \{r\}, D_2 \setminus \{r\}])$ and $(D_1 \setminus \{r\}, D_2 \setminus \{r\}) \in D^{\pm}(M^{mr(\theta,\mathcal{K})}, br_P)$. Note that this reasoning applies regardless of whether $r \in D_1$ holds.

(ii) Case $body(r) = \{\bot\}$ and $r \in D_1$. Then

$$app(br_{n+1}(M^{mr(\theta,\mathcal{K})}[D_1 \setminus \{r\}, D_2]), S) = app(br_{n+1}(M^{mr(\theta,\mathcal{K})}[D_1, D_2]), S)$$

since r either is not applicable (left-hand side), or it is not a bridge rule of the modified MCS (right-hand side). Consequently, $S \in \text{EQ}(M^{mr(\theta,\mathcal{K})}[D_1 \setminus \{r\}, D_2])$ and therefore $(D_1 \setminus \{r\}, D_2) \in D^{\pm}(M^{mr(\theta,\mathcal{K})}, br_P)$.

Each of these statements contradicts that $(D_1, D_2) \in D^{\pm}_m(M^{mr(\theta,\mathcal{K})}, br_P)$, hence the statement of the lemma follows.

□

We can apply the observation of Lemma 4.5 to regular diagnoses and contexts with unique heads by considering $br_P = \emptyset$.

Corollary 4.4. *Let M be an MCS, let $(D_1, D_2) \in D^{\pm}_m(M)$, and let C_i be a context of M. Then, for every $r \in br_i$ it holds that $body(r) = \{\top\}$ implies $r \notin D_1$ and $body(r) = \{\bot\}$ implies $r \notin D_1$.*

Proof. Consider the proof of Lemma 4.4, with $br_P = \emptyset$. The statement here can be proven analogously. □

The following lemma shows that there are no diagnoses in $D^{\pm}_m(M^{mr(\theta,\mathcal{K})}, br_P)$ other than those which correspond to diagnoses of M.

Lemma 4.6. *Let M be an MCS and $M^{mr(\theta,\mathcal{K})}$ be some meta-reasoning encoding for M. For every $(R_1, R_2) \in D^{\pm}_m(M^{mr(\theta,\mathcal{K})}, br_P)$ there exist $D_1, D_2 \subseteq br(M)$ and $K \subseteq \mathcal{K}$ such that $R_1 = d1(D_1)$ and $R_2 = d2(D_2) \cup K$.*

Proof. Recall that br_P contains all bridge rules of form (4.11) and (4.12), hence the only bridge rules not in br_P are those of br_{n+1}, because $br_{M^{mr(\theta,\mathcal{K})}} = br_P \cup br_{n+1}$. Since $br_{n+1} = d1(br(M)) \cup d2(br(M)) \cup \mathcal{K}$, it follows directly that for every $(R_1, R_2) \in D^{\pm}_m(M^{mr(\theta,\mathcal{K})}, br_P)$ there exist $D_1, D'_1, D_2, D'_2 \subseteq br(M)$ and $K, K' \subseteq \mathcal{K}$ such that $R_1 = d1(D_1) \cup d2(D'_1) \cup K'$ and $R_2 = d1(D'_2) \cup d2(D_2) \cup K$. Observe that for all $r \in d2(D'_1) \cup K'$ it holds that $body(r) = \{\bot\}$, hence by Lemma 4.5 it follows that $d2(D'_1) \cup K' = \emptyset$. Furthermore, it holds for all $r \in d1(D'_2)$ that $body(r) = \{\top\}$, hence by Lemma 4.5 it follows that $d1(D'_2) = \emptyset$. Together, this means that $D'_1 = D'_2 = K' = \emptyset$ and therefore it holds for every $(R_1, R_2) \in D^{\pm}_m(M^{mr(\theta,\mathcal{K})}, br_P)$ that there exist $D_1, D_2 \subseteq br(M)$ and $K \subseteq \mathcal{K}$ such that $R_1 = d1(D_1)$ and $R_2 = d2(D_2) \cup K$. □

We can now combine Lemma 4.6 with Proposition 4.5 to establish the correspondence between minimal θ-satisfying diagnoses of M and minimal diagnoses of $M^{mr(\theta,\mathcal{K})}$.

Proposition 4.6. *Let M be an MCS and $M^{mr(\theta,\mathcal{K})}$ be a meta-reasoning encoding, then the set of minimal θ-satisfying diagnoses with protected bridge rules br_P is*

$$\begin{aligned} D^{\pm}_m(M^{mr(\theta,\mathcal{K})}, br_P) = &\{(d1(D_1), d2(D_2) \cup K) \mid (D_1, D_2) \in D^{\pm}(M), \theta(D_1, D_2, K) \text{ holds}, \\ &\big[\nexists (D'_1, D'_2) \in D^{\pm}(M), K' \subseteq \mathcal{K} : \\ &(D'_1, D'_2 \cup K') \subset (D_1, D_2 \cup K) \text{ and } \theta(D'_1, D'_2, K') \text{holds}\big]\}. \end{aligned}$$

Proof. By definition of minimal diagnosis, it holds that

$$\begin{aligned} &D^{\pm}_m(M^{mr(\theta,\mathcal{K})}, br_P) = \\ &\quad \{(R_1, R_2) \mid (R_1, R_2) \in D^{\pm}(M^{mr(\theta,\mathcal{K})}, br_P) \\ &\qquad \text{and there exists no } (R'_1, R'_2) \in D^{\pm}(M^{mr(\theta,\mathcal{K})}, br_P) \\ &\qquad \text{such that } (R'_1, R'_2) \subset (R_1, R_2)\} \end{aligned}$$

By Lemma 4.6, it holds for every $(R_1, R_2) \in D^{\pm}_m(M^{mr(\theta,\mathcal{K})}, br_P)$ that there exist $D_1, D_2 \subseteq br(M)$ and $K \subseteq \mathcal{K}$ such that $R_1 = d1(D_1)$ and $R_2 = d2(D_2) \cup K$, hence we obtain that

$$\begin{aligned} &D^{\pm}_m(M^{mr(\theta,\mathcal{K})}, br_P) = \\ &\{(d1(D_1), d2(D_2) \cup K) \mid (d1(D_1), d2(D_2) \cup K) \in D^{\pm}(M^{mr(\theta,\mathcal{K})}, br_P) \\ &\qquad \text{and there exists no } (d1(D'_1), d2(D'_2) \cup K') \in D^{\pm}(M^{mr(\theta,\mathcal{K})}, br_P) \\ &\qquad \text{such that } (d1(D'_1), d2(D'_2) \cup K') \subset (d1(D_1), d2(D_2) \cup K) \\ &\qquad \text{holds for some } K, K' \subseteq \mathcal{K}\} \end{aligned}$$

By Proposition 4.5 we know that $(d1(D_1), d2(D_2) \cup K) \in D^{\pm}(M^{mr(\theta,\mathcal{K})}, br_P)$ holds iff $(D_1, D_2) \in D^{\pm}(M)$ and $\theta(D_1, D_2, K)$ hold. Therefore we obtain

$$
\begin{aligned}
&D^{\pm}_m(M^{mr(\theta,\mathcal{K})}, br_P) = \\
&\{(d1(D_1), d2(D_2) \cup K) \mid (D_1, D_2) \in D^{\pm}(M) \text{ and } \theta(D_1, D_2, K) \text{ holds} \\
&\qquad \text{and there exists no } (D'_1, D'_2) \in D^{\pm}(M) \text{ such that} \\
&\qquad (d1(D'_1), d2(D'_2) \cup K') \subset (d1(D_1), d2(D_2) \cup K) \text{ and } \theta(D'_1, D'_2, K') \\
&\qquad \text{holds for some } K, K' \subseteq \mathcal{K}\}.
\end{aligned}
$$

Since $d1$ and $d2$ are bijective, $(d1(D'_1), d2(D'_2) \cup K') \subset (d1(D_1), d2(D_2) \cup K)$ holds iff $(D'_1, D'_2 \cup K') \subset (D_1, D_2 \cup K)$ holds.

$$
\begin{aligned}
&D^{\pm}_m(M^{mr(\theta,\mathcal{K})}, br_P) = \\
&\{(d1(D_1), d2(D_2) \cup K) \mid (D_1, D_2) \in D^{\pm}(M) \text{ and } \theta(D_1, D_2, K) \text{ holds} \\
&\qquad \text{and there exists no } (D'_1, D'_2) \in D^{\pm}(M) \text{ such that} \\
&\qquad (D'_1, D'_2 \cup K') \subset (D_1, D_2 \cup K) \text{ and } \theta(D'_1, D'_2, K') \\
&\qquad \text{holds for some } K, K' \subseteq \mathcal{K}\}.
\end{aligned}
$$

□

In the following, we say that θ is *functional* (or *a function*), if for every $D_1, D_2 \subseteq br(M)$ there exists at most one $K \subseteq \mathcal{K}$ such that $\theta(D_1, D_2, K)$ holds. We say that θ is *functional increasing* if θ is functional and if $\theta(D_1, D_2, K)$, $\theta(D'_1, D'_2, K')$, and $(D_1, D_2) \subseteq (D'_1, D'_2)$ implies that $K \subseteq K'$, where $D_1, D_2, D'_1, D'_2 \subseteq br(M), K, K' \subseteq \mathcal{K}$.

For functional increasing θ we can extend the previous lemma.

Lemma 4.7. *Let M be an MCS and $M^{mr(\theta,\mathcal{K})}$ be a meta-reasoning encoding such that θ is functional increasing. Then, the set of minimal θ-satisfying diagnoses with protected bridge rules br_P is*

$$
\begin{aligned}
&D^{\pm}_m(M^{mr(\theta,\mathcal{K})}, br_P) = \\
&\{(d1(D_1), d2(D_2) \cup K) \mid (D_1, D_2) \in D^{\pm}(M) \textit{ and } \theta(D_1, D_2, K) \textit{ holds} \\
&\qquad \textit{and there exists no } (D'_1, D'_2) \in D^{\pm}(M) \textit{ such that} \\
&\qquad (D'_1, D'_2) \subset (D_1, D_2) \textit{ and } \theta(D'_1, D'_2, K') \textit{ holds for some } K, K' \subseteq \mathcal{K}\}
\end{aligned}
$$

Proof. From Proposition 4.6 we know that

$$
\begin{aligned}
D^{\pm}_m(M^{mr(\theta,\mathcal{K})}, br_P) = &\{(d1(D_1), d2(D_2) \cup K) \mid (D_1, D_2) \in D^{\pm}(M), \theta(D_1, D_2, K) \text{ holds}, \\
&\big[\nexists (D'_1, D'_2) \in D^{\pm}(M), K' \subseteq \mathcal{K} : \\
&(D'_1, D'_2 \cup K') \subset (D_1, D_2 \cup K) \text{ and } \theta(D'_1, D'_2, K') \text{holds}\big]\}.
\end{aligned}
$$

Because θ is functional increasing, it holds that $(D'_1, D'_2 \cup K') \subset (D_1, D_2 \cup K)$ holds iff $(D'_1, D'_2) \subset (D_1, D_2)$. We therefore obtain that:

$$\begin{aligned}
&D^{\pm}_m(M^{mr(\theta,\mathcal{K})}, br_P) = \\
&\{(d1(D_1), d2(D_2) \cup K) \mid (D_1, D_2) \in D^{\pm}(M) \text{ and } \theta(D_1, D_2, K) \text{ holds} \\
&\qquad \text{and there exists no } (D'_1, D'_2) \in D^{\pm}(M) \text{ such that} \\
&\qquad (D'_1, D'_2) \subset (D_1, D_2) \text{ and } \theta(D'_1, D'_2, K') \text{ holds for some } K, K' \subseteq \mathcal{K}\}.
\end{aligned}$$

□

Filter Encoding

We use the meta-reasoning encoding to realise filters, by simply requiring that the observation context becomes inconsistent, if the observed diagnosis does not pass the filter, i.e., we put $f(D_1, D_2) = 1$ as the property $\theta(D_1, D_2, K)$ for $K = \emptyset$. Since no further bridge rules are needed to realise filtered diagnoses, we pick $\mathcal{K} = \emptyset$.

Definition 4.19. *Let M be an MCS and let f be a filter. Let $\mathcal{K} = \emptyset$ and let $\theta(D_1, D_2, \emptyset)$ hold iff $f(D_1, D_2) = 1$. Then $M^{mr(\theta,\mathcal{K})}$ is the* filter-encoding *of M wrt. f, which we also denote by M^f.*

Example 4.16. *Reconsider the MCS $M = (C_1, C_2)$ of Example 4.4 where two scientists write a paper and diagnoses are to be filtered by a filter f if the authorship information is modified by a diagnosis in an incoherent way. The filter f (see Example 4.4) is defined as follows:*

$$f(D_1, D_2) = \begin{cases} 0 & \textit{if } r_3 \in D_1, r_2 \notin D_1 \textit{ or } r_3 \notin D_1, r_2 \in D_1 \\ 0 & \textit{if } r_3 \in D_2, r_2 \notin D_2 \textit{ or } r_3 \notin D_2, r_2 \in D_2 \\ 1 & \textit{otherwise} \end{cases}$$

The resulting filter encoding M^f is the MCS $M^{mr(\theta,\mathcal{K})} = (C'_1, C'_2, C_3)$, which has the same shape as the MCS of Example 4.15. It only differs in the contents of the observation/encoding context C_3 which now realises the filter f. We use ASP again for the logic of $C_3 = (L^{asp}_{\Sigma}, kb_3, br_3)$.

Recall that the knowledge-base formulas added by bridge rules to C_3 are either of the form $uncond_r$ or $not\_removed_r$ and this information has to be exposed accordingly in the accepted belief set. Also remember that the definition of the meta-reasoning encoding requires that every accepted belief set only consists of beliefs in $\{removed_r, uncond_r \mid r \in br(M)\}$, but since no other bridge rule of $M^{mr(\theta,\mathcal{K})}$ uses any other belief, we may allow further beliefs in the accepted belief set, i.e., our ASP program may use additional atoms.

The knowledge base kb_3 of C_3 then is:

$$\begin{aligned} kb_3 = \{\, & removed_{r_1} \leftarrow not\ not\_removed_{r_1}.\\ & removed_{r_2} \leftarrow not\ not\_removed_{r_2}.\\ & removed_{r_3} \leftarrow not\ not\_removed_{r_3}.\\ & \bot \leftarrow removed_{r_3}, not\ removed_{r_2}.\\ & \bot \leftarrow not\ removed_{r_3}, removed_{r_2}.\\ & \bot \leftarrow uncond_{r_3}, not\ uncond_{r_2}.\\ & \bot \leftarrow not\ uncond_{r_3}, uncond_{r_2}.\} \end{aligned}$$

The first three rules of kb_3 ensure that the removal information in correct while nothing is needed to ensure that the information about condition-free bridge rules is exposed (if bridge rule r_i is made unconditional, then the fact $uncond_{r_i}$ is added to kb_3 by the bridge rule $d2(r_i) \in br_3(M^{mr(\theta,\mathcal{K})})$ being applicable and hence $uncond_{r_i}$ is also present in the answer set and thus in the belief set of C_3.

The four constraints of kb_3 finally encode the filter condition and they ensure that the context has no acceptable belief set if the corresponding diagnoses are applied.

Observe that the definition of θ follows the definition of f and because f is an abstraction/generalisation of some desired actual behaviour, it is possible to use the desired actual behaviour directly to realise the context C_{n+1} of $M^{mr(\theta,\mathcal{K})}$, i.e., for a concrete use case where some logic is used to describe which diagnoses should be filtered out, it is not really necessary to first abstract the concrete case to a filter f, build θ accordingly and then derive a concrete instantiation of C_{n+1}. Rather, it is sufficient to take the definition of the meta-reasoning encoding and interpret it as the definition of the interfacing between the logic that does the filtering and the rest of the MCS framework. The reason why we introduced filters in general lies in the fact that this allows us to prove that all such filterings can be realised correctly. The following theorem now shows that diagnoses with protected bridge rules of M^f indeed correspond one-to-one to filtered diagnoses of M.

Theorem 4.3. *Let M be an MCS, let f be a filter and let M^f be the corresponding filter-encoding. Then, $D^{\pm}_{m,f}(M) = \{(D_1, D_2) \mid (d1(D_1), d2(D_2)) \in D^{\pm}_m(M^f, br_P)\}$.*

Proof. Recall that $M^f = M^{mr(\theta,\mathcal{K})}$ where θ is defined such that $\theta(D_1, D_2, \emptyset)$ holds iff it holds that $f(D_1, D_2) = 1$, hence θ is functional increasing. By Lemma 4.7 it therefore holds that

$$\begin{aligned} & D^{\pm}_m(M^{mr(\theta,\mathcal{K})}, br_P) = \\ & \{(d1(D_1), d2(D_2) \cup K) \mid (D_1, D_2) \in D^{\pm}(M) \text{ and } \theta(D_1, D_2, K) \text{ holds} \\ & \qquad \text{and there exists no } (D'_1, D'_2) \in D^{\pm}(M) \text{ such that} \\ & \qquad (D'_1, D'_2) \subset (D_1, D_2) \text{ and } \theta(D'_1, D'_2, K') \text{ holds for some } K', K' \subseteq \mathcal{K}\} \end{aligned}$$

which in case of M^f becomes

$$D^{\pm}_m(M^f, br_P) = \{(d1(D_1), d2(D_2)) \mid (D_1, D_2) \in D^{\pm}(M) \text{ and } \theta(D_1, D_2, \emptyset) \text{ holds}$$
$$\text{and there exists no } (D'_1, D'_2) \in D^{\pm}(M) \text{ such that}$$
$$(D'_1, D'_2) \subset (D_1, D_2) \text{ and } \theta(D'_1, D'_2, \emptyset) \text{ holds}\}.$$

By definition of M^f it furthermore holds that $\theta(D_1, D_2, \emptyset)$ holds iff $f(D_1, D_2) = 1$, hence we obtain that

$$D^{\pm}_m(M^f, br_P) = \{(d1(D_1), d2(D_2)) \mid (D_1, D_2) \in D^{\pm}(M) \text{ and } f(D_1, D_2) = 1$$
$$\text{and there exists no } (D'_1, D'_2) \in D^{\pm}(M) \text{ such that}$$
$$(D'_1, D'_2) \subset (D_1, D_2) \text{ and } f(D'_1, D'_2) = 1\}$$

Applying the definition of minimal-filtered diagnoses, we thus obtain that

$$D^{\pm}_m(M^f, br_P) = \{(d1(D_1), d2(D_2)) \mid (D_1, D_2) \in D^{\pm}_{m,f}(M)\}.$$

Note that this statement is equivalent to

$$D^{\pm}_{m,f}(M) = \{(D_1, D_2) \mid (d1(D_1), d2(D_2)) \in D^{\pm}_m(M^f, br_P)\}.$$

□

To obtain all minimal-filtered diagnoses of an MCS M wrt. the filter f, it is therefore sufficient to compute all subset-minimal diagnoses (with protected bridge rules) of the MCS $M^f = M^{mr(\theta,\mathcal{K})}$. Note that this encoding does not come with increased computational cost, since M and M^f have the same number of bridge rules possibly occurring in a diagnosis with protected bridge rules. Consider M^f and the respective bridge rules, i.e., the set $br(M^f) \setminus br_P = d1(br(M)) \cup d2(br(M))$: since $body(r) = \{\top\}$ for $r \in d1(br(M))$ and $body(r) = \{\bot\}$ for $r \in d2(br(M))$, it follows from Corollary 4.4 that for every $(R_1, R_2) \in D^{\pm}_m(M^f, br_P)$ it holds that $r \in R_1 \Rightarrow r \in d1(br(M))$ and $r \in R_2 \Rightarrow r \in d2(br(M))$. Hence, there are $2^{|d1(br(M))|} \times 2^{|d2(br(M))|}$ possibly relevant diagnoses for M^f while there are $2^{|br(M)|} \times 2^{|br(M)|}$ possible diagnoses for M; since $|d1(br(M))| = |d2(br(M))| = |br(M)|$, the problem size for deciding whether a minimal-filtered diagnosis exists for M is the same as the problem size for deciding whether a minimal diagnosis with protected bridge rules exists for M^f.

Preference Encoding

We now show how to use the meta-reasoning encoding $M^{mr(\theta,\mathcal{K})}$ for realising preferences. The set $\mathcal{K}$ used in the meta-reasoning encoding plays a crucial role, since it is used to map a given preference order on diagnoses to the $\subseteq$ relation on $\mathcal{K}$. This allows to select minimal $\preceq$-preferred diagnoses by considering $\subseteq$-minimal diagnoses of $M^{mr(\theta,\mathcal{K})}$. Since the $\subseteq$-minimality on $\mathcal{K}$ should take precedence over the remaining modified bridge rules of $M^{mr(\theta,\mathcal{K})}$, we introduce a lexicographic order on bridge rules in which the latter are behind those of $\mathcal{K}$. As we show in Section 4.4, the complexity of identifying a diagnosis with respect to prioritised bridge rules $\mathcal{K}$ is not higher than identifying a minimal diagnosis.

In the remainder of this section, we present two approaches to realise preferences. The first approach is plain and simple, but comes at the cost of $\mathcal{K}$ being exponentially larger than $br(M)$, i.e., $M^{mr(\theta,\mathcal{K})}$ contains exponentially many more bridge rules than M. We also prove that the approach is correct for total preference orders. The second approach adds only linearly many bridge rules, specifically it holds for this approach that $|\mathcal{K}| = 4|br(M)| + 1$, but it requires that the original MCS M is cloned. So, first an MCS $2.M$ is built which consists of two independent copies of M, and then the meta-reasoning encoding is applied on $2.M$, i.e., the resulting MCS is $(2.M)^{mr(\theta,\mathcal{K})}$. We show that minimal $\preceq$-preferred diagnoses can be selected from $(2.M)^{mr(\theta,\mathcal{K})}$ using this MCS and a slightly more involved diagnosis with prioritised bridge rules. The complexity of selecting these diagnoses increases, but as it is later shown, it is still worst-case optimal.

We come now to define the notion of a diagnosis with protected bridge rules and then continue with the plain meta-reasoning encoding for total preference orders. In the following, we write $(D_1, D_2) \subseteq_{br_H} (D'_1, D'_2)$ as shorthand for $(D_1 \cap br_H, D_2 \cap br_H) \subseteq (D'_1 \cap br_H, D'_2 \cap br_H)$, i.e., we denote by $\subseteq_{br_H}$ the restriction of $\subseteq$ to the set br_H; furthermore, we write $=_{br_H}$ for an analogous restriction on $=$. To realise a total preference order, the following definition is sufficient where we select from the set of minimal diagnoses with protected bridge rules those that are minimal with respect to the prioritised bridge rules. The bridge rules that are marked as prioritised take precedence for minimality. A prioritised-minimal diagnosis is subset-minimal with respect to prioritised bridge rules (regardless of minimality of the remaining bridge rules).

Definition 4.20. *Let M be an MCS with bridge rules $br(M)$, protected rules $br_P \subseteq br(M)$, and prioritised rules $br_H \subseteq br(M)$. The set of* prioritised-minimal *diagnoses is*

$$D^{\pm}(M, br_P, br_H) = \big\{D \in D^{\pm}_m(M, br_P) \big| \forall D' \in D^{\pm}_m(M, br_P) : D' \subseteq_{br_H} D \Rightarrow D' =_{br_H} D\big\}.$$

Before showing the plain preference encoding, we show how an arbitrary order relation over a pair of sets may be mapped to the $\subseteq$-relation on an exponentially larger set, i.e., we map $\preceq$ on diagnoses of an MCS M, to another set which is exponentially larger than the set of diagnoses of M.

Definition 4.21. *Let $\preceq$ be a preference relation on $2^{br(M)} \times 2^{br(M)}$ and let $g : 2^{br(M)} \times 2^{br(M)} \to \mathcal{K}$ be a bijective mapping where $\mathcal{K}$ is arbitrary. Then, the* subset-mapping *$map^g_{\preceq} : 2^{br(M)} \times 2^{br(M)} \to 2^{\mathcal{K}}$ is defined as follows. For every $(D_1, D_2) \in 2^{br(M)} \times 2^{br(M)}$:*

$$map^g_{\preceq}(D_1, D_2) = \big\{K \in \mathcal{K} \mid K = g(D'_1, D'_2) \text{ for some } (D'_1, D'_2) \preceq (D_1, D_2)\big\} \cup \big\{g(D_1, D_2)\big\}.$$

Observe that $map^g_{\preceq}(D_1, D_2)$ collects $g(D'_1, D'_2)$ of all (D'_1, D'_2) "below" (D_1, D_2). Furthermore, by adding $g(D_1, D_2)$ it establishes reflexivity regardless of the reflexivity of $\preceq$.

The following lemma shows that the subset-mapping correctly maps a preference relation on diagnoses to the subset-relation on an exponentially larger set. This allows to decide whether a diagnosis is more preferred than another solely based on subset relationship.

Lemma 4.8. *Let $\preceq$ be a preference on diagnosis candidates of an MCS M, let $\mathcal{K}$ be a set, and let g be a bijective mapping $g : 2^{br(M)} \times 2^{br(M)} \to \mathcal{K}$. Then, for any $(D_1, D_2) \neq (D'_1, D'_2) \in 2^{br(M)} \times 2^{br(M)}$ it holds that $(D_1, D_2) \preceq (D'_1, D'_2)$ iff $map^g_{\preceq}(D_1, D_2) \subseteq map^g_{\preceq}(D'_1, D'_2)$.*

Proof. "$\Rightarrow$": Suppose that $(D_1, D_2) \preceq (D'_1, D'_2)$. We have to show that for every $K \in map^g_{\preceq}(D_1, D_2)$ it holds that $K \in map^g_{\preceq}(D'_1, D'_2)$. Let $K \in map^g_{\preceq}(D_1, D_2)$ hold. Then it follows by definition that $K = g(D''_1, D''_2)$ for some $(D''_1, D''_2) \in 2^{br(M)} \times 2^{br(M)}$. In the case that $(D''_1, D''_2) = (D_1, D_2)$ it trivially follows that $(D''_1, D''_2) \preceq (D'_1, D'_2)$ and thus by definition of $map^g_{\preceq}(D'_1, D'_2)$ it holds that $K \in map^g_{\preceq}(D'_1, D'_2)$. In the case that $(D''_1, D''_2) \neq (D_1, D_2)$ it follows by the definition of $map^g_{\preceq}(D_1, D_2)$ that $(D''_1, D''_2) \preceq (D_1, D_2)$. Since $(D_1, D_2) \preceq (D'_1, D'_2)$ and $\preceq$ is transitive, it follows that $(D''_1, D''_2) \preceq (D'_1, D'_2)$ and consequently, it holds that $K \in map^g_{\preceq}(D'_1, D'_2)$. Thus for any $K \in map^g_{\preceq}(D_1, D_2)$ it holds that $K \in map^g_{\preceq}(D'_1, D'_2)$, i.e., $map^g_{\preceq}(D_1, D_2) \subseteq map^g_{\preceq}(D'_1, D'_2)$.

"$\Leftarrow$": Suppose that $map^g_{\preceq}(D_1, D_2) \subseteq map^g_{\preceq}(D'_1, D'_2)$. We have to show that $(D_1, D_2) \preceq (D'_1, D'_2)$. By definition $g(D_1, D_2) \in map^g_{\preceq}(D_1, D_2)$ and hence $g(D_1, D_2) \in map^g_{\preceq}(D'_1, D'_2)$. By definition of $map^g_{\preceq}(D'_1, D'_2)$ and since $(D_1, D_2) \neq (D'_1, D'_2)$, it then follows that $(D_1, D_2) \preceq (D'_1, D'_2)$. □

We now use $map^g_{\preceq}$ to map the preference of a total order $\preceq$ to the set $\mathcal{K}$ which occurs in the meta-reasoning transformation $M^{mr(\theta,\mathcal{K})}$. To that end, we choose $\theta(D_1, D_2, K)$ such that it holds iff $map^g_{\preceq}(D_1, D_2) = K$. By that, every diagnosis with protected bridge rules $(d1(D_1), d2(D_2) \cup K)$ of $M^{mr(\theta,\mathcal{K})}$ contains the preference $\preceq$ encoded in K. Selecting a diagnosis of $M^{mr(\theta,\mathcal{K})}$ where K is minimal then selects a preferred diagnosis according to $\preceq$.

Definition 4.22. *Let M be an MCS and let $\preceq$ be a preference relation. Furthermore, let*

$$\mathcal{K} = \{(n{+}1 : diag_{D_1,D_2}) \leftarrow \bot. \mid D_1, D_2 \subseteq br(M)\} \tag{4.15}$$

and let g be a bijective function such that $g(D_1, D_2) = (n{+}1 : diag_{D_1,D_2}) \leftarrow \bot.$ for all $D_1, D_2 \subseteq br(M)$. Let $\theta(D_1, D_2, K)$ hold iff $map^g_{\preceq}(D_1, D_2) = K$. Then the MCS $M^{mr(\theta,\mathcal{K})}$ is called the plain encoding *of M wrt. $\preceq$, which we also denote by $M^{pl\preceq}$; all bridge rules of $\mathcal{K}$ are prioritised, i.e., $br_H = \mathcal{K}$.*

Note that since $map^g_{\preceq}$ is a function, also θ is equivalent to a function $2^{br(M)} \times 2^{br(M)} \to \mathcal{K}$.

Example 4.17. *We consider the hospital MCS M of Example 4.1 again using a preference order on diagnoses that is similar to the one of Example 4.6, i.e., we prefer diagnoses that change the bridge rules regarding health, r_1, r_2, as little as possible. To make the preference of the latter example total, we use cardinality-minimality, i.e., given $(D_1, D_2), (D'_1, D'_2) \in 2^{br(M)} \times 2^{br(M)}$ the preference order $\preceq$ is such that:*

$$(D_1, D_2) \preceq (D'_1, D'_2) \text{ iff } \big|\{r_1, r_2\} \cap (D_1 \cup D_2)\big| \leq \big|(D'_1 \cup D'_2) \cap \{r_1, r_2\}\big|$$

The resulting MCS $M^{mr(\theta,\mathcal{K})}$ is shown in Figure 4.6, where for illustration purposes only bridge rules stemming from $r_5 \in br(M)$ and some of the bridge rules of the observation context,

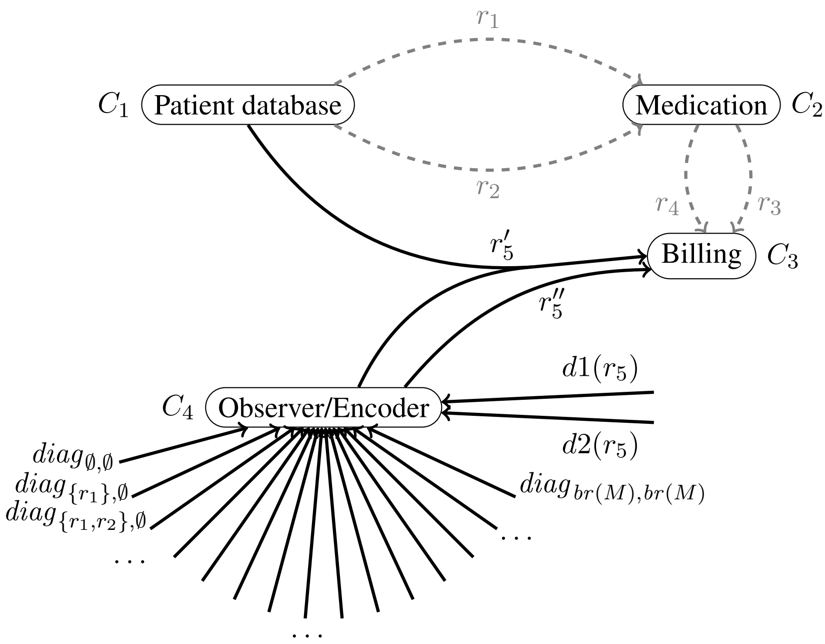


Figure 4.6: Contexts and some bridge rules of the plain encoding $M^{pl\preceq} = (C_1, C_2, C_3, C_4)$ of the hospital MCS wrt. $\preceq$ from Example 4.17. For illustration purposes, only bridge rules stemming from r_5 and some from $\mathcal{K}$ are shown; dashed lines indicate bridge rules $r_1, \dots, r_4$ from M whose corresponding bridge rules in $M^{pl\preceq}$ are not shown.

i.e., some of the bridge rules from $\mathcal{K}$ are indicated. Note that $br_4(M^{mr(\theta,\mathcal{K})})$ contains for every possible diagnosis of M a certain bridge rule.

Regarding the logic and knowledge base employed in $C_4 = (L^{asp}_{\Sigma}, kb_4, br_4)$, we use ASP again to demonstrate a possible realisation, where kb_4 contains the following rules:

$$\begin{array}{rll}
removed_r \leftarrow & not\ not_removed_r. & \forall r \in br(M)\\
\bot \leftarrow & cur_diag_{D_1,D_2}, not\ diag_{D_1,D_2}. & \forall D_1, D_2 \subseteq br(M)\\
cur_diag_{D'_1,D'_2} \leftarrow & cur_diag_{D_1,D_2}. & \forall (D'_1, D'_2) \preceq (D_1, D_2)\\
cur_diag_{D_1,D_2} \leftarrow & removed_{r_1}, \dots, removed_{r_k}, uncond_{r'_1}, \dots, uncond_{r'_m}. & \\
& & \forall D_1, D_2 \subseteq br(M), D_1 = \{r_1, \dots, r_k\}, D_2 = \{r'_1, \dots, r'_m\}
\end{array}$$

Intuitively, the rules of the first line ensure that diagnosis observation is exposed correctly in an accepted belief set of C_4; the following constraints ensure the presence of condition-free bridge rules (i.e., they map each diagnosis candidate to the corresponding bridge rule being condition-free); rules of the third line guarantee that all bridge rules corresponding to more-preferred diagnoses also need to be condition-free, following the ASP semantics these rules do the same as $map^g_{\preceq}(D_1, D_2)$; finally, the rules of the last line recognise one of the exponentially many diagnosis candidates.

The following lemma shows that the set $D^{\pm}_m(M^{pl\preceq}, br_P)$ of minimal diagnoses with protected bridge rules of $M^{pl\preceq}$ corresponds to those diagnoses of M which are at the same time, preferred according to $\preceq$ and $\subseteq$-minimal. These diagnoses not yet correspond to minimal $\preceq$-preferred diagnoses since preference among $\subseteq$-incomparable diagnoses is not captured by $D^{\pm}_m(M^{pl\preceq}, br_P)$.

Lemma 4.9. *Given an MCS M and a preference $\preceq$ on its diagnoses, it holds that*

$$
\begin{aligned}
D^{\pm}_m(M^{pl\preceq}, br_P) = \{(d1(D_1), d2(D_2) \cup K) \mid {} & (D_1, D_2) \in D^{\pm}(M) \wedge \\
& K = map^g_{\preceq}(D_1, D_2) \wedge \forall (D'_1, D'_2) \in D^{\pm}(M) : \\
& \big((D'_1, D'_2) \preceq (D_1, D_2) \wedge (D'_1, D'_2) \subseteq (D_1, D_2)\big) \Rightarrow (D_1, D_2) = (D'_1, D'_2)\}.
\end{aligned}
$$

Proof. By Proposition 4.6 it holds that:

$$
\begin{aligned}
& D^{\pm}_m(M^{mr(\theta,\mathcal{K})}, br_P) = \\
& \{(d1(D_1), d2(D_2) \cup K) \mid (D_1, D_2) \in D^{\pm}(M) \text{ and } \theta(D_1, D_2, K) \text{ holds} \\
& \qquad \text{and there exists no } (D'_1, D'_2) \in D^{\pm}(M) \text{ such that} \\
& \qquad (d1(D'_1), d2(D'_2) \cup K') \subset (d1(D_1), d2(D_2) \cup K) \text{ and} \\
& \qquad \theta(D'_1, D'_2, K') \text{ holds for some } K' \subseteq \mathcal{K}\} \\
= {} & \{(d1(D_1), d2(D_2) \cup K) \mid (D_1, D_2) \in D^{\pm}(M) \wedge \theta(D_1, D_2, K) \wedge \forall (D'_1, D'_2) \in D^{\pm}(M): \\
& \qquad \big(\exists K' : \theta(D'_1, D'_2, K') \wedge \\
& \qquad (d1(D'_1), d2(D'_2) \cup K') \subseteq (d1(D_1), d2(D_2) \cup K)\big) \\
& \qquad \Rightarrow (d1(D'_1), d2(D'_2) \cup K') = (d1(D_1), d2(D_2) \cup K)\}
\end{aligned}
$$

Next we substitute θ by its definition, i.e., $\theta(D_1, D_2, K)$ iff $map^g_{\preceq}(D_1, D_2) = K$.

$$
\begin{aligned}
D^{\pm}_m(M^{mr(\theta,\mathcal{K})}, br_P) = \{(d1(D_1), d2(D_2) \cup K) \mid {} & (D_1, D_2) \in D^{\pm}(M) \\
& \wedge map^g_{\preceq}(D_1, D_2) = K \wedge \forall (D'_1, D'_2) \in D^{\pm}(M): \\
& \big(\exists K' : map^g_{\preceq}(D'_1, D'_2) = K' \wedge \\
& (d1(D'_1), d2(D'_2) \cup K') \subseteq (d1(D_1), d2(D_2) \cup K)\big) \\
& \Rightarrow (d1(D'_1), d2(D'_2) \cup K') = (d1(D_1), d2(D_2) \cup K)\}
\end{aligned}
$$

Since $d1$ and $d2$ both are bijective, $map^g_{\preceq}(D_1, D_2) = K$, and $map^g_{\preceq}(D'_1, D'_2) = K'$, it follows that $(d1(D'_1), d2(D'_2) \cup K') = (d1(D_1), d2(D_2) \cup K)$ holds iff $(D'_1, D'_2) = (D_1, D_2)$. Hence,

$$
\begin{aligned}
D^{\pm}_m(M^{mr(\theta,\mathcal{K})}, br_P) = \{(d1(D_1), d2(D_2) \cup K) \mid {} & (D_1, D_2) \in D^{\pm}(M) \\
& \wedge map^g_{\preceq}(D_1, D_2) = K \wedge \forall (D'_1, D'_2) \in D^{\pm}(M): \\
& \big(\exists K' : (map^g_{\preceq}(D'_1, D'_2) = K' \wedge \\
& (d1(D'_1), d2(D'_2) \cup K') \subseteq (d1(D_1), d2(D_2) \cup K)\big) \\
& \Rightarrow (D_1, D_2) = (D'_1, D'_2)\}
\end{aligned}
$$

Towards the next step, we need to show that the following is true for $(D_1, D_2) \in D^{\pm}(M)$, $(D'_1, D'_2) \in D^{\pm}(M)$, and $map^g_{\preceq}(D_1, D_2) = K$:

$$\begin{aligned}
&\big(map^g_{\preceq}(D'_1, D'_2) = K' \wedge (d1(D'_1), d2(D'_2) \cup K') \subseteq (d1(D_1), d2(D_2) \cup K)\big) \\
&\qquad\qquad\qquad\qquad \Rightarrow (D_1, D_2) = (D'_1, D'_2) \qquad (4.16)\\
&\qquad\qquad\qquad\qquad \text{iff}\\
&\big((D'_1, D'_2) \preceq (D_1, D_2) \wedge (D'_1, D'_2) \subseteq (D_1, D_2)\big)\\
&\qquad\qquad\qquad\qquad \Rightarrow (D_1, D_2) = (D'_1, D'_2) \qquad (4.17)
\end{aligned}$$

Observe that $(d1(D'_1), d2(D'_2) \cup K') \subseteq (d1(D_1), d2(D_2) \cup K)$ holds iff $(D'_1, D'_2) \subseteq (D_1, D_2)$ and $K' \subseteq K$ both hold. Furthermore, by Lemma 4.8 it holds that $K' = map^g_{\preceq}(D'_1, D'_2) \subseteq map^g_{\preceq}(D_1, D_2) = K$ iff $(D'_1, D'_2) \preceq (D_1, D_2)$, given that $(D_1, D_2) \neq (D'_1, D'_2)$. In the case that $(D_1, D_2) = (D'_1, D'_2)$, the implication of (4.16) is trivially true; in this case, (4.17) also holds since its consequent is the same. Therefore, (4.16) holds iff (4.17) holds. After substitution, it therefore holds that:

$$\begin{aligned}
D^{\pm}_m(M^{pl\preceq}, br_P) = \{&(d1(D_1), d2(D_2) \cup K) \mid (D_1, D_2) \in D^{\pm}(M)\\
&\wedge map^g_{\preceq}(D_1, D_2) = K \wedge \forall (D'_1, D'_2) \in D^{\pm}(M) :\\
&\big((D'_1, D'_2) \preceq (D_1, D_2) \wedge (D'_1, D'_2) \subseteq (D_1, D_2)\big) \Rightarrow (D_1, D_2) = (D'_1, D'_2)\}
\end{aligned}$$

□

The following theorem shows the relation between minimal $\preceq$-preferred diagnoses of M wrt. a total preference $\preceq$ and prioritised-minimal diagnoses of $M^{pl\preceq}$. Observe that $map^g_{\preceq}$ is injective since $map^g_{\preceq}(D_1, D_2)$ contains $g(D_1, D_2)$, which by g being a bijection is different for every diagnosis candidate (D_1, D_2). Therefore, $map^g_{\preceq}$ is bijective on its range and it allows to establish a one-to-one relation between minimal $\preceq$-preferred diagnoses of M and prioritised-minimal ones of $M^{pl\preceq}$. Intuitively, this shows that for a total preference order, the set of prioritised-minimal diagnoses of the plain encoding of M wrt. $\preceq$ can be used to select the minimal $\preceq$-preferred diagnoses of M.

Theorem 4.4. *For every MCS M and total preference $\preceq$ on its diagnoses, it holds that*

$$\begin{aligned}
&D^{\pm}(M^{pl\preceq}, br_P, br_H) =\\
&\qquad\qquad \{(d1(D_1), d2(D_2) \cup K) \mid (D_1, D_2) \in D^{\pm}_{m,\preceq}(M), map^g_{\preceq}(D_1, D_2) = K\}.
\end{aligned}$$

Proof. In the following, let θ, $\mathcal{K}$, and $map^g_{\preceq}$ be according to $M^{pl\preceq} = M^{mr(\theta,\mathcal{K})}$.

"⇒": Let $(R_1, R_2) \in D^{\pm}(M^{pl\preceq}, br_P, br_H)$, i.e., $(R_1, R_2) \in D^{\pm}_m(M^{pl\preceq}, br_P)$ and for all $(R'_1, R'_2) \in D^{\pm}_m(M^{pl\preceq}, br_P)$ holds that $(R'_1, R'_2) \subseteq_{br_H} (R_1, R_2) \Rightarrow (R'_1, R'_2) =_{br_H} (R_1, R_2)$. By Lemma 4.9 it holds that $(R_1, R_2) = (d1(D_1), d2(D_2) \cup K)$ where $K = map^g_{\preceq}(D_1, D_2)$ and $(D_1, D_2) \in D^{\pm}(M)$. To show that $(D_1, D_2) \in D^{\pm}_{m,\preceq}(M)$, we have to show that (D_1, D_2) is $\preceq$-preferred and subset minimal among all $\preceq$-preferred diagnoses. Assume that (D_1, D_2) is not $\preceq$-preferred. Then by (4.2) there exists a diagnosis $(D'_1, D'_2) \in D^{\pm}(M)$ such that $(D'_1, D'_2) \preceq$

(D_1, D_2), $(D_1, D_2) \neq (D'_1, D'_2)$, and $(D_1, D_2) \not\preceq (D'_1, D'_2)$ all hold. Let $map^g_{\preceq}(D'_1, D'_2) = K'$ and $map^g_{\preceq}(D_1, D_2) = K$. Since it holds that $(D'_1, D'_2) \neq (D_1, D_2)$ and $(D'_1, D'_2) \preceq (D_1, D_2)$ it follows from Lemma 4.8 that $K' \subseteq K$. From $(D_1, D_2) \not\preceq (D'_1, D'_2)$ it also follows that $K \not\subseteq K'$ holds and thus $K' \subset K$ holds. This means that $(R'_1, R'_2) = (d1(D'_1), d2(D'_2) \cup K') \subset_{br_H} (d1(D_1), d2(D_2) \cup K) = (R_1, R_2)$ holds.

Now suppose $(R'_1, R'_2) \in D^{\pm}_m(M^{pl\preceq}, br_P)$ holds; then $(R_1, R_2) \in D^{\pm}(M^{pl\preceq}, br_P, br_H)$ contradicts that $(R'_1, R'_2) \subset_{br_H} (R_1, R_2)$. On the other hand, $(R'_1, R'_2) \notin D^{\pm}_m(M^{pl\preceq}, br_P)$ implies that some $(R''_1, R''_2) \in D^{\pm}_m(M^{pl\preceq}, br_P)$ exists with $(R''_1, R''_2) \subset (R'_1, R'_2)$, i.e., there exist $D''_1, D''_2 \subseteq br(M)$ such that $(D''_1, D''_2) \preceq (D'_1, D'_2) \preceq (D_1, D_2)$ and $K'' \subseteq K' \subset K$ both hold where $K'' = map^g_{\preceq}(D''_1, D''_2)$, $R''_1 = d1(D''_1)$, and $R''_2 = d2(D''_2) \cup K''$. Since $K'' \subset K$ it therefore holds that $(R''_1, R''_2) \subset_{br_H} (R_1, R_2)$ and together with $(R''_1, R''_2) \in D^{\pm}_m(M^{pl\preceq}, br_P)$ this contradicts that $(R_1, R_2) \in D^{\pm}(M^{pl\preceq}, br_P, br_H)$. Since every case yields a contradiction, it therefore follows that there exists no such (D'_1, D'_2), i.e., (D_1, D_2) indeed is a $\preceq$-preferred diagnosis.

It remains to show that (D_1, D_2) is subset-minimal among all $\preceq$-preferred diagnoses. Towards contradiction, assume there exists $(D'_1, D'_2) \in D^{\pm}_{\preceq}(M)$ such that $(D'_1, D'_2) \subset (D_1, D_2)$. We distinguish on how $\preceq$ relates (D_1, D_2) and (D'_1, D'_2).

- case $(D_1, D_2) \preceq (D'_1, D'_2) \wedge (D'_1, D'_2) \preceq (D_1, D_2)$: since $(R_1, R_2) \in D^{\pm}_m(M^{pl\preceq}, br_P)$, it holds by Lemma 4.9 that $(D'_1, D'_2) \preceq (D_1, D_2) \wedge (D'_1, D'_2) \subseteq (D_1, D_2) \Rightarrow (D'_1, D'_2) = (D_1, D_2)$ which directly contradicts that $(D'_1, D'_2) \subset (D_1, D_2)$.

- case $(D_1, D_2) \preceq (D'_1, D'_2) \wedge (D'_1, D'_2) \not\preceq (D_1, D_2)$: in this case, (D'_1, D'_2) is not $\preceq$-preferred, because $(D_1, D_2) \prec (D'_1, D'_2)$. Hence, it contradicts that $(D'_1, D'_2) \in D^{\pm}_{\preceq}(M)$.

- case $(D_1, D_2) \not\preceq (D'_1, D'_2) \wedge (D'_1, D'_2) \preceq (D_1, D_2)$: this case is analogous to the first one, i.e., $(R_1, R_2) \in D^{\pm}_m(M^{pl\preceq}, br_P)$ contradicts that $(D'_1, D'_2) \preceq (D_1, D_2)$ and $(D'_1, D'_2) \subset (D_1, D_2)$ both hold.

- case $(D_1, D_2) \not\preceq (D'_1, D'_2) \wedge (D'_1, D'_2) \not\preceq (D_1, D_2)$: this case contradicts with $\preceq$ being total.

Consequently, there exists no $(D'_1, D'_2) \in D^{\pm}_{\preceq}(M)$ such that $(D'_1, D'_2) \subset (D_1, D_2)$ and therefore it holds that $(D_1, D_2) \in D^{\pm}_{m,\preceq}(M)$.

"$\Leftarrow$": Let $(D_1, D_2) \in D^{\pm}_{m,\preceq}(M)$. We have to show that

$$(d1(D_1), d2(D_2) \cup K) \in D^{\pm}(M^{pl\preceq}, br_P, br_H)$$

holds with $map^g_{\preceq}(D_1, D_2) = K$. By definition, it holds that

$$D^{\pm}(M^{pl\preceq}, br_P, br_H) = \{D \in D^{\pm}_m(M^{pl\preceq}, br_P) \mid \forall D' \in D^{\pm}_m(M^{pl\preceq}, br_P) : D' \subseteq_{br_H} D \Rightarrow D' =_{br_H} D\}.$$

While by Lemma 4.9 it holds that:

$$
\begin{aligned}
D^{\pm}_m(M^{pl\preceq}, br_P) = \{(d1(D_1), d2(D_2) \cup K) \mid {} & (D_1, D_2) \in D^{\pm}(M) \wedge \\
& K = map^g_{\preceq}(D_1, D_2) \wedge \forall (D'_1, D'_2) \in D^{\pm}(M) : \\
& \big((D'_1, D'_2) \preceq (D_1, D_2) \wedge (D'_1, D'_2) \subseteq (D_1, D_2)\big) \Rightarrow (D_1, D_2) = (D'_1, D'_2)\}.
\end{aligned}
$$

Observe that $br_H = \mathcal{K}$ and $\big(d1(br(M)) \cup d2(br(M))\big) \cap \mathcal{K} = \emptyset$, hence $(d1(D_1), d2(D_2) \cup K) \subseteq_{br_H} (d1(D'_1), d2(D'_2) \cup K')$ holds iff $K \subseteq K'$ holds.

Therefore, it also holds that:

$$
\begin{aligned}
D^{\pm}(M^{pl\preceq}, br_P) = \{(d1(D_1), d2(D_2) \cup K) \in D^{\pm}_m(M^{pl\preceq}, br_P) \mid {} & \\
& \forall (D'_1, D'_2) \in D^{\pm}(M) : \\
& \big[\forall (D''_1, D''_2) \in D^{\pm}(M) : \big((D''_1, D''_2) \preceq (D'_1, D'_2) \\
& \wedge (D''_1, D''_2) \subseteq (D'_1, D'_2)\big) \Rightarrow (D'_1, D'_2) = (D''_1, D''_2)\big] \\
& \Rightarrow \big(map^g_{\preceq}(D'_1, D'_2) \subseteq K \Rightarrow K = map^g_{\preceq}(D'_1, D'_2)\big)\}.
\end{aligned} \tag{4.18}
$$

First, we show that $(d1(D_1), d2(D_2) \cup K) \in D^{\pm}_m(M^{pl\preceq}, br_P)$, which by Lemma 4.9 holds iff the following holds: $(D_1, D_2) \in D^{\pm}(M) \wedge map^g_{\preceq}(D_1, D_2) = K \wedge \forall (D'_1, D'_2) \in D^{\pm}(M) : \big((D'_1, D'_2) \preceq (D_1, D_2) \wedge (D'_1, D'_2) \subseteq (D_1, D_2)\big) \Rightarrow (D_1, D_2) = (D'_1, D'_2)$. Since it holds that $(D_1, D_2) \in D^{\pm}_{m,\preceq}(M)$, it also holds that $(D_1, D_2) \in D^{\pm}(M)$, and $K = map^g_{\preceq}(D_1, D_2)$ by construction.

It remains to show that $\forall (D'_1, D'_2) \in D^{\pm}(M) : \big((D'_1, D'_2) \preceq (D_1, D_2) \wedge (D'_1, D'_2) \subseteq (D_1, D_2)\big) \Rightarrow (D_1, D_2) = (D'_1, D'_2)$. Assume towards contradiction that there exists some $(D'_1, D'_2) \in D^{\pm}(M)$ such that $(D'_1, D'_2) \preceq (D_1, D_2) \wedge (D'_1, D'_2) \subseteq (D_1, D_2)$ and $(D_1, D_2) \neq (D'_1, D'_2)$, i.e., it holds for (D'_1, D'_2) that $(D'_1, D'_2) \subset (D_1, D_2) \wedge (D'_1, D'_2) \preceq (D_1, D_2)$. We distinguish whether $(D_1, D_2) \preceq (D'_1, D'_2)$ also holds: if $(D_1, D_2) \preceq (D'_1, D'_2)$ holds, (D'_1, D'_2) is $\preceq$-preferred since (D_1, D_2) is. Since $(D_1, D_2) \in D^{\pm}_{m,\preceq}(M)$, (D_1, D_2) is subset-minimal among all $\preceq$-preferred diagnoses, which contradicts that $(D'_1, D'_2) \subset (D_1, D_2)$ holds. In the case that $(D_1, D_2) \not\preceq (D'_1, D'_2)$, it holds that $(D_1, D_2) \notin D^{\pm}_{\preceq}(M)$, since it holds that $(D'_1, D'_2) \preceq (D_1, D_2) \wedge (D_1, D_2) \neq (D'_1, D'_2) \wedge (D_1, D_2) \not\preceq (D'_1, D'_2)$. This contradicts that $(D_1, D_2) \in D^{\pm}_{m,\preceq}(M)$. Hence it follows that no such (D'_1, D'_2) exists. Consequently, it holds that $(d1(D_1), d2(D_2) \cup K) \in D^{\pm}_m(M^{pl\preceq}, br_P)$.

According to (4.18), it remains to show that for all $(D'_1, D'_2) \in D^{\pm}(M)$ it holds that

$$
\begin{aligned}
& \big[\forall (D''_1, D''_2) \in D^{\pm}(M) : \big((D''_1, D''_2) \preceq (D'_1, D'_2) \wedge (D''_1, D''_2) \subseteq (D'_1, D'_2)\big) \\
& \qquad \Rightarrow (D'_1, D'_2) = (D''_1, D''_2)\big] \Rightarrow \big(map^g_{\preceq}(D'_1, D'_2) \subseteq K \Rightarrow K = map^g_{\preceq}(D'_1, D'_2)\big).
\end{aligned}
$$

Towards contradiction, assume there exists $(D'_1, D'_2) \in D^{\pm}(M)$ such that $\forall (D''_1, D''_2) \in D^{\pm}(M) : \big((D''_1, D''_2) \preceq (D'_1, D'_2) \wedge (D''_1, D''_2) \subseteq (D'_1, D'_2)\big) \Rightarrow (D'_1, D'_2) = (D''_1, D''_2)$ holds and also $map^g_{\preceq}(D'_1, D'_2) \subsetneq K$ holds. Since $map^g_{\preceq}(D'_1, D'_2) \subsetneq K$, it follows that $(D_1, D_2) \neq (D'_1, D'_2)$ and hence by Lemma 4.8 that $(D'_1, D'_2) \preceq (D_1, D_2)$ and $(D_1, D_2) \not\preceq (D_1', D_2')$ both hold, which implies $(D_1, D_2) \notin D^{\pm}_{m,\preceq}(M)$, in contradiction to the assumption. Therefore, no such (D'_1, D'_2) can exist. This proves that $(d1(D_1), d2(D_2) \cup K) \in D^{\pm}(M^{pl\preceq}, br_P, br_H)$, which completes the proof. □

To select minimal $\preceq$-preferred diagnoses based on an arbitrary preference order, another encoding can be utilised, which we describe next.

Clone Encoding. We now present an approach to meta-reasoning in MCS which allows to select minimal $\preceq$-preferred diagnoses with respect to an arbitrary preference order. This approach, called clone encoding, uses the meta-reasoning encoding $M^{mr(\theta,\mathcal{K})}$ as before, but it is applied not to M directly, but to $M \otimes M$, i.e., to the MCS which consists of two independent copies of M. Any diagnosis of $M \otimes M$ thus contains two possible diagnoses of M and as such the observation/encoding context is able to observe and compare two diagnoses.

The advantage of this approach is that it provably is correct for all preference orders and the resulting MCS is only linearly larger than M. A drawback, however, is that cloning the original MCS may be impractical for some MCS where expensive equipment is needed to implement the contexts of M.

For the purpose of encoding preferences in general, we consider the extension of an MCS by a clone of itself, i.e., for an MCS $M = (C_1, \dots, C_n)$, we define the MCS $2M = (C_1, \dots, C_{2n})$ to be the MCS which contains two independent clones of M; formally, $2M = M \otimes M$ (cf. Section 3.3 for the definition of $\otimes$ on MCS). For easier reference, we write $2.r$ to denote the clone of the bridge rule r, i.e., $2.r = I(r)$ where I is the mapping wrt. $M \otimes M$. Note that $2.br(M)$ is the set of bridge rules of M shifted by n, i.e., $2.br(M)$ is the set of bridge rules of the second clone of M.

The following lemma shows that diagnoses of $2M$ correspond to diagnoses of M in such a way that every diagnosis of $2M$ is composed of two diagnoses of M.

Lemma 4.10. *Let M be an MCS. Then $(D_1, D_2) \in D^{\pm}(2M)$ holds iff there exist $(D'_1, D'_2) \in D^{\pm}(M)$ and $(D''_1, D''_2) \in D^{\pm}(M)$ such that $D_1 = D'_1 \cup 2.D''_1$ and $D_2 = D'_2 \cup 2.D''_2$.*

Proof. Observe that $2M = M \otimes M$ and that $2.R = I(R)$ where I is the mapping wrt. $M \otimes M$. The statement then follows directly from Proposition 3.10. □

The underlying idea of the encoding is that a specific prioritised bridge rule t_{max} indicates whether the diagnosis applied to the second clone is preferred over the diagnosis applied to the first clone. Additionally, the diagnosis of the first clone is exhibited via prioritised bridge rules, while the diagnosis of the second clone is only exhibited via non-prioritised bridge rules.

If the diagnosis applied to the second clone is more preferred than the one applied to the first, then t_{max} needs not become condition-free. Similar to the saturation technique from Answer-Set Programming, if for a given diagnosis of the first clone, there exists some more preferred diagnosis of the second clone, then there exists a diagnosis where t_{max} is not included. So, a diagnosis D such that no more preferred diagnosis D' exists is maximal with respect to the inclusion of t_{max}, because there exists no more preferred diagnosis D' of M that could occur at the second clone. Selecting a diagnosis that modifies a minimal set of prioritised bridge rules and that contains t_{max} thus selects a $\preceq$-preferred diagnosis.

We define t_{max} as follows:

$$t_{max} : \qquad (2n{+}1 : ismax) \leftarrow \bot.$$

To represent the diagnosis of the first clone, we use the following prioritised bridge rules. For a bridge rule $r \in br(M)$ let $in_1(r)$, $\overline{in}_1(r)$, $in_2(r)$, and $\overline{in}_2(r)$ denote the following bridge rules:

$$
\begin{aligned}
in_1(r): &\qquad (2n{+}1 : in_1(r)) \leftarrow \bot.\\
\overline{in}_1(r): &\qquad (2n{+}1 : \overline{in}_1(r)) \leftarrow \bot.\\
in_2(r): &\qquad (2n{+}1 : in_2(r)) \leftarrow \bot.\\
\overline{in}_2(r): &\qquad (2n{+}1 : \overline{in}_2(r)) \leftarrow \bot.
\end{aligned}
$$

Notation. We denote a diagnosis candidate $(D_1, D_2) \in 2^{br(M)} \times 2^{br(M)}$ using these bridge rules by the set $K(D_1, D_2) = \{in_1(r) \mid r \in D_1\} \cup \{\overline{in}_1(r) \mid r \notin D_1\} \cup \{in_2(r) \mid r \in D_2\} \cup \{\overline{in}_2(r) \mid r \notin D_2\}$. The clone encoding then formally is as follows.

Definition 4.23. *Let $M = (C_1, \dots, C_n)$ be an MCS and $\preceq$ a preference order. The* clone encoding *of M wrt. $\preceq$ is the MCS $2M^{mr(\theta,\mathcal{K})}$ where $2M = (C_1, \dots, C_{2n}) = M \otimes M$,*

$$\mathcal{K} = \bigcup_{r \in br(M)} \big\{(2n{+}1 : q) \leftarrow \bot., \mid q \in \{in_1(r), \overline{in}_1(r), in_2(r), \overline{in}_2(r)\}\big\} \cup \{t_{max}\}$$

and for any $R_1, R_2 \subseteq br(2M)$, and $R_3 \subseteq \mathcal{K}$, $\theta(R_1, R_2, R_3)$ holds iff $R_1 = D_1 \cup 2.D'_1$, $R_2 = D_2 \cup 2.D'_2$ and either

- *$(D_1, D_2) = (D'_1, D'_2)$ and $R_3 = K(D_1, D_2) \cup \{t_{max}\}$ or*
- *$(D'_1, D'_2) \preceq (D_1, D_2)$, $(D_1, D_2) \not\preceq (D'_1, D'_2)$, and $R_3 = K(D_1, D_2)$.*

We denote the clone encoding by $M^{\preceq} = 2M^{mr(\theta,\mathcal{K})}$.

Note that the second case above with $(D'_1, D'_2) \preceq (D_1, D_2)$ implies that $(D_1, D_2), (D'_1, D'_2)$ are two diagnoses of M, because the MCS $2M$ only admits a diagnosis if $(D_1, D_2) \in D^{\pm}(M)$ and $(D'_1, D'_2) \in D^{\pm}(M)$ both hold (cf. Lemma 4.10). Also observe that the clone encoding $M^{\preceq} = (2M)^{mr(\theta,\mathcal{K})} = (M \otimes M)^{mr(\theta,\mathcal{K})}$ is linear in the size of M, since for every bridge rule in M there exist $2 \cdot 4 + 4$ bridge rules in $M^{\preceq}$, where the factor 2 is from $M \otimes M$, the factor 4 is from the meta-reasoning encoding itself and the $+4$ is due to $\mathcal{K}$. In summary, it holds that $|br(M^{\preceq})| = 12 \cdot |br(M)| + 1$, where the 1 is t_{max}.

Example 4.18. *Consider the unit-based preference order $\preceq_U$ of Example 4.11 over the MCS M from Example 4.1. The resulting MCS $M^{\preceq} = (C_1, C_2, C_3, C_4, C_5, C_6, C_7)$ is based on two clones of M, where the first comprises the contexts C_1, C_2, C_3 and the second the contexts C_4, C_5, C_6. The context C_7 finally is the observation/encoding context.*

We first recall the bridge rules of $2M = M \otimes M$ using the permutation I corresponding to $M \otimes M$. Accordingly $br(2M)$ is:

$$
\begin{array}{ll}
r_1: & (2:\mathit{hyperglycemia}) \leftarrow (1:\mathit{hyperglycemia}).\\
r_2: & (2:\mathit{allow\_animal\_insulin}) \leftarrow \mathbf{not}\ (1:\mathit{allergic\_animal\_insulin}).\\
r_3: & (3:\mathit{bill\_animal\_insulin}) \leftarrow (2:\mathit{give\_animal\_insulin}).\\
r_4: & (3:\mathit{bill\_human\_insulin}) \leftarrow (2:\mathit{give\_human\_insulin}).\\
r_5: & (3:\mathit{insurance\_B}) \leftarrow (1:\mathit{insurance\_B}).\\
I(r_1): & (5:\mathit{hyperglycemia}) \leftarrow (4:\mathit{hyperglycemia}).\\
I(r_2): & (5:\mathit{allow\_animal\_insulin}) \leftarrow \mathbf{not}\ (4:\mathit{allergic\_animal\_insulin}).\\
I(r_3): & (6:\mathit{bill\_animal\_insulin}) \leftarrow (5:\mathit{give\_animal\_insulin}).\\
I(r_4): & (6:\mathit{bill\_human\_insulin}) \leftarrow (5:\mathit{give\_human\_insulin}).\\
I(r_5): & (6:\mathit{insurance\_B}) \leftarrow (4:\mathit{insurance\_B}).
\end{array}
$$

A graphical rendering of $M^{\preceq}$ is given in Figure 4.7, where for readability only some of the bridge rules of $M^{\preceq}$ are shown. The set of bridge rules of the observation context C_7 is as follows:

$$
\begin{array}{lll}
br_7(M^{\preceq}) = \{ & (7:\mathit{not\_removed}_{r_1}) \leftarrow \top. & (7:\mathit{uncond}_{r_1}) \leftarrow \bot.\\
& (7:\mathit{not\_removed}_{r_2}) \leftarrow \top. & (7:\mathit{uncond}_{r_2}) \leftarrow \bot.\\
& \dots & \\
& (7:\mathit{not\_removed}_{I(r_4)}) \leftarrow \top. & (7:\mathit{uncond}_{I(r_4)}) \leftarrow \bot.\\
& (7:\mathit{not\_removed}_{I(r_5)}) \leftarrow \top. & (7:\mathit{uncond}_{I(r_5)}) \leftarrow \bot.\\
& (7:\mathit{in}_1(r_1)) \leftarrow \bot. & (7:\overline{\mathit{in}}_1(r_1)) \leftarrow \bot.\\
& (7:\mathit{in}_2(r_1)) \leftarrow \bot. & (7:\overline{\mathit{in}}_2(r_1)) \leftarrow \bot.\\
& \dots & \\
& (7:\mathit{in}_1(r_5)) \leftarrow \bot. & (7:\overline{\mathit{in}}_1(r_5)) \leftarrow \bot.\\
& (7:\mathit{in}_2(r_5)) \leftarrow \bot. & (7:\overline{\mathit{in}}_2(r_5)) \leftarrow \bot. \qquad \}
\end{array}
$$

To fully realise the property θ and the preference order $\preceq_U$ based on the units $U_M = \{\mathit{treatment}, \mathit{billing}\}$ with bridge rules associated to units and dependency among units given as in Example 4.11, we may use for the observation context C_7 an ASP program that consists of the following rules:

$$
\begin{array}{lll}
\mathit{removed}_r \leftarrow \mathit{not}\ \mathit{not\_removed}_r. & \forall r \in br(M \otimes M) & (4.19)\\
\bot \leftarrow \mathit{removed}_r, \mathit{not}\ \mathit{in}_1(r). & \forall r \in \{r_1, \dots, r_5\} & (4.20)\\
\bot \leftarrow \mathit{not}\ \mathit{removed}_r, \mathit{in}_1(r). & \forall r \in \{r_1, \dots, r_5\} & \\
\bot \leftarrow \mathit{not}\ \mathit{removed}_r, \mathit{not}\ \overline{\mathit{in}}_1(r). & \forall r \in \{r_1, \dots, r_5\} & \\
\bot \leftarrow \mathit{removed}_r, \overline{\mathit{in}}_1(r). & \forall r \in \{r_1, \dots, r_5\} & \\
\bot \leftarrow \mathit{uncond}_r, \mathit{not}\ \mathit{in}_2(r). & \forall r \in \{r_1, \dots, r_5\} &
\end{array}
$$

$$\bot \leftarrow uncond_r, \overline{in}_2(r). \qquad \forall r \in \{r_1, \dots, r_5\}$$
$$\bot \leftarrow not\ uncond_r, not\ \overline{in}_2(r). \qquad \forall r \in \{r_1, \dots, r_5\}$$
$$\bot \leftarrow uncond_r, \overline{in}_2(r). \qquad \forall r \in \{r_1, \dots, r_5\} \tag{4.21}$$

$$mod(clone1, billing) \leftarrow removed_r. \qquad \forall r \in \{r_3, \dots, r_5\} \tag{4.22}$$
$$mod(clone1, billing) \leftarrow uncond_r. \qquad \forall r \in \{r_3, \dots, r_5\}$$
$$mod(clone2, billing) \leftarrow removed_r. \qquad \forall r \in \{I(r_3), \dots, I(r_5)\}$$
$$mod(clone2, billing) \leftarrow uncond_r. \qquad \forall r \in \{I(r_3), \dots, I(r_5)\}$$
$$mod(clone1, treatment) \leftarrow removed_r. \qquad \forall r \in \{r_1, r_2\}$$
$$mod(clone1, treatment) \leftarrow uncond_r. \qquad \forall r \in \{r_1, r_2\}$$
$$mod(clone2, treatment) \leftarrow removed_r. \qquad \forall r \in \{I(r_1), I(r_2)\}$$
$$mod(clone2, treatment) \leftarrow uncond_r. \qquad \forall r \in \{I(r_1), I(r_2)\} \tag{4.23}$$
$$mod(clone1, billing) \leftarrow mod(clone1, treatment). \tag{4.24}$$
$$mod(clone2, billing) \leftarrow mod(clone2, treatment). \tag{4.25}$$

$$clones\_different \leftarrow removed_r, not\ removed_{r'}. \qquad \forall r \in br(M), \forall r' \in I(br(M)) \tag{4.26}$$
$$clones\_different \leftarrow not\ removed_r, removed_{r'}. \qquad \forall r \in br(M), \forall r' \in I(br(M))$$
$$clones\_different \leftarrow uncond_r, not\ uncond_{r'}. \qquad \forall r \in br(M), \forall r' \in I(br(M))$$
$$clones\_different \leftarrow not\ uncond_r, uncond_{r'}. \qquad \forall r \in br(M), \forall r' \in I(br(M)) \tag{4.27}$$

$$clone1\_modifies\_more \leftarrow mod(clone1, U), not\ mod(clone2, U). \tag{4.28}$$
$$clone2\_modifies\_more \leftarrow mod(clone2, U), not\ mod(clone1, U).$$
$$clone1\_less\_preferred \leftarrow clone1\_modifies\_more, not\ clone2\_modifies\_more. \tag{4.29}$$
$$\bot \leftarrow not\ ismax, clone1\_less\_preferred, clones\_different. \tag{4.30}$$
$$\bot \leftarrow not\ clone1\_less\_preferred, clones\_different. \tag{4.31}$$

The intuition of the above rules is as follows: rules of form (4.19) *expose the diagnoses of both clones; the constraints of form* (4.20)–(4.21) *ensure that the diagnosis of the first clone is exhibited via prioritised bridge rules; rules of form* (4.22)–(4.23) *deduce which units of bridge rules have been modified in the first and second clone; rules* (4.24) *and* (4.25) *take care of the dependency between the units treatment and billing; rules of form* (4.26)–(4.27) *infer whether the diagnosis of the first clone is different from the diagnosis of the second clone; rules* (4.28)–(4.29) *infer whether the modified units of the first clone is a superset of the modified units of the second clone, which means the diagnosis of the second clone is more preferred than the one of the first clone. Finally, the constraint* (4.30) *ensures that* t_{max} *is made condition-free if the diagnosis of the second clone is more preferred than the diagnosis of the first clone, and the constraint* (4.31) *ensures that only comparable diagnoses (or if both diagnoses are equal) yield a diagnosis of the MCS* $M^{\preceq}$.

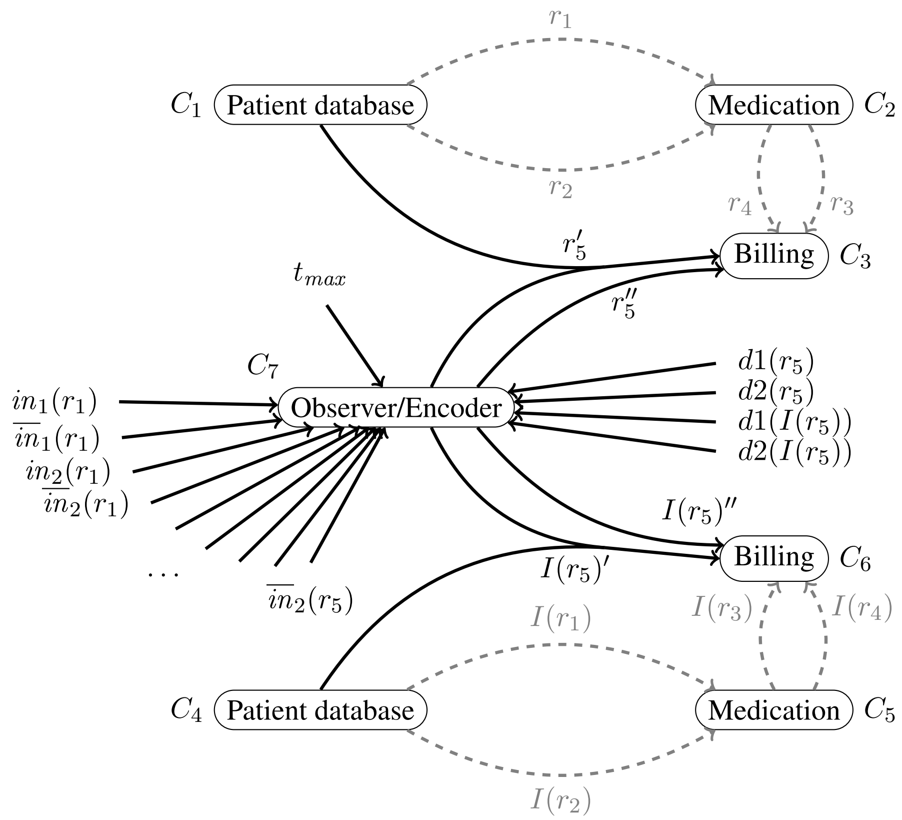


Figure 4.7: The MCS $M^{\preceq} = (C_1, C_2, \dots, C_7)$ of Example 4.18. Some of the bridge rules of the observation context C_7 are shown and the bridge rules stemming from r_5 of the MCS M. Dashed and gray lines indicate the other bridge rules of $M \otimes M$ whose resulting bridge rules in $M^{\preceq}$ are omitted for presentation purpose. The prioritised bridge rules of $M^{\preceq}$ are t_{max} and all bridge rules of form $in_i(r_j)$ and $\overline{in}_i(r_j)$.

For selecting minimal $\preceq$-preferred diagnoses based on an arbitrary preference order, we strengthen Definition 4.20 in two steps: first, if two diagnoses are equal considering their prioritised bridge rules, then subset-minimality on the remaining bridge rules is taken into account. Second, since we only want to select diagnoses where no more preferred ones exist, we consider only prioritised-minimal diagnoses that contain the bridge rule t_{max}.

For the first step, let M be an MCS with bridge rules $br(M)$, protected rules br_P, and prioritised rules $br_H \subseteq br(M)$. The set of subset-minimal prioritised-minimal diagnoses then is:

$$
\begin{aligned}
D^{\pm}_m(M, br_P, br_H) = \Big\{ & D \in D^{\pm}_m(M, br_P) \mid \\
& \big[\forall D' \in D^{\pm}_m(M, br_P) : D' \subseteq_{br_H} D \Rightarrow D' =_{br_H} D\big] \wedge \\
& \Big[\forall D' \in D^{\pm}_m(M, br_P) : \\
& \quad \big[\big(\forall D'' \in D^{\pm}_m(M, br_P) : D'' \subseteq_{br_H} D' \Rightarrow D'' =_{br_H} D'\big)\big]
\end{aligned}
\tag{4.32}
$$

$$\Rightarrow \big[D' \subseteq_{br(M)\setminus br_H} D \Rightarrow D =_{br(M)\setminus br_H} D'\big]\Big]\Big\}$$

The first condition ensures that a diagnosis D is prioritised-minimal (lines one and two) and for all other diagnoses D' that are prioritised-minimal (lines three and four) it holds that D is a subset of D' (line five).

For the second step, we just add to $D^{\pm}_{m}(M, br_P, br_H)$ the condition that D and D' make t_{max} condition-free. Formally:

Definition 4.24. *Given an MCS M with protected bridge rules br_P and prioritised bridge rules br_H, the set of* subset-minimal prioritised-minimal (mpm) diagnoses wrt. t_{max} *is*

$$\begin{aligned}
D^{\pm}_{m,t_{max}}(M, br_P, br_H) = \Big\{ & D \in D^{\pm}_{m}(M, br_P) \mid t_{max} \in D \wedge \\
& \big[\forall D' \in D^{\pm}_{m}(M, br_P) : D' \subseteq_{br_H} D \Rightarrow D' =_{br_H} D\big] \wedge \\
& \Big[\forall D' \in D^{\pm}_{m}(M, br_P) : \\
& \quad \big[\big(\forall D'' \in D^{\pm}_{m}(M, br_P) : D'' \subseteq_{br_H} D' \Rightarrow D'' =_{br_H} D'\big) \wedge t_{max} \in D'\big] \\
& \Rightarrow \big[D' \subseteq_{br(M)\setminus br_H} D \Rightarrow D =_{br(M)\setminus br_H} D'\big]\Big]\Big\}
\end{aligned}$$

where $t_{max} \in D$ stands for $D = (D_1, D_2) \wedge t_{max} \in D_2$.

Intuitively, D is an mpm-diagnosis, if it respects protected bridge rules and contains t_{max} (first line), if it is preferred, i.e., it is minimal wrt. prioritised bridge rules br_H among all other diagnoses of the MCS M (second line), and if for all all other preferred diagnoses that contain t_{max} (third and fourth line) it holds that D is subset-minimal wrt. regular bridge rules (fifth line).

As we show in the following section on complexity, this notion is computationally harder than the notion of prioritised-minimal diagnosis. Nevertheless, the problem itself (i.e., the problem of selecting a minimal $\preceq$-preferred diagnosis) is shown to be as hard as this notion, which means the notion is worst-case optimal.

Note that $D, D' \in D^{\pm}(M, br_P)$ implies that $D \subseteq_{br(M)\setminus br_H} D'$ holds iff $D \subseteq_{br(M)\setminus br_H \setminus br_P} D'$ holds, because $D = (D_1, D_2) \in D^{\pm}(M, br_P)$ implies that $D_1 \cap br_P = \emptyset = D_2 \cap br_P$. The same also holds for $=_{br(M)\setminus br_H}$ and $=_{br(M)\setminus br_H \setminus br_P}$.

Towards proving that $D^{\pm}_{m,t_{max}}$ applied on $M^{\preceq}$ allows to select $\subseteq$-minimal, preferred diagnoses of M according to $\preceq$, we use the following lemmas about the set $K(D_1, D_2)$. Recall that $K(D_1, D_2)$ is the set of prioritised bridge rules of $M^{\preceq}$ that represent the diagnosis candidate (D_1, D_2) of M, i.e., $K(D_1, D_2)$ is as follows:

$$\begin{aligned}
K(D_1, D_2) = & \{in_1(r) \mid r \in D_1\} \cup \{\overline{in}_1(r) \mid r \notin D_1\} \cup \\
& \{in_2(r) \mid r \in D_2\} \cup \{\overline{in}_2(r) \mid r \notin D_2\}
\end{aligned}$$

The next lemma shows that the set $K(D_1, D_2)$ is unique for every $D_1, D_2 \subseteq br(M)$.

Lemma 4.11. *Let $M^{\preceq}$ be a clone encoding, $D_1, D_2 \subseteq br(M)$, and $R = K(D_1, D_2)$. Then, there exists no $D'_1, D'_2 \subseteq br(M)$ with $(D_1, D_2) \neq (D'_1, D'_2)$ such that $R = K(D'_1, D'_2)$.*

Proof. Towards contradiction, let $(D_1, D_2) \neq (D'_1, D'_2)$ be such that $K(D_1, D_2) = K(D'_1, D'_2)$. By $(D_1, D_2) \neq (D'_1, D'_2)$ follows that either $D_1 \neq D'_1$ or $D_2 \neq D'_2$. Let $D_1 \neq D'_1$ and observe that $K(D_1, D_2) \cap \{in_1(r) \mid r \in br(M)\} = \{in_1(r) \mid r \in D_1\} \neq \{in_1(r) \mid r \in D'_1\} = K(D'_1, D'_2) \cap \{in_1(r) \mid r \in br(M)\}$. Consequently $K(D_1, D_2) \neq K(D'_1, D'_2)$ which contradicts the assumption. The case $D_2 \neq D'_2$ is similar. It therefore follows that for $R = K(D_1, D_2)$ no $D'_1, D'_2 \subseteq br(M)$ with $(D_1, D_2) \neq (D'_1, D'_2)$ exists such that $R = K(D'_1, D'_2)$. □

The next lemma shows that two sets $K(D_1, D_2)$ and $K(D'_1, D'_2)$ are incomparable iff (D_1, D_2) is different from (D'_1, D'_2).

Lemma 4.12. *Given $M^{mr(\theta,\mathcal{K})}$ and some $D_1, D_2, D'_1, D'_2 \subseteq br(M)$, let $R = K(D_1, D_2)$ and let $R' = K(D'_1, D'_2)$; then $R \subseteq R'$ or $R' \subseteq R$ holds iff $(D_1, D_2) = (D'_1, D'_2)$.*

Proof. Let M be an MCS, $D_1, D_2, D'_1, D'_2 \subseteq br(M)$, $R = K(D_1, D_2)$, and $R' = K(D'_1, D'_2)$. Observe that by definition of K it holds that $|R| = |R'|$. Hence, $R \subseteq R'$ or $R' \subseteq R$ only holds iff $R = R'$. By Lemma 4.11 it holds that K is injective, i.e., $R = R'$ iff $(D_1, D_2) = (D'_1, D'_2)$. Consequently, $R \subseteq R'$ or $R' \subseteq R$ holds iff $(D_1, D_2) = (D'_1, D'_2)$.

□

In the following, we write $t(D_1, D_2)$ as a shorthand for the corresponding diagnosis in the MCS $M^{\preceq}$, so $t(D_1, D_2) = (d1(D_1 \cup 2.D_1), d2(D_2 \cup 2.D_2) \cup K(D_1, D_2) \cup \{t_{max}\})$; for $D = (D_1, D_2)$ we also write $t(D)$ to denote $t(D_1, D_2)$. The following lemma shows the relationship between $\preceq$-preferred diagnoses of M and prioritised-minimal ones of $M^{\preceq}$.

Lemma 4.13. *Given an MCS M and a preference order $\preceq$, $D \in 2^{br(M)} \times 2^{br(M)}$ is $\preceq$-preferred iff both (1) $t(D) \in D^{\pm}_m(M^{\preceq}, br_P)$ and (2) for every $D' \in D^{\pm}_m(M^{\preceq}, br_P) : D' \subseteq_{br_H} t(D) \Rightarrow D' =_{br_H} t(D)$ hold.*

Proof. "$\Rightarrow$": Let D be $\preceq$-preferred, then $D \in D^{\pm}(M)$ holds. We first show that $t(D) \in D^{\pm}_m(M^{\preceq}, br_P)$ holds: by Proposition 4.6 and the definition of $M^{\preceq} = M^{mr(\theta,\mathcal{K})}$ it holds that $(d1(D_1 \cup 2.D_1), d2(D_2 \cup 2.D_2) \cup K(D_1, D_2) \cup \{t_{max}\}) \in D^{\pm}(M^{\preceq}, br_P)$ iff

1. $(D_1 \cup 2.D_1, D_2 \cup 2.D_2) \in D^{\pm}(2M)$ holds,
2. $\theta(D_1 \cup 2.D_1, D_2 \cup 2.D_2, K(D_1, D_2) \cup \{t_{max}\})$ holds, and
3. there exists no $(D'_1 \cup 2.D''_1, D'_2 \cup 2.D''_2) \in D^{\pm}(2M)$ such that (i) $(d1(D'_1 \cup 2.D''_1), d2(D'_2 \cup 2.D''_2) \cup K') \subset (d1(D_1 \cup 2.D_1), d2(D_2 \cup 2.D_2) \cup K(D_1, D_2) \cup \{t_{max}\})$ and (ii) $\theta(D'_1 \cup 2.D''_1, D'_2 \cup 2.D''_2, K')$ holds for some $K' \subseteq \mathcal{K}$.

We show that each of those statements holds:

1. Since $D \in D^{\pm}(M)$ holds, it follows from Lemma 4.10 that $(D_1 \cup 2.D_1, D_2 \cup 2.D_2) \in D^{\pm}(2M)$ holds.

2. Recall that $\theta(R_1, R_2, R_3)$ for $M^{\preceq} = (2M)^{mr(\theta,\mathcal{K})}$ is defined such that it holds if $R_1 = D_1 \cup 2.D_1$, $R_2 = D_2 \cup 2.D_2$, and $R_3 = K(D_1, D_2) \cup \{t_{max}\}$, hence $\theta(D_1 \cup 2.D_1, D_2 \cup 2.D_2, K(D_1, D_2) \cup \{t_{max}\})$ holds.

3. Towards contradiction, assume that there exists $(D'_1 \cup 2.D''_1, D'_2 \cup 2.D''_2) \in D^{\pm}(2M)$ and $K' \subseteq \mathcal{K}$ such that it holds that $(d1(D'_1 \cup 2.D''_1), d2(D'_2 \cup 2.D''_2) \cup K') \subset (d1(D_1 \cup 2.D_1), d2(D_2 \cup 2.D_2) \cup K(D_1, D_2) \cup \{t_{max}\})$ and $\theta(D'_1 \cup 2.D''_1, D'_2 \cup 2.D''_2, K')$ holds. Note that from this it follows that $K' \subseteq K(D_1, D_2) \cup \{t_{max}\}$ and from the definition of θ that $K' \subseteq K(D'_1, D'_2) \cup \{t_{max}\}$. Hence by Lemma 4.12, it follows that $(D'_1, D'_2) = (D_1, D_2)$. If $(D''_1, D''_2) = (D_1, D_2)$ then it holds by definition of θ that $t_{max} \in K'$, i.e., $(d1(D_1 \cup 2.D_1), d2(D_2 \cup 2.D_2) \cup K(D_1, D_2) \cup \{t_{max}\}) = (d1(D'_1 \cup 2.D''_1), d2(D'_2 \cup 2.D''_2) \cup K')$ which contradicts that the latter is a proper subset of the former. If $(D''_1, D''_2) \neq (D_1, D_2)$ holds, then by definition of θ it follows that $(D''_1, D''_2) \preceq (D_1, D_2) = (D'_1, D'_2)$ and $(D_1, D_2) = (D'_1, D'_2) \not\preceq (D''_1, D''_2)$ both hold, which contradicts that (D_1, D_2) is $\preceq$-preferred. It therefore follows that no such $(D'_1 \cup 2.D''_1, D'_2 \cup 2.D''_2) \in D^{\pm}(2M)$ exists.

Since all three statements hold, it follows that $t(D) \in D^{\pm}_m(M^{\preceq}, br_P)$ holds.

It remains to show that $\forall T \in D^{\pm}_m(M^{\preceq}, br_P) : T \subseteq_{br_H} t(D) \Rightarrow T =_{br_H} t(D)$ holds. Assume that $T \in D^{\pm}_m(M^{\preceq}, br_P)$ is such that $T \subseteq_{br_H} t(D)$ holds. Then by definition of θ it holds that $T = (d1(T_1 \cup 2.T'_1), d2(T_2 \cup 2.T'_2) \cup K(T_1, T_2) \cup T_m)$ for some $T_1, T_2, T'_1, T'_2 \subseteq br(M)$ and $T_m \subseteq \{t_{max}\}$. Since $K(T_1, T_2) \subseteq \mathcal{K}$, it holds by $T \subseteq_{br_H} t(D)$ that $K(T_1, T_2) \subseteq K(D_1, D_2)$, hence by Lemma 4.12 it follows that $(T_1, T_2) = (D_1, D_2)$. Since (D_1, D_2) is $\preceq$-preferred, i.e., there exists no $(D'_1, D'_2) \in D^{\pm}(M)$ such that $(D'_1, D'_2) \preceq (D_1, D_2)$ and $(D_1, D_2) \not\preceq (D'_1, D'_2)$ both hold, it follows from the definition of θ that $(T'_1, T'_2) = (D_1, D_2)$ and consequently it holds that $T_m = \{t_{max}\}$. Altogether this means that $T = t(D)$ and thus it holds that $T =_{br_H} t(D)$. It therefore holds that $\forall T \in D^{\pm}_m(M^{\preceq}, br_P) : T \subseteq_{br_H} t(D) \Rightarrow T =_{br_H} t(D)$.

"$\Leftarrow$": Suppose $t(D_1, D_2) \in D^{\pm}_m(M^{\preceq}, br_P)$ and $\forall T \in D^{\pm}_m(M^{\preceq}, br_P) : T \subseteq_{br_H} t(D) \Rightarrow T =_{br_H} t(D)$ with $D = (D_1, D_2)$ hold. Since $t(D_1, D_2) \in D^{\pm}_m(M^{\preceq}, br_P)$ holds, it follows from Proposition 4.6 that $(d1(D_1 \cup 2.D_1), d2(D_2 \cup 2.D_2)) \in D^{\pm}(2M)$, hence by Lemma 4.10 it holds that $(D_1, D_2) \in D^{\pm}(M)$.

To show that D is $\preceq$-preferred, consider the set F of diagnoses that are more preferred than D, i.e., $F = \{D'' \in D^{\pm}(M) \mid D'' \preceq D, D \not\preceq D''\}$. Towards contradiction, assume that F is non-empty, hence there exists some subset-minimal $D' \in F$, i.e., $D' \in F$ and for all $D'' \in F$ holds $D'' \not\subseteq D'$. Next we consider $(T'_1, T'_2) = (d1(D_1 \cup D'_1), d2(D_2 \cup D'_2) \cup K(D_1, D_2))$ and observe that $\theta(D_1 \cup D'_1, D_2 \cup D'_2, K(D_1, D_2))$ holds, because $D' \preceq D$ and $D \not\preceq D'$ both hold.

Since $(D_1, D_2) \in D^{\pm}(M)$ and $(D'_1, D'_2) \in D^{\pm}(M)$ it holds that $(D_1 \cup 2.D'_1, D_2 \cup 2.D'_2) \in D^{\pm}(2M)$. Observe that there exists no other $D'' \subset D'$ with $D \preceq D''$, $D'' \not\preceq D$, and $D'' \in D^{\pm}(M)$. Therefore, there exists no $(D''_1, D''_2) \in D^{\pm}(M)$ such that $(d1(D_1 \cup 2.D''_1), d2(D_2 \cup 2.D''_2) \cup K(D_1, D_2)) \subset (T'_1, T'_2)$ and $\theta(D_1 \cup 2.D''_1, D_2 \cup 2.D''_2, K(D_1, D_2))$ both hold. Thus Proposition 4.6 applies and it follows that $(T'_1, T'_2) \in D^{\pm}_m(M^{\preceq}, br_P)$. Observe that $(T'_1, T'_2) \subseteq_{br_H} t(D)$ since $T'_2 \cap br_H = K(D_1, D_2) \cup \{t_{max}\}$ and for $t(D) = (T_1, T_2)$ holds $T_2 \cap br_H = K(D_1, D_2)$. This directly contradicts that $\forall T \in D^{\pm}_m(M^{\preceq}, br_P) : T \subseteq_{br_H} t(D) \Rightarrow T =_{br_H} t(D)$ holds. Thus the set F cannot be non-empty, i.e., there exists no $D' \in D^{\pm}(M)$ such that $D' \preceq D$ and $D \not\preceq D'$ both hold. Therefore, D is $\preceq$-preferred. □

We immediately obtain from the previous lemma that $D^{\pm}(M^{\preceq}, br_P, br_H)$ suffices to obtain those diagnoses of M that are $\preceq$-preferred according to $\preceq$.

Theorem 4.5. *Let M be an MCS and $\preceq$ be a preference on diagnoses of M. Then $D \in D^{\pm}(M)$ is $\preceq$-preferred iff $t(D) \in D^{\pm}(M^{\preceq}, br_P, br_H)$ holds.*

Proof. Recall that $D^{\pm}(M, br_P, br_H) = \{D \in D^{\pm}_m(M, br_P) \mid \forall D' \in D^{\pm}_m(M, br_P) : D' \subseteq_{br_H} D \Rightarrow D' =_{br_H} D\}$. Hence, $t(D) \in D^{\pm}(M^{\preceq}, br_P, br_H)$ holds iff $t(D) \in D^{\pm}_m(M^{\preceq}, br_P)$ holds and for every $D' \in D^{\pm}_m(M^{\preceq}, br_P)$ it holds that $D' \subseteq_{br_H} t(D) \Rightarrow D' =_{br_H} t(D)\}$. By Lemma 4.13 this condition holds iff D is $\preceq$-preferred. In summary, D is $\preceq$-preferred iff $t(D) \in D^{\pm}(M^{\preceq}, br_P, br_H)$ holds. □

Note that $t(D) \in D^{\pm}(M^{\preceq}, br_P, br_H)$ implies that $t_{max} \in t(D)$; but there also are diagnoses $T \in D^{\pm}(M^{\preceq}, br_P, br_H)$ such that $t_{max} \notin T$. Nevertheless, it follows directly from the definition of $M^{\preceq}$ that for any $T \in D^{\pm}(M^{\preceq}, br_P, br_H)$ with $t_{max} \in T$ there exist $D_1, D_2 \subseteq br(M)$ such that $T = t(D_1, D_2)$. So, diagnoses of $D^{\pm}(M^{\preceq}, br_P, br_H)$ that contain t_{max} correspond one-to-one to $\preceq$-preferred diagnoses of M.

The next theorem shows that the clone encoding $M^{\preceq}$ and the notion of mpm-diagnosis $D^{\pm}_{m,t_{max}}$ allows to select all minimal $\preceq$-preferred diagnoses of M according to $\preceq$. This theorem therefore establishes that the clone encoding is sound and complete.

Theorem 4.6. *Let M be an MCS and $\preceq$ be a preference order on diagnoses of M. Then $(D_1, D_2) \in D^{\pm}_{m,\preceq}(M)$ holds iff $t(D_1, D_2) \in D^{\pm}_{m,t_{max}}(M^{\preceq}, br_P, br_H)$ holds.*

Proof. "⇒": Let $D = (D_1, D_2) \in D^{\pm}_{m,\preceq}(M)$ hold. Then $D \in D^{\pm}_{\preceq}(M)$ holds, i.e., D is $\preceq$-preferred and $D \in D^{\pm}(M)$ holds. From Lemma 4.13 we then conclude that $t(D) \in D^{\pm}_m(M^{\preceq}, br_P)$ and that the following holds: $\forall T \in D^{\pm}_m(M^{\preceq}, br_P) : T \subseteq_{br_H} t(D) \Rightarrow T =_{br_H} t(D)$. By construction of $t(D)$ it furthermore holds that $t_{max} \in t(D)$. Hence it remains to show that $\forall T' \in D^{\pm}_m(M^{\preceq}, br_P) : \left[\left(\forall T'' \in D^{\pm}_m(M^{\preceq}, br_P) : T'' \subseteq_{br_H} T' \Rightarrow T'' =_{br_H} T'\right) \wedge t_{max} \in T'\right] \Rightarrow \left[T' \subseteq_{br(M^{\preceq})\setminus br_H} t(D) \Rightarrow t(D) =_{br(M^{\preceq})\setminus br_H} T'\right]$.

Towards contradiction, assume that $T' \in D^{\pm}_m(M^{\preceq}, br_P)$ exists with $\left(\forall T'' \in D^{\pm}_m(M^{\preceq}, br_P) : T'' \subseteq_{br_H} T' \Rightarrow T'' =_{br_H} T'\right) \wedge t_{max} \in T'$ and $T' \subset_{br(M^{\preceq})\setminus br_H} t(D)$. Note that the definition of θ and $t_{max} \in T'$ together imply that there exists some $D' = (D'_1, D'_2)$ with $D'_1, D'_2 \subseteq br(M)$ such that $T' = t(D')$ holds. Further note that $T' = t(D')$ satisfies all conditions of Lemma 4.13, thus it holds that $D' \in D^{\pm}(M)$ and that D' is $\preceq$-preferred.

From $T' = t(D') \subset_{br(M^{\preceq})\setminus br_H} t(D)$ it follows that $(d1(D'_1 \cup 2.D'_1), d2(D'_2 \cup 2.D'_2)) \subset (d1(D_1 \cup 2.D_1), d2(D_2 \cup 2.D_2))$ and since $d1$, $d2$, and $2.$ are bijective, it holds that $(D'_1, D'_2) \subset (D_1, D_2)$. Since D' is $\preceq$-preferred, this contradicts that D is subset-minimal among all $\preceq$-preferred diagnoses, i.e., it contradicts that $D \in D^{\pm}_{m,\preceq}(M)$. Therefore no such T' can exist and it holds that $t(D) \in D^{\pm}_{m,t_{max}}(M^{\preceq}, br_P, br_H)$.

"⇐": Let $t(D_1, D_2) \in D^{\pm}_{m,t_{max}}(M^{\preceq}, br_P, br_H)$ hold. Since $t(D_1, D_2) \in D^{\pm}_m(M^{\preceq}, br_P)$ and $t_{max} \in t(D_1, D_2)$ hold, it follows from Lemma 4.13 that $D = (D_1, D_2) \in D^{\pm}(M)$ and that D is $\preceq$-preferred. It remains to show that D is subset-minimal among diagnoses in $D^{\pm}_{\preceq}(M)$.

Towards contradiction, assume that there exists $D' \in D^{\pm}_{\preceq}(M)$ with $D' \subset D$. Since D' is $\preceq$-preferred and $D' \in D^{\pm}(M)$ holds, it follows from Lemma 4.13 that $t(D') \in D^{\pm}_m(M^{\preceq}, br_P)$ and

$\forall T \in D^{\pm}_{m}(M^{\preceq}, br_P) : T \subseteq_{br_H} t(D') \Rightarrow T =_{br_H} t(D')$ holds. Let $T' = t(D')$. Then it holds for T' that $\big(\forall T'' \in D^{\pm}_{m}(M, br_P) : T'' \subseteq_{br_H} T' \Rightarrow T'' =_{br_H} T'\big) \wedge t_{max} \in T'$. Let $T = t(D)$. Because $d1, d2$, and 2. are bijective and $D' \subset D$, it follows that $\big[T' \subseteq_{(br(M)\setminus br_H)} T \Rightarrow T =_{br(M)\setminus br_H} T'\big]$ does not hold. This contradicts that $t(D_1, D_2) \in D^{\pm}_{m,t_{max}}(M^{\preceq}, br_P, br_H)$ holds. Therefore no such D' exists and it holds that D is subset-minimal among $D^{\pm}_{\preceq}(M)$, i.e., $D \in D^{\pm}_{m,\preceq}(M)$ holds. □

Recall that, given a CP-net N compatible with an MCS M, the minimal $\preceq$-preferred diagnoses according to $\preceq^N$ and the optimal ones according to N coincide, i.e., $D^{\pm}_{opt}(M, N) = D^{\pm}_{m,\preceq^N}(M)$ (cf. Proposition 4.2). One thus can realise the selection of optimal diagnoses according to a CP-net using the clone encoding $M^{\preceq^N}$ and the methods provided in this section. Also note that the size of $M^{\preceq^N}$ is only linearly larger than M.

Since the approaches only specify some of the behaviour of the observation context, the concrete choice of logic to use and realise the observation is open to the choice of the user. This is especially useful for preference formalisms like CP-nets where algorithms may be chosen according to the computational complexity of the employed CP-net.

De-centralized meta-reasoning: All approaches at meta-reasoning use one central observation context which knows all bridge rules and knows for each bridge rule whether and how it is modified. Although this observation context does not know the actual status of the information exchange, it is still violating information hiding to some extent. In the previous chapter the decomposition of a context is investigated and some criteria are given which preserve a one-to-one correspondence between the diagnoses of the original MCS and the MCS where one context is decomposed into two contexts (cf. Proposition 3.11). These results suggest that the central observation context of the filter encoding M^f and the clone encoding $M^{\preceq}$ may be decomposed without interfering with the correctness of the diagnosis observations.

Indeed, if a filter f for a given MCS M is such that there exists $A, B \subset br(M)$ with $A \cup B = br(M)$ and $A \cap B = \emptyset$ and for all $D_1, D_2 \subseteq br(M)$ it holds that $f(D_1, D_2) = 1$ iff $f(D_1 \cap A, D_2 \cap A) = 1$ and $f(D_1 \cap B, D_2 \cap B) = 1$, then the observation context of $M^f = (C_1, \ldots, C_n, C_{n+1})$ is decomposable. Intuitively, f is such that the modifications of bridge rules in A can be checked independently from the modifications of bridge rules in B and vice versa. The decomposition of C_{n+1} then is based on the following sets: $br^A_{n+1} = \{d1(r), d2(r) \mid r \in A\}$, $br^B_{n+1} = \{d1(r), d2(r) \mid r \in B\}$, $OUT^A_{n+1} = \{removed_r, uncond_r \mid r \in A\}$, and $OUT^B_{n+1} = \{removed_r, uncond_r \mid r \in B\}$.

Since the meta-reasoning encoding does not yield a specific knowledge base for C_{n+1}, we cannot state the decomposed knowledge base in general, but for any reasonable logical formalism that realises the check whether $f(D_1, D_2) = 1$ it is possible to realise the checks whether $f(D_1 \cap A, D_2 \cap A) = 1$ and whether $f(D_1 \cap B, D_2 \cap B) = 1$ by two (independent) knowledge bases. These knowledge bases then complete the decomposition of the observation context. If the filter f permits it, this decomposition can be repeated several times, where each time one context is decomposed into two independent ones, until the observation of diagnoses is fully decentralised.

Example 4.19. *Consider the MCS $M^f = (C_1, C_2, C_3)$ of Example 4.16 realising the filter f on the MCS M whose bridge rules are $br(M) = \{r_1, r_2, r_3\}$. Recall that f is defined by:*

$$f(D_1, D_2) = \begin{cases} 0 & \textit{if } r_3 \in D_1, r_2 \notin D_1 \textit{ or } r_3 \notin D_1, r_2 \in D_1 \\ 0 & \textit{if } r_3 \in D_2, r_2 \notin D_2 \textit{ or } r_3 \notin D_2, r_2 \in D_2 \\ 1 & \textit{otherwise} \end{cases}$$

Obviously, $br(M)$ can be partitioned into $A = \{r_2, r_3\}$ and $B = \{r_3\}$, because for all $D_1, D_2 \subseteq br(M)$ holds that $f(D_1 \cap B, D_2 \cap B) = 1$ and $f(D_1 \cap A, D_2 \cap A) = f(D_1, D_2)$. The resulting sets for decomposing C_3 then are as follows: $br_3^A = \{d1(r_2), d2(r_2), d1(r_3), d2(r_3)\}$, $br_3^B = \{d1(r_1), d2(r_1)\}$, $OUT_3^A = \{removed_{r_2}, uncond_{r_2}, removed_{r_3}, uncond_{r_3}\}$, and $OUT_3^B = \{removed_{r_1}, uncond_{r_1}\}$.

Since the knowledge base kb_3 of M^f uses ASP, we can easily get the knowledge bases kb_3^A and kb_3^B by partitioning kb_3:

$$\begin{aligned} kb_3^A = \{\ & removed_{r_2} \leftarrow not\ not\_removed_{r_2}. \\ & removed_{r_3} \leftarrow not\ not\_removed_{r_3}. \\ & \bot \leftarrow removed_{r_3}, not\ removed_{r_2}. \\ & \bot \leftarrow not\ removed_{r_3}, removed_{r_2}. \\ & \bot \leftarrow uncond_{r_3}, not\ uncond_{r_2}. \\ & \bot \leftarrow not\ uncond_{r_3}, uncond_{r_2}.\} \\ kb_3^B = \{\ & removed_{r_1} \leftarrow not\ not\_removed_{r_1}. \end{aligned}$$

The resulting decomposed MCS then is $M' = (C_1', C_2', C_3^A, C_3^B)$ whose remaining details follow Definition 3.12. By Proposition 3.11 the diagnoses of M' correspond one-to-one to the diagnoses of M^f. Since diagnoses with protected bridge rules are directly based on ordinary diagnoses, these results carry over, and M' can be used to obtain minimal filtered diagnoses of M, where the filter itself is realised in a decentralised way.

Regarding preferences and the clone encoding $M^{\preceq}$, the decomposition results, however, are not yet sufficient, because of the bridge rule t_{max} which must be forced to be made condition-free whenever the diagnosis of the second clone in $M^{\preceq}$ is less (or equally) preferred than the diagnosis of the first component. This requires some additional information flow between the decomposed contexts; in fact, it only requires one additional bridge rule that signals to that context of the decomposition where t_{max} belongs to, that t_{max} should be forced to be made condition-free. Intuitively, one protected bridge rule is sufficient for that, since it can transport all that is required and by protecting it, it is guaranteed to be not modified by any diagnosis. We assume that the aforementioned decomposition results can be lifted to diagnoses with protected bridge rules such that additional information flow between the decomposed contexts is allowed using protected bridge rules. Such decomposition results are interesting in their own, but outside the scope of this thesis.

Context complexity $\mathcal{CC}(M)$	Checking $(D_1, D_2) \overset{?}{\in}$ $D^{\pm}_m(M)$ MCSD$_m$	$D^{\pm}_m(M, br_P)$ MCSDP$_m$	$D^{\pm}(M, br_P, br_H)$ MCSDPH	$D^{\pm}_{m,t_{max}}(M, br_P, br_H)$ MCSDPH$_{m,t_{max}}$
$\mathbf{P}$	$\mathbf{D_1^P}$	$\mathbf{D_1^P}$	$\mathbf{D_1^P}$	in $\mathbf{\Pi_2^P}$
$\mathbf{NP}$	$\mathbf{D_1^P}$	$\mathbf{D_1^P}$	$\mathbf{D_1^P}$	in $\mathbf{\Pi_2^P}$
$\mathbf{\Sigma_i^P}, i \geq 1$	$\mathbf{D_i^P}$	$\mathbf{D_i^P}$	$\mathbf{D_i^P}$	in $\mathbf{\Pi_{i+1}^P}$
Proposition	cf. Section 3.4	4.7	4.8	4.9

Table 4.1: Membership results of the computational complexity of deciding whether a diagnosis candidate is protected, prioritised-minimal, or an mpm-diagnosis. If at least one context is hard for the given context complexity, then hardness also holds for MCSDP$_m$ and MCSDPH.

4.4 Computational Complexity

This section analyses the computational complexity of the more sophisticated notions of diagnosis in an MCS that are introduced in the preceding sections. We show that deciding whether a pair $D = (D_1, D_2)$ of sets of bridge rules is a subset-minimal diagnosis with protected bridge rules br_P of M is not harder than deciding whether D is a subset-minimal diagnosis, i.e., deciding whether $D \in D^{\pm}_m(M, br_P)$ holds is not harder than deciding whether $D \in D^{\pm}_m(M)$ holds. We also demonstrate that the same is true for prioritised-minimal diagnoses, i.e., deciding whether $D \in D^{\pm}(M, br_P, br_H)$ holds is as hard as deciding whether $D \in D^{\pm}_m(M)$ holds. This notion of diagnosis can be applied to the plain encoding $M^{pl\preceq}$ for total preference orders to select minimal $\preceq$-preferred diagnoses according to a total preference order $\preceq$. The drawback of this approach, however, is the exponentially many bridge rules in $M^{pl\preceq}$.

Since the clone encoding $M^{\preceq}$ incurs no exponential blow-up of bridge rules, it is reasonable to expect that the computational complexity of deciding whether D is a subset-minimal prioritised-minimal diagnosis wrt. t_{max}, i.e., deciding whether $D \in D^{\pm}_{m,t_{max}}(M, br_P, br_H)$ holds, is higher than the one of deciding whether $D \in D^{\pm}_m(M)$ holds. Indeed, we prove in the following that deciding whether $D \in D^{\pm}_m(M, br_P, br_H)$ holds is in $\mathbf{\Pi_2^P}$ for context complexity $\mathcal{CC}(M)$ in $\mathbf{NP}$. In contrast to this deciding whether $D \in D^{\pm}_m(M)$ holds is in $\mathbf{D_1^P}$ for the same context complexity.

Since deciding whether $t(D) \in D^{\pm}_{m,t_{max}}(M^{\preceq}, br_P, br_H)$ holds is only a means to decide whether $D \in D^{\pm}_{m,\preceq}(M)$ holds, we also investigate the lower bound for the latter problem. We prove that it is $\mathbf{\Pi_2^P}$-hard, hence we show that the clone encoding using $M^{\preceq}$ and $D^{\pm}_{m,t_{max}}(M^{\preceq}, br_P, br_H)$ is in fact worst-case optimal.

In Table 4.1 the results for the introduced notions of diagnosis are summarised; the table also shows the names of the corresponding decision problems as introduced in the remainder of this section. The rest of this section finally contains statements and proofs of these complexity results.

Recall that MCSD$_m$ is the problem of deciding whether for a given $D \in 2^{br(M)} \times 2^{br(M)}$ it holds that $D \in D^{\pm}_m(M)$. Analogous to that, we denote by MCSDP$_m$ the problem of deciding whether $D \in D^{\pm}_{m}(M, br_P)$ holds for given $D \in 2^{br(M)} \times 2^{br(M)}$, MCS M, and $br_P \subseteq br(M)$.

Proposition 4.7. *The computational complexity (hardness and membership) of* MCSD_m *in Table 4.1 is the same as for* MCSDP_m.

Proof. In the remainder of this proof we assume **C** to be the computational complexity of MCSD_m.

Membership: In the following we give a polynomial-time reduction $\leq^p_m$ from MCSDP_m to MCSD_m. Given an instance of MCSDP_m, i.e., given an MCS M, a set $br_P \subseteq br(M)$, and a diagnosis candidate $D \in 2^{br(M)} \times 2^{br(M)}$, we define $\leq^p_m$ such that

$$(M, br_P, D) \mapsto \begin{cases} (M, D) & \text{if } D_1 \cap br_P = \emptyset = D_2 \cap br_P \text{ where } D = (D_1, D_2) \\ (M_\bot, (\emptyset, \emptyset)) & \text{otherwise} \end{cases}$$

where $M_\bot = (C_\bot)$, $C_\bot = (L^{asp}_\Sigma, kb_\bot, br_\bot)$, $br_\bot = \{(1 : a) \leftarrow \top.\}$, and $kb_\bot = \{\bot \leftarrow a.\}$ is such that $(\emptyset, \emptyset) \notin D^{\pm}_m(M_\bot)$. Intuitively, the reduction checks whether D contains bridge rules from br_P and if so, maps to an instance which is not in MCSD_m. If D contains no bridge rules from br_P, then $\leq^p_m$ simply drops br_P. Since the check whether D contains bridge rules of br_P is possible in polynomial time, $\leq^p_m$ is a polynomial-time many-one reduction.

It remains to show that indeed (M, br_P, D) is a yes-instance of MCSDP_m iff $\leq^p_m (M, br_P, D)$ is a yes-instance of MCSD_m.

"$\Rightarrow$": Let (M, br_P, D) be a yes-instance of MCSDP_m, i.e., $D \in D^{\pm}_m(M, br_P)$ holds. Then, $D = (D_1, D_2)$ is such that $D_1 \cap br_P = \emptyset = D_2 \cap br_P$, hence $\leq^p_m (M, br_P, D) = (M, D)$. By Proposition 4.1 it holds that $D^{\pm}_m(M, br_P) \subseteq D^{\pm}_m(M)$, hence it follows that $D \in D^{\pm}_m(M)$ holds, i.e., (M, D) is a yes-instance of MCSD_m.

"$\Leftarrow$": Let $\leq^p_m (M, br_P, D)$ be a yes-instance of MCSD_m. Note that it cannot be the case that $\leq^p_m (M, br_P, D) = (M_\bot, (\emptyset, \emptyset))$, because $(\emptyset, \emptyset) \notin D^{\pm}_m(M_\bot)$ contradicts that $\leq^p_m (M, br_P, D)$ is a yes-instance of MCSD_m. Consequently, it holds that $\leq^p_m (M, br_P, D) = (M, D)$ and thus $D = (D_1, D_2)$ is such that $D_1 \cap br_P = \emptyset = D_2 \cap br_P$. Furthermore, $D \in D^{\pm}_m(M)$ holds, thus it follows that $D \in D^{\pm}(M, br_P)$ holds. Assume that $D \notin D^{\pm}_m(M, br_P)$ holds. Then there exists $D' \subset D$ such that $D' \in D^{\pm}_m(M, br_P)$ holds. By Proposition 4.1 then follows that $D' \in D^{\pm}_m(M)$, which contradicts that $D \in D^{\pm}_m(M)$. Therefore no such D' exists and it follows that $D \in D^{\pm}_m(M, br_P)$ holds.

Since $\leq^p_m$ is a polynomial reduction from MCSDP_m to MCSD_m, it follows that the computational complexity of MCSDP_m is in **C**, i.e., the same complexity class where MCSD_m is in.

Hardness: Let $D \in D^{\pm}_m(M)$ be hard for some complexity class **C**. Observe that by definition of diagnoses with protected bridge rules, it holds that $D \in D^{\pm}_m(M)$ is true iff $D \in D^{\pm}_m(M, \emptyset)$ is true. Since deciding whether $D \in D^{\pm}_m(M)$ is **C**-hard, it thus follows that deciding whether $D \in D^{\pm}_m(M, br_P)$ also is **C**-hard. □

Now we consider the problem MCSDPH which is defined as follows: given an MCS M, a diagnosis candidate $D \in 2^{br(M)} \times 2^{br(M)}$, protected bridge rules $br_P \subseteq br(M)$, and prioritised bridge rules $br_H \subseteq br(M)$, decide whether it holds for every $T \in D^{\pm}_m(M, br_P)$ that $T \subseteq_{br_H} D$ implies $T =_{br_H} D$. In other words, MCSDPH is the problem of deciding whether $D \in$

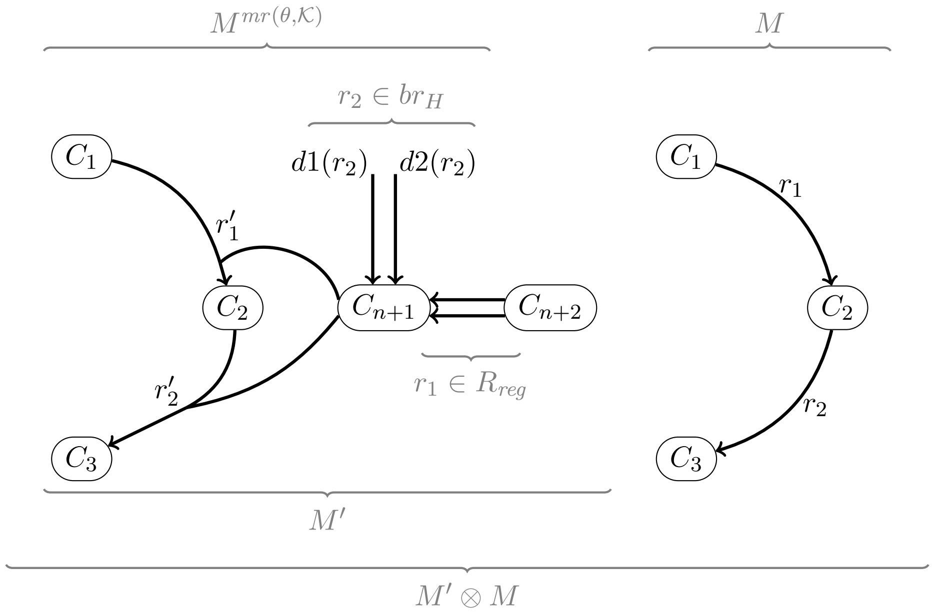


Figure 4.8: The reduction from MCSDPH to MCSDP$_m$ exemplified on the MCS $M = (C_1, C_2, C_3)$ with two bridge rules $br(M) = \{r_1, r_2\}$, with $r_1 : (2\colon b) \leftarrow (1\colon a).$, and $r_2 : (3\colon d) \leftarrow (2\colon c).$, protected bridge rules $br_P = \emptyset$, and prioritised bridge rules $br_H = \{r_2\}$, thus $R_{reg} = \{r_1\}$. Shown is the resulting MCS $M' \otimes M$; its components are indicated in gray.

$D^{\pm}(M, br_P, br_H)$ holds. As we later show, this problem is in the same complexity class as deciding whether $D \in D^{\pm}_m(M, br_P)$.

We now present a polynomial-time reduction from MCSDPH to MCSDP$_m$. We remark that a direct membership proof would be simpler, but the reduction is of interest in its own.

The underlying idea of the reduction is that, given an MCS M with protected bridge rules br_P and prioritised bridge rules br_H, we simulate the modifications of regular bridge rules inside the resulting MCS. The set R_{reg} of regular (non-prioritised, non-protected) bridge rules is $R_{reg} = br(M) \setminus br_H \setminus br_P$ and their modifications can be simulated by using a meta-reasoning transformation $M^{mr(\theta,\mathcal{K})} = (C_1, \dots, C_{n+1})$ where the bridge rules of C_{n+1} which correspond to modifications of bridge rules in R_{reg} take their values from an additional context C_{n+2} which generates all possible modifications, i.e., every possible modification corresponds to an acceptable belief set of C_{n+2}. In the resulting MCS $M' = (C_1, \dots, C_{n+2})$ we protect all bridge rules except those that correspond to modifications of bridge rules in br_H, i.e., every diagnosis of M' corresponds to one (or more) diagnoses of M, but the diagnoses of M' only contain bridge rules corresponding to subsets of br_H. Hence any minimal diagnosis of M' is $\subseteq_{br_H}$-minimal wrt. M. To ensure that the diagnosis indeed is $\subseteq$-minimal, we further add a copy of M, i.e., the resulting MCS is $M' \otimes M$ where M' ensures minimality wrt. $\subseteq_{br_H}$ and M ensures minimality wrt. $\subseteq$. An illustration of the resulting MCS is given in Figure 4.8.

We now give the formal definition of the reduction. Given an MCS M and a set $R_{reg} \subseteq br(M)$, let $\mathcal{K} = \emptyset$ and let θ be such that for all $D_1, D_2 \subseteq br(M)$ the property $\theta(D_1, D_2, \emptyset)$

holds. We craft an MCS based on the meta-reasoning MCS $M^{mr(\theta,\mathcal{K})} = (C_1, \dots, C_n, C_{n+1})$ to obtain an MCS where the modification of all bridge rules in R_{reg} is hidden in the set of possible belief states. To that end, we introduce another context C_{n+2} with no bridge rules and whose acceptable belief sets encode all respective modifications of bridge rules of R_{reg}. Formally, $C_{n+2} = (L_{\Sigma}^{asp}, kb_{n+2}, \emptyset)$ where

$$\begin{aligned} kb_{n+2} = \; & \{\mathit{not\_removed}_r \leftarrow \mathit{not}\ \mathit{removed}_r \mid r \in R_{reg}\} \\ & \cup \{\mathit{not\_removed}_r \leftarrow \mathit{not}\ \mathit{removed}_r \mid r \in R_{reg}\} \\ & \cup \{\mathit{uncond}_r \leftarrow \mathit{not}\ \mathit{not\_uncond}_r \mid r \in R_{reg}\} \\ & \cup \{\mathit{not\_uncond}_r \leftarrow \mathit{not}\ \mathit{uncond}_r \mid r \in R_{reg}\}. \end{aligned}$$

Observe that for every $D_1, D_2 \subseteq R_{reg}$, there is a belief set S_{n+2} with $\{\mathit{not\_removed}_r \mid r \in D_1'\} \cup \{\mathit{uncond}_r \mid r \in D_2\} = S_{n+2} \cap \big(\{\mathit{not\_removed}_r, \mathit{uncond}_r \mid r \in R_{reg}\}\big)$; furthermore, since C_{n+2} has no bridge rules, it follows that $S_{n+2} \in \mathbf{ACC}_{n+2}(kb_{n+2} \cup \mathit{app}(br_{n+2}, S))$ holds for all belief states $S = (S_1, \dots, S_{n+2})$.

Recall that all bridge rules of C_{n+1} are either of the form $(n+1 : \mathit{not\_removed}_r) \leftarrow \top.$ or of the form $(n+1 : \mathit{uncond}_r) \leftarrow \bot.$ (with $r \in br(M)$). Let $C_{n+1} = (L, kb_{n+1}, br_{n+1})$; then $C'_{n+1} = (L, kb_{n+1}, br'_{n+1})$ where

$$br'_{n+1} = \{(n+1 : \mathit{not\_removed}_r) \leftarrow (n+2 : \mathit{not\_removed}_r). \mid r \in br(M), r \in R_{reg}\} \tag{4.33}$$

$$\cup \{(n+1 : \mathit{uncond}_r) \leftarrow (n+2 : \mathit{uncond}_r). \mid r \in br(M), r \in R_{reg}\} \tag{4.34}$$

$$\cup \{(n+1 : \mathit{not\_removed}_r) \leftarrow \top. \mid r \in br(M), r \notin R_{reg}\} \tag{4.35}$$

$$\cup \{(n+1 : \mathit{uncond}_r \leftarrow \bot. \mid r \in br(M), r \notin R_{reg}\}. \tag{4.36}$$

Intuitively, C'_{n+1} equals C_{n+1} but those bridge rules occurring in R_{reg} refer to C_{n+2}. Similar to the meta-reasoning encoding, we denote by $d1(r)$ the corresponding bridge rule of form (4.35) and by $d2(r)$ the corresponding bridge rule of form (4.36). We also extend these notions to sets of bridge rules and, e.g., write $d1(br(M) \setminus R_1)$ to denote the set of bridge rules of line (4.35).

Finally, we call $M' = (C_1, \dots, C_n, C'_{n+1}, C_{n+2})$ the *meta-guessing* MCS for M and R_{reg}. The effect of the redirection to C_{n+2} is that the acceptable belief sets of C_{n+2} guess all possible modifications. The rest of M' behaves like an ordinary meta-reasoning encoding. The protected bridge rules of M' is the set $br_P{}' = br_{M'} \setminus (d1(br(M) \setminus R_{reg}) \cup d2(br(M) \setminus R_{reg}))$, i.e., all bridge rules are protected except those bridge rules of C_{n+1} which do not correspond to bridge rules in R_{reg}.

Now the reduction $\leq_m^p$ from MCSDPH to MCSDP$_m$ is as follows:

$$(M, (D_1, D_2), br_P, br_H) \mapsto (M' \otimes M, (D'_1, D'_2), br_P{}'')$$

where M' is the meta-guessing MCS wrt. $R_{reg} = br(M) \setminus br_P \setminus br_H$ and $br_P{}'' = br_P{}' \cup I(br_P)$ where I is the mapping wrt. $M' \otimes M$ and $br_P{}'$ is the set of protected bridge rules of the meta-guessing MCS M'; furthermore $D'_1 = I(D_1) \cup d1(D_1 \cap br_H)$ and $D'_2 = I(D_2) \cup d2(D_2 \cap br_H)$, i.e., (D'_1, D'_2) contains a diagnosis candidate of M, and it contains a diagnosis candidate over br_H with modifications to the remaining bridge rules of M being simulated by M'.

Observe that $(M' \otimes M, (D'_1, D'_2), br_P'')$ is polynomial in the size of $(M, (D_1, D_2), br_P, br_H)$, because $M' \otimes M$ only has four times as many bridge rules as M and all other sets are subsets of these bridge rules. Also note that $(M' \otimes M, (D'_1, D'_2), br_P'')$ can be computed in polynomial time, more precisely, it can be computed in time linear in the size of $(M, (D_1, D_2), br_P, br_H)$.

In the following, we show that $\leq^p_m$ indeed is a correct reduction from MCSDPH to MCSDP$_m$.

Lemma 4.14. *$\leq^p_m$ is a polynomial-time reduction from* MCSDPH *to* MCSDP$_m$.

Proof. "⇒": Let $(M, (D_1, D_2), br_P, br_H)$ be a yes-instance of MCSDPH, i.e., $(D_1, D_2) \in D^{\pm}(M, br_P, br_H)$ holds. We have to show that $(D'_1, D'_2) \in D^{\pm}_m(M' \otimes M, br_P'')$ also holds.

From $(D_1, D_2) \in D^{\pm}(M, br_P, br_H)$ and (4.32) it follows that $(D_1, D_2) \in D^{\pm}_m(M, br_P)$ holds.

By Proposition 4.1 it then holds that $(D_1, D_2) \in D^{\pm}_m(M)$, thus there exists $S = (S_1, \ldots, S_n)$ with $S \in \text{EQ}(M[D_1, D_2])$. We now show that $(d1(D_1 \cap br_H), d2(D_2 \cap br_H)) \in D^{\pm}_m(M', br_P')$ holds; to that end consider the belief state $S' = (S_1, \ldots, S_n, S_{n+1}, S_{n+2})$ where

$$S_{n+1} = \{\mathit{removed}_r \mid r \in r \in D_1\} \cup \{\mathit{uncond}_r \mid r \in D_2\}$$
$$S_{n+2} = \{\mathit{not\_removed}_r \mid r \in D_1 \setminus br_H\} \cup \{\mathit{uncond}_r \mid r \in D_2 \setminus br_H\}.$$

By construction of C_{n+2}, it holds that

$$S_{n+2} \in \mathbf{ACC}_{n+2}(kb_{n+2} \cup \mathit{app}(br_{n+2}(M'[d1(D_1 \cap br_H), d2(D_2 \cap br_H)]), S').$$

Consider the set of applicable bridge rules of C_{n+1} under S' and the diagnosis candidate $(D_1 \cap br_H, D_2 \cap br_H)$ (where $R_{reg} = (br(M) \setminus br_P \setminus br_H)$:

$$\begin{aligned}
\{\varphi(r) \mid r \in & \mathit{app}(br_{n+1}(M'[d1(D_1 \cap br_H), d2(D_2 \cap br_H)]), S')\} \\
&= \{\mathit{not_removed}_r \mid r \in br(M), r \notin R_{reg}, r \notin D_1 \setminus br_H\} \\
&\quad \cup \{\mathit{not_removed}_r \mid r \in br(M), r \in R_{reg}, r \notin D_1 \cap br_H\} \\
&\quad \cup \{\mathit{uncond}_r \mid r \in br(M), r \notin R_{reg}, r \in D_2 \cap br_H\} \\
&\quad \cup \{\mathit{uncond}_r \mid r \in br(M), r \in R_{reg}, r \in D_2 \setminus br_H\} \\
&= \{\mathit{not_removed}_r \mid r \in br(M), r \notin D_1\} \\
&\quad \cup \{\mathit{uncond}_r \mid r \in br(M), r \in D_2\} \\
&=: H
\end{aligned}$$

Since $S_{n+1} = \{\mathit{removed}_r \mid r \in r \in D_1\} \cup \{\mathit{uncond}_r \mid r \in D_2\}$ and $\theta(D_1, D_2, \emptyset)$ holds, it follows from the definition of C_{n+1} (cf. Definition 4.18 and Lemma 4.4) that $S_{n+1} \in \mathbf{ACC}_{n+1}(kb_{n+1} \cup H)$ holds.

Following the reasoning in Lemma 4.3 it is then possible to construct a proof showing that for all $1 \leq i \leq n$ it holds that

$$\mathit{app}(br_i(M[D_1, D_2]), S) = \mathit{app}(br_i(M'[d1(D_1 \cap br_H), d2(D_2 \cap br_H)]), S').$$

Since the semantics $\mathbf{ACC}_i$ and knowledge base kb_i of each context C_i are the same in M and M', it then follows from $S \in \text{EQ}(M[D_1, D_2])$ that for all $1 \leq i \leq n$ holds $S_i \in \mathbf{ACC}_i(kb_i \cup \mathit{app}(br_i(M'[d1(D_1 \cap br_H), d2(D_2 \cap br_H)]), S'))$.

In summary, it holds that $S' \in \text{EQ}(M'[d1(D_1 \cap br_H), d2(D_2 \cap br_H)])$.

Since $(D_1, D_2) \in D^{\pm}(M)$ holds, we then conclude from Proposition 3.10 that $(I(D_1) \cup d1(D_1 \cap br_H), I(D_2) \cup d2(D_2 \cap br_H) = (D'_1, D'_2) \in D^{\pm}(M' \otimes M)$ holds. Note that $D'_1 \cap br_P'' = \emptyset = D'_2 \cap br_P''$, hence $(D'_1, D'_2) \in D^{\pm}(M' \otimes M, br_P'')$ also holds.

It remains to show that $(D'_1, D'_2) \in D^{\pm}_m(M' \otimes M, br_P'')$. Towards contradiction assume that there exists $(T_1, T_2) \in D^{\pm}(M' \otimes M, br_P'')$ with $(T_1, T_2) \subset (D'_1, D'_2)$, i.e., by construction of $M' \otimes M$ it either is the case that $(T_1 \cap I(br(M)), T_2 \cap I(br(M)) \subset (D'_1 \cap I(br(M)), D'_2 \cap I(br(M)))$ holds or $(T_1 \cap br_{M'}, T_2 \cap br_{M'}) \subset (D'_1 \cap br_{M'}, D'_2 \cap br_{M'})$ holds.

In the former case, Proposition 3.10 implies that $(I^{-1}(T_1 \cap I(br(M))), I^{-1}(T_2 \cap I(br(M)))) \in D^{\pm}(M)$; furthermore, since $(D'_1 \cap I(br(M)), D'_2 \cap I(br(M))) = (D_1, D_2)$ it holds that $(I^{-1}(T_1 \cap I(br(M))), I^{-1}(T_2 \cap I(br(M)))) \subset (D_1, D_2)$. This contradicts that $(D_1, D_2) \in D^{\pm}(M, br_P, br_H)$.

In the latter case, i.e., $(T_1 \cap br_{M'}, T_2 \cap br_{M'}) \subset (D'_1 \cap br_{M'}, D'_2 \cap br_{M'})$, it holds that $(T_1 \cap br_{M'}, T_2 \cap br_{M'}) \subset (br_H, br_H)$ since all other bridge rules of $br_{M'}$ are contained in br_P''. Let S be a witnessing equilibrium, i.e., let $S = (S_1, \ldots, S_{n+2}) \in \text{EQ}(M'[(T_1 \cap br_{M'}, T_2 \cap br_{M'})])$ hold. Consider the modifications of bridge rules in $br(M) \setminus br_P \setminus br_H$ which are represented by S, i.e., consider $T'_1 = \{r \in br(M) \setminus br_P \setminus br_H \mid not\_removed_r \notin S_{n+2}\}$ and $T'_2 = \{r \in br(M) \setminus br_P \setminus br_H \mid uncond_r \in S_{n+2}\}$. It holds that $((T_1 \cap br_{M'}) \cup T'_1, (T_2 \cap br_{M'}) \cup T'_2)$ is a diagnosis candidate of M. Since $S \in \text{EQ}(M'[(T_1 \cap br_{M'}, T_2 \cap br_{M'})])$ holds and M' stems from $M^{mr(\theta,\mathcal{K})}$, one can show using Lemma 4.3 that $((T_1 \cap br_{M'}) \cup T'_1, (T_2 \cap br_{M'}) \cup T'_2) \in D^{\pm}(M, br_P)$ holds. This contradicts that $(D_1, D_2) \in D^{\pm}(M, br_P, br_H)$, because $((T_1 \cap br_{M'}) \cup T'_1, (T_2 \cap br_{M'}) \cup T'_2) \subset_{br_H} (D_1, D_2)$.

Therefore, no such (T_1, T_2) exists and it holds that $(D'_1, D'_2) \in D^{\pm}_m(M' \otimes M, br_P'')$.

"$\Leftarrow$": We prove the converse, i.e., we assume that $(M, (D_1, D_2), br_P, br_H)$ is not a yes-instance of MCSDPH and show that $\leq^p_m (M, (D_1, D_2), br_P, br_H) = (M' \otimes M, (D'_1, D'_2), br_P'')$ also is not a yes-instance of MCSDP$_m$. By assumption it therefore holds that $(D_1, D_2) \notin D^{\pm}(M, br_P, br_H)$ holds. From the definition of $D^{\pm}(M, br_P, br_H)$ we then obtain that either (i) $(D_1, D_2) \notin D^{\pm}_m(M, br_P)$ holds or (ii) that there exists $(D'_1, D'_2) \in D^{\pm}_m(M, br_P)$ with $(D'_1, D'_2) \subset_{br_H} (D_1, D_2)$.

In case (i) $(D_1, D_2) \notin D^{\pm}_m(M, br_P)$, hence by Proposition 3.10 it holds that $(I(D_1) \cup d1(D_1 \cap br_H), I(D_2) \cup d2(D_2 \cap br_H)) \notin D^{\pm}_m(M \otimes M', br_P'')$.

In case (ii) $(D'_1, D'_2) \in D^{\pm}_m(M, br_P)$ with $(D'_1, D'_2) \subset_{br_H} (D_1, D_2)$. W.l.o.g. we assume that (D'_1, D'_2) is minimal wrt. $\subset_{br_H}$, i.e., there exists no $(D''_1, D''_2) \in D^{\pm}_m(M, br_P)$ with $(D''_1, D''_2) \subset_{br_H} (D'_1, D'_2)$. This means that $(M, (D'_1, D'_2), br_P, br_H)$ is a yes-instance of MCSDPH. We can further assume that $(D_1, D_2) \in D^{\pm}_m(M, br_P)$ from (i).

Now consider $(T_1, T_2) = (I(D_1) \cup d1(D'_1 \cap br_H), I(D_2) \cup d2(D'_2 \cap br_H))$. Since $(M, (D'_1, D'_2), br_P, br_H)$ is a yes-instance of MCSDPH, the "$\Rightarrow$" direction above can be applied to it; this yields that $(d1(D'_1 \cap br_H), d2(D_2 \cap br_H)) \in D^{\pm}_m(M', br_P')$ holds. Applying Proposition 3.10 and the fact that $T_1 \cap br_P'' = \emptyset = T_2 \cap br_P''$ then implies that $(T_1, T_2) \in D^{\pm}_m(M' \otimes M, br_P'')$ holds. Note that $(T_1, T_2) \subset (D'_1, D'_2)$ holds which in turn implies that $(D'_1, D'_2) \notin D^{\pm}_m(M' \otimes M, br_P'')$ holds. In other words, $(M' \otimes M, (D'_1, D'_2), br_P'')$ is not a yes-instance of MCSDP$_m$.

In all cases, we showed that $\leq^p_m (M, (D_1, D_2), br_P, br_H)$ is not a yes-instance of MCSDP$_m$, which concludes the "⇐" direction of the proof.

In summary, we showed that $(M, (D_1, D_2), br_P, br_H)$ is a yes-instance of MCSDPH iff $(M' \otimes M, (D'_1, D'_2), br_P'') = \leq^p_m (M, (D_1, D_2), br_P, br_H)$ is a yes-instance of MCSDP$_m$, i.e., $\leq^p_m$ is a reduction from MCSDPH to MCSDP$_m$. Since $(M' \otimes M, (D'_1, D'_2), br_P'')$ can be computed in time linear in the size of $(M, (D_1, D_2), br_P, br_H)$, it furthermore holds that $\leq^p_m$ a polynomial-time reduction. □

The following Proposition shows, that MCSDPH indeed is in the same complexity class as deciding whether $D \in D^{\pm}_m(M, br_P)$ holds and hence whether $D \in D^{\pm}_m(M)$ holds.

Proposition 4.8. MCSDPH *is in* $\mathbf{D^P_i}$ *for context complexity* $\mathcal{CC}(M)$ *in* $\mathbf{\Sigma^P_i}$*,* $i \geq 1$*; if at least one context of* M *is hard for* $\mathbf{\Sigma^P_i}$*, then* MCSDPH *is* $\mathbf{D^P_i}$*-hard.*

Proof. Membership: By Lemma 4.14 it holds that $\leq^p_m$ is a polynomial-time reduction from MCSDPH to MCSDP$_m$, hence membership immediately follows.

Hardness: Note that MCSD$_m$ is $\mathbf{D^P_i}$-hard if at least one context is hard for $\mathbf{\Sigma^P_i}$ (cf. Section 3.4). Let M' and D' be any MCS and diagnosis candidate, respectively, used for showing hardness of MCSD$_m$ (i.e., M' is the result of the reduction showing hardness of MCSD$_m$ and D' is the diagnosis resulting from the reduction of M'). Now pick $br_P' = br_H' = \emptyset$.

By definition, it holds for all M, br_P, br_H and D, that $D \in D^{\pm}(M, br_P, br_H)$ implies $D \in D^{\pm}_m(M, br_P)$ which in turn implies $D \in D^{\pm}_m(M)$. Therefore, $D' \in D^{\pm}(M', br_P', br_H')$ implies that $D' \in D^{\pm}_m(M')$ holds. Furthermore, since $br_P' = br_H' = \emptyset$ it also follows from the definition of prioritised-minimal diagnosis and protected diagnosis that $D' \in D^{\pm}_m(M')$ implies $D' \in D^{\pm}(M', br_P', br_H')$. In summary, $D' \in D^{\pm}_m(M')$ holds iff $D' \in D^{\pm}(M', br_P', br_H')$ holds. Therefore MCSDPH also is $\mathbf{D^P_i}$-hard if at least one context of M is hard for $\mathbf{\Sigma^P_i}$. □

Next we consider the problem MCSDPH$_{t_{max}}$ which we define as follows: given an MCS M, a diagnosis candidate $D \in 2^{br(M)} \times 2^{br(M)}$, protected bridge rules $br_P \subseteq br(M)$, prioritised bridge rules $br_H \subseteq br(M)$, and $t_{max} \in br(M)$; decide whether the following both hold:

1. $t_{max} \in D_2$ with $D = (D_1, D_2)$ and
2. for all $T \in D^{\pm}_m(M, br_P)$ it holds that $T \subseteq_{br_H} D \Rightarrow T =_{br_H} D$.

Notice that MCSDPH$_{t_{max}}$ basically amounts to checking the presence of t_{max} in a diagnosis candidate of MCSDPH. As the following lemma shows, the complexity of the former is in the same complexity class as the latter.

Lemma 4.15. MCSDPH$_{t_{max}}$ *is in* $\mathbf{D^P_i}$ *for context complexity* $\mathcal{CC}(M)$ *in* $\mathbf{\Sigma^P_i}$ *(where* $i \geq 1$*).*

Proof. For membership, we give a reduction $\leq^p_m$ from MCSDPH$_{t_{max}}$ to MCSDPH as follows:

$$(M, D, br_P, br_H, t_{max}) \mapsto \begin{cases} (M, D, br_P, br_H) & \text{if } D = (D_1, D_2), t_{max} \in D_2 \\ (M_\perp, (\emptyset, \emptyset), br_{M_\perp}, \emptyset) & \text{otherwise} \end{cases}$$

where $M_\perp$ is the inconsistent MCS from the proof of Proposition 4.7, i.e., $(M_\perp, (\emptyset, \emptyset), br_{M_\perp}, \emptyset)$ is not a yes-instance of MCSDPH since the MCS is inconsistent but all its bridge rules are protected.

"⇒": Let $(M, D, br_P, br_H, t_{max})$ be a yes-instance of MCSDPH$_{t_{max}}$, this means that $D \in D^\pm(M, br_P, br_H)$ and $t_{max} \in D_2$ with $D = (D_1, D_2)$ hold. Then $D \in D^\pm(M, br_P, br_H)$ also holds, i.e., (M, D, br_P, br_H) is a yes-instance of MCSDPH.

"⇐": Let $(M, D, br_P, br_H, t_{max})$ be not a yes-instance of MCSDPH$_{t_{max}}$, i.e., let it not be the case that $D \in D^\pm(M, br_P, br_H)$ and $t_{max} \in D_2$ with $D = (D_1, D_2)$ both hold. In case $t_{max} \notin D_2$ it holds that $(\emptyset, \emptyset) \notin D^\pm(M_\perp, br_{M_\perp}, \emptyset)$ since $M_\perp$ is inconsistent but all its bridge rules are protected, i.e., $\leq^p_m$ maps to a no-instance of MCSDPH. In case $t_{max} \in D_2$ holds, it follows that $D \in D^\pm(M, br_P, br_H)$ does not hold by the assumption. Therefore (M, D, br_P, br_H) is not a yes-instance of MCSDPH. Hence in all cases, $\leq^p_m$ (M, br_P, br_H, t_{max}) is not a yes-instance of MCSDPH. □

We denote by MCSDPH$_{m,t_{max}}$ the problem of deciding given an MCS M, a diagnosis candidate $D \in 2^{br(M)} \times 2^{br(M)}$, and protected and prioritised bridge rules $br_P, br_H \subseteq br(M)$ whether $D \in D^\pm_{m,t_{max}}(M, br_P, br_H)$.

Proposition 4.9. MCSDPH$_{m,t_{max}}$ *is in* $\mathbf{\Pi^P_{i+1}}$ *for context complexity* $\mathcal{CC}(M)$ *in* $\mathbf{\Sigma^P_i}$ *(with* $i \geq 1$*).*

Proof. By Lemma 4.15 MCSDPH$_{t_{max}}$ is in $\mathbf{D^P_i}$ for context complexity $\mathcal{CC}(M)$ in $\mathbf{\Sigma^P_i}$.

Algorithm 1 decides whether $(D_1, D_2) \in D^\pm_{m,t_{max}}(M, br_P, br_H)$ holds using an oracle for MCSDPH$_{t_{max}}$, which checks that $(D_1, D_2) \in D^\pm(M, br_P, br_H)$ and $t_{max} \in D_2$ hold. Then it checks for each $T_1, T_2 \subseteq br(M)$ that $(T_1, T_2) \in D^\pm(M, br_P, br_H)$ and $(T_1, T_2) \subset (D_1, D_2)$ do not hold.

The for-each check clearly is in $\mathbf{coNP^{D^P_i}}$ since each check contains a call to a $\mathbf{D^P_i}$ oracle. Also note that MCSDPH$_{t_{max}}$ is in $\mathbf{D^P_i}$, hence it also is in $\mathbf{coNP^{D^P_i}}$. Since this complexity class is closed under conjunction, it follows that the overall algorithm is in the complexity class $\mathbf{coNP^{D^P_i}}$. Furthermore, it holds that $\mathbf{coNP^{D^P_i}} = \mathbf{coNP^{\Sigma^P_i}}$, hence $\mathbf{coNP^{D^P_i}} = \mathbf{\Pi^P_{i+1}}$. Thus checking whether $D \in D^\pm_{m,t_{max}}(M, br_P, br_H)$ holds is in $\mathbf{\Pi^P_{i+1}}$ for context complexity $\mathcal{CC}(M)$ in $\mathbf{\Sigma^P_i}$.

□

The previous decision problems arise from our approach at realising the selection of preferred and filtered diagnoses of an MCS. To give a full picture, we also investigate the complexity of the basic problem, i.e., we investigate the computational complexity of deciding whether $D \in D^\pm_{m,\preceq}(M)$ holds for a given MCS M, $D \in 2^{br(M)} \times 2^{br(M)}$, and an arbitrary preference order $\preceq$; we denote this decision problem by MCSD$_{\text{MPREF}}$.

As the following proposition shows, MCSD$_{\text{MPREF}}$ itself is $\mathbf{\Pi^P_2}$-hard even if the context complexity and deciding whether $D \preceq D'$ holds is tracktable. This result also shows that our approach of realising the selection of minimal $\preceq$-preferred diagnoses is worst-case optimal.

Proposition 4.10. MCSD$_{\text{MPREF}}$ *is* $\mathbf{\Pi^P_2}$*-hard, even if* $\mathcal{CC}(M)$ *and deciding whether* $D' \preceq D''$ *hold are both in* $\mathbf{P}$.

Algorithm 1: Deciding whether $(D_1, D_2) \in D^{\pm}_{m,t_{max}}(M, br_P, br_H)$ holds.

Input : MCS M, (D_1, D_2), br_P, and br_H with $D_1, D_2 \subseteq br(M)$, $br_P, br_H \subseteq br(M)$.
Output : accepts iff $(D_1, D_2) \in D^{\pm}(M, br_P, br_H)$
res $\leftarrow$ `oracle`$_{\texttt{MCSDPH}_{t_{max}}}$ $((D_1, D_2), M, br_P, br_H)$;
if res $=$ *NO* **then**
  fail;
end
foreach $T_1, T_2 \subseteq br(M)$ **do**
  res $\leftarrow$ `oracle`$_{\texttt{MCSDPH}_{t_{max}}}$ $((T_1, T_2), M, br_P, br_H)$;
  if res $=$ *YES* **then**
    if $(T_1, T_2) \subseteq_{br(M)\setminus br_H} (D_1, D_2)$, $(T_1, T_2) \neq (D_1, D_2)$, *and* $(T_1, T_2) \neq_{br(M)\setminus br_H} (D_1, D_2)$ **then**
      fail;
    end
  end
end
accept;

Proof. To prove that deciding MCSD$_{\text{MPREF}}$ is $\mathbf{\Pi_2^P}$-hard, we reduce the problem of deciding whether a $\mathbf{QBF_{2,\forall}}$-formula is valid to MCSD$_{\text{MPREF}}$.

A formula G is in $\mathbf{QBF_{2,\forall}}$ if it is of the form $\forall\vec{X}\exists\vec{Y} : F$ where F is a propositional formula over the variables of $\vec{X} = \{x_1, \ldots, x_k\}$ and $\vec{Y} = \{y_1, \ldots, y_\ell\}$ for some $k, \ell \in \mathbb{N}$. In the following we call F the *matrix* of G. Let $\psi[x/t]$ denote the substitution of the propositional variable x by $t \in \{\top, \bot\}$. The semantics is given in terms of valuations over $\vec{X}$ and $\vec{Y}$, where a valuation V_χ is a mapping $V_\chi : \vec{\chi} \to \{\top, \bot\}$ for $\chi \in \{X, Y\}$. Since the matrix F contains only variables in $\vec{X}$ and $\vec{Y}$ a valuation $V_X \cup V_Y$ allows to evaluate the truth value of F, where this evaluation clearly is in $\mathbf{P}$. The semantics of the quantifiers then is defined in the usual way in terms of valuations. Formally, $\forall\vec{X}\exists\vec{Y} : F$ is valid iff for all valuations V_X there exists a valuation V_Y such that $F[x_1/V_X(x_1), \ldots, x_k/V_X(x_k), y_1/V_Y(y_1), \ldots, y_\ell/V_Y(y_\ell)]$ evaluates to true.

We now define an MCS M^G whose single context utilises this evaluation of F given a valuation of all variables. Given a $\mathbf{QBF_{2,\forall}}$-formula G with $\vec{X}$, $\vec{Y}$, and F as above. Let br_1^X and br_1^Y be defined as follows:

$$br_1^X = \{(1{:}x) \leftarrow \top. \mid x \in \vec{X}\} \cup \{(1{:}\bar{x}) \leftarrow \top. \mid x \in \vec{X}\}$$
$$br_1^Y = \{(1{:}y) \leftarrow \top. \mid y \in \vec{Y}\} \cup \{(1{:}\bar{y}) \leftarrow \top. \mid y \in \vec{Y}\}.$$

Then $M^G = (C_1)$ where the bridge rules of C_1 are $br_1 = br_1^X \cup br_1^Y$. Note that br_1 is polynomial (even linear) in the size of G.

In the remainder of this proof, we write for a set R of bridge rules $\varphi(R)$ to denote the set of head-formulas of the bridge rules of R, i.e., $\varphi(R) = \{\varphi(r) \mid r \in R\}$. Let $H \subseteq \varphi(br_1)$, we say that H is consistent wrt. $\vec{X}$ (resp. $\vec{Y}$) iff for all $x \in \vec{X}$ (resp. $x \in \vec{Y}$) holds that $x \in H$ iff $\bar{x} \notin H$. We call H consistent if it is both consistent wrt. $\vec{X}$ and $\vec{Y}$. Note that if H is consistent wrt. $\vec{X}$

then there exists a consistent valuation V_X^H such that $V_X(x) = \top$ iff $x \in H$ and $V_X(x) = \bot$ iff $\bar{x} \in H$; if H is consistent wrt. $\vec{Y}$ then there exists a consistent valuation V_Y^H defined analogously.

Intuitively, we design $\mathbf{ACC}_1$ such that C_1 admits an equilibrium if the set H of heads of applicable bridge rules either is such that $H \cap br_1^X = \emptyset$ or H is consistent, hence represents a consistent valuation of variables of G and this valuation makes F true. In these cases, C_1 accepts the belief set $\emptyset$ while in all other cases, C_1 accepts no belief set (hence it makes M^G inconsistent). Formally, $\mathbf{ACC}_1$ and kb_1 are such that:

$$\mathbf{ACC}_1(kb_1 \cup H) = \begin{cases} \{\emptyset\} & \text{if } H \text{ is consistent and } F \text{ evaluates to true under } H, \text{ or} \\ & \text{if } H \cap \varphi(br_1^X) = \emptyset, \text{ or} \\ & \text{if } H \text{ is consistent wrt. } \vec{X} \text{ and } H \supseteq \varphi(br_1^Y), \\ \emptyset & \text{otherwise.} \end{cases}$$

Note that $\mathcal{CC}(M^G)$ is in $\mathbf{P}$ since deciding whether $H \cap \varphi(br_1^X) = \emptyset$, deciding whether H is consistent, and deciding whether F evaluates to true under H are all possible in polynomial time. Furthermore note that there is only one belief state $S_\emptyset = (\emptyset)$ for M^G.

To give a concrete instance of $\mathbf{ACC}_1$ and kb_1, let $F = \{c_1, \dots, c_m\}$ be given as a set of clauses each of the form $c_\ell = (l_{\ell_1} \vee l_{\ell_2} \vee \ldots \vee l_{\ell_k})$ with $k \in \mathbb{N}$. We associate with each clause c_ℓ of this form a set of rules

$$\begin{aligned} lp(c_\ell) = \{\ & clause\_c_\ell \leftarrow l_{\ell_1}. \quad \ldots \quad clause\_c_\ell \leftarrow l_{\ell_j}. \\ & clause\_c_\ell \leftarrow \overline{l_{\ell_{j+1}}}. \quad \ldots \quad clause\_c_\ell \leftarrow \overline{l_{\ell_k}}.\} \end{aligned}$$

where l_{ℓ_1} to l_{ℓ_j} are the positive literals and $l_{\ell_{j+1}}$ to l_{ℓ_k} are the negative literals of c_ℓ.

Finally, $C_1 = (L_\Sigma^{asp}, kb_1, br_1)$ uses the abstract logic of ASP and kb_1 is as follows:

$$\begin{aligned} kb_1 = \{ \quad & consistentX \leftarrow not\ inconsistentX. \\ & inconsistentX \leftarrow x, \bar{x}. && \forall x \in \vec{X} \\ & consistentY \leftarrow not\ inconsistentY. \\ & inconsistentY \leftarrow y, \bar{y}. && \forall y \in \vec{Y} \\ & \bot \leftarrow not\ ok. \\ & ok \leftarrow consistentX, consistentY, trueF. \\ & ok \leftarrow not\ nonempty\_intersect. \\ & ok \leftarrow consistentX, not\ notfullY. \\ & nonempty\_intersect \leftarrow x. && \forall x \in \vec{X} \\ & nonempty\_intersect \leftarrow \bar{x}. && \forall x \in \vec{X} \\ & notfullY \leftarrow not\ y. && \forall y \in \vec{Y} \\ & notfullY \leftarrow not\ \overline{y}. && \forall y \in \vec{Y} \\ & trueF \leftarrow clause\_c_1, \ldots, clause\_c_k. && \text{for } F = \{c_1, \ldots, c_k\} \\ \} \cup \{ & r \in lp(c_\ell) \mid c_\ell \in F\} \end{aligned}$$

Observe that kb_1 is a stratifiable logic program while bridge rules only add facts, thus $\mathbf{ACC}_1(kb_1 \cup H)$ can be computed in polynomial time. Also note that kb_1 is polynomial (even linear) in the size of G.

Next, we define $\preceq$ in such a way that the diagnosis candidate $D_{valid} = (br_1^X, \emptyset)$ is in $D^{\pm}_{m,\preceq}(M^G)$ iff G is valid. Formally, $\preceq$ is such that:

$$
\begin{aligned}
(D_1, D_2) \preceq (D_1', D_2') \text{ holds iff } & D_2 = D_2' = \emptyset, \\
& (D_1, D_2) \neq D_{valid} \neq (D_1', D_2'), \\
& D_1 \cap br_1^X = D_1' \cap br_1^X, \text{ and} \\
& D_1 \cap br_1^Y \subseteq D_1' \cap br_1^Y \text{ all hold.}
\end{aligned}
$$

Observe that deciding whether $D \preceq D'$ holds clearly is in $\mathbf{P}$. Further note that M^G is polynomial in the size of G since kb_1 and br_1 are both polynomial (even linear) in the size of G. We now show that $D_{valid} \in D^{\pm}_{m,\preceq}(M^G)$ holds iff G is valid. In some abuse of notation, in the following we write $M[D]$ to denote the MCS obtained from modifying M according to a diagnosis candidate D, i.e., $M[D]$ with $D = (D_1, D_2)$ here denotes $M[D_1, D_2]$.

"$\Rightarrow$": Let $D_{valid} \in D^{\pm}_{m,\preceq}(M^G)$ hold. Towards contradiction, assume that G is not valid, i.e., there exists a valuation V_X for $\vec{X}$ such that no valuation V_Y for $\vec{Y}$ makes F true. Let $R \subseteq br_1^X$ be such that $V_X^{\varphi(R)} = V_X$ and consider the diagnosis $D = (br_1^X \setminus R, \emptyset)$. Let $H = \{\varphi(r) \mid r \in app(br_1(M^G[D]), S_\emptyset)\}$ and observe that H is consistent wrt. $\vec{X}$ since R is consistent. Since $(br_1^X \setminus R) \cap br_1^Y = \emptyset$, if follows that $H \cap \varphi(br_1^Y) = \varphi(br_1^Y)$ and it thus holds that $\{\emptyset\} \in \mathbf{ACC}_1(kb_1 \cup H)$, i.e., $S_\emptyset$ is an equilibrium of $M^G[D]$, hence $D \in D^{\pm}(M^G)$. Further note that $D \subset D_{valid}$ holds. Since $D_{valid} \in D^{\pm}_{m,\preceq}(M^G)$ and $D \subset D_{valid}$, it follows that $D \notin D^{\pm}_{\preceq}(M^G)$ holds; i.e. there exists a diagnosis $D' \in D^{\pm}(M^G)$ such that $D' \preceq D$ and $D \not\preceq D'$ both hold, which implies that $D' \neq D$.

Let $D' = (D_1', D_2')$ and $D = (D_1, D_2)$; from the definition of $\preceq$ we obtain that $D_2' = \emptyset$, $D' \neq D_{valid}$, $D_1' \cap br_1^X = D_1 \cap br_1^X$, and $D_1' \cap br_1^Y \subseteq D_1 \cap br_1^Y$ all hold. Let $H' = \{\varphi(r) \mid r \in app(br_1(M^G[D']), S_\emptyset)\}$ and observe that H' is consistent wrt. $\vec{X}$ since D_1 is consistent wrt. $\vec{X}$ and $D_1' \cap br_1^X = D_1 \cap br_1^X$. Since $D' \neq D$ holds, it is the case that $D_1' \cap br_1^Y \subset D_1 \cap br_1^Y$ and thus $D_1 \cap br_1^Y \neq \emptyset$, i.e., $H \cap \varphi(br_1^Y) \neq \varphi(br_1^Y)$. This contradicts with $H \cap \varphi(br_1^Y) = \varphi(br_1^Y)$ established earlier. Therefore no such D exists and consequently no valuation V_X exists such that all valuations V_Y make F false, i.e., G is valid.

"$\Leftarrow$": Let G be valid, i.e., for every valuation of $x_1, \ldots, x_k$ there exists a valuation of $y_1, \ldots, y_\ell$ such that F evaluates to true. Observe that $br_1(M^G[D_{valid}]) = br_1^Y$, hence $H = app(br_1(M^G[D_{valid}]), S_\emptyset)$ is such that $H \cap \varphi(br_1^X) = \emptyset$, thus $\mathbf{ACC}_1(kb_1 \cup H) = \{\emptyset\}$ and $S_\emptyset$ is a witnessing equilibrium of $D_{valid} \in D^{\pm}(M^G)$. Furthermore, since D_{valid} is, by definition of $\preceq$, in no relation to any other diagnosis candidate it thus follows that $D_{valid} \in D^{\pm}_{\preceq}(M^G)$.

It remains to show that D_{valid} is subset-minimal among all diagnoses in $D^{\pm}_{\preceq}(M^G)$. Consider any $D' \subset D_{valid}$, i.e., $D' = (D_1', \emptyset)$ where $D_1' \subset br_1^X$. Recall that D' is not a diagnosis, if there exists no witnessing equilibrium; since $S_\emptyset$ is the only belief state of M^G, it follows that D' is a diagnosis if and only if $S_\emptyset$ is an equilibrium of $M^G[D']$. In the following, let $H' = app(br_1(M^G[D']), S_\emptyset)$. Since $D_1' \subset br_1^X$ holds, it follows that $H' \supseteq \varphi(br_1^Y)$, because for any $r \in br_1^Y$ it holds that $body(r) = \{\top\}$, i.e., r is applicable in any belief state. Since $H' \supseteq \varphi(br_1^Y)$ holds, it cannot be the case that H' is consistent wrt. $\vec{Y}$; thus H' is not consistent. Furthermore, by $D'_1 \subset br_1^X$ it follows that $H' \cap \varphi(br_1^X) \neq \emptyset$. By the definition of $\mathbf{ACC}_1$ it then follows that D' only is a diagnosis if H' is consistent wrt. $\vec{X}$.

Assume that H' is consistent wrt. $\vec{X}$ then $V_X^{H'}$ is a consistent valuation for variables in $\vec{X}$. Since G is valid and all variables in $\vec{X}$ are $\forall$-quantified, there exists a valuation V_Y for the variables of $\vec{Y}$ such that F evaluates to true under $V_X^{H'}$ and V_Y. Let $R \subset br_1^Y$ be the set of bridge rules consistent wrt. $\vec{Y}$ such that $V_Y^{\varphi(R)} = V_Y$ and consider the diagnosis candidate $D'' = (D'_1 \cup (br_1^Y \setminus R), \emptyset)$. Let $H'' = \{\varphi(r) \mid r \in app(br_1(M^G[D'']), S_\emptyset)\}$ and observe that H'' is consistent since D'_1 and R both are consistent. Furthermore, $V_X^{H''} = V_X^{H'}$ and $V_Y^{H''} = V_Y$, thus F evaluates to true under H'', hence $S_\emptyset$ is an equilibrium of $M[D'']$ and $D'' \in D^\pm(M^G)$ holds.

Now consider whether $D'' \preceq D'$ holds: $D' = (D'_1, \emptyset)$, $D'' = (D'_1 \cup (br_1^Y \setminus R), \emptyset)$, $D'' \neq D_{valid} \neq D'$, $D'_1 \cap br_1^X = (D'_1 \cup (br_1^Y \setminus R)) \cap br_1^X$, and $D'_1 \cap br_1^Y \subseteq (D'_1 \cup (br_1^Y \setminus R)) \cap br_1^Y$ all hold. Therefore $D'' \preceq D'$ holds. Since $R \subset br_1^Y$ holds, it follows that $br_1^Y \setminus R \neq \emptyset$ and by $D'_1 \subset br_1^X$ it then follows that $(D'_1 \cup (br_1^Y \setminus R)) \cap br_1^Y \subseteq D'_1 \cap br_1^Y$ does not hold. Therefore $D' \preceq D''$ does not hold and consequently, it holds that $D' \notin D^\pm_\preceq(M^G)$.

Since $D' \subset D_{valid}$ was chosen arbitrary, it follows that D_{valid} is subset-minimal among all diagnoses in $D^\pm_\preceq(M^G)$, hence $D_{valid} \in D^\pm_{m,\preceq}(M^G)$ holds.

In summary, it thus follows that the above reduction from $\mathbf{QBF}_{\mathbf{2},\forall}$ to MCSDmpref is correct. Since $\mathbf{QBF}_{\mathbf{2},\forall}$ is $\mathbf{\Pi_2^P}$-complete, this proves that MCSDmpref is $\mathbf{\Pi_2^P}$-hard. □

4.5 Summary

In this chapter we addressed the problem of identifying and selecting among all diagnoses of an MCS those which are most preferred. We considered two basic types of preference: filters, which allow to discard diagnoses that do not fulfil certain criteria, and preference orders which allow to compare diagnoses. Since MCS are a flexible framework that is open to integrate information from many different logical formalisms, we think it is necessary that the formalism in which filters and preferences are specified also is open to the user's choice.

To achieve this, we used the same concepts that underlie the MCS formalism, i.e., if the user can specify its conditions on diagnoses in the same way as specifying a context of an MCS, then in principle any logic formalism may be used to do so. We thus developed several techniques for meta-reasoning about diagnoses in MCS, i.e., given an MCS M and some filter or preference order it is shown in the previous sections how to transform these into an MCS M' such that the diagnoses of M' correspond one-to-one to the filtered or most-preferred diagnoses of M.

In order to do so, we first presented filters on diagnoses and preference orders on diagnoses in their most general form. We also showed that these can capture well-known formalisms for preferences specification like CP-nets [25]. Then we presented two approaches at meta-reasoning where the first observes the beliefs in the body and knowledge-base formulas in the heads of existing bridge rules, and the second approach uses a more direct encoding of bridge rule modifications that require all bridge rules of an existing MCS to be modified. The former approach is less intrusive, but does not allow perfect observation, hence for meta-reasoning on

MCS M and ...	Transformation		Size	Diagnosis notion	Complexity
deletion-parsimon. filter f	M_f	(Def. 4.16)	linear	$D^{\pm}_m(M_f, br_P)$	$\mathbf{D^P_i}$
filter f	M^f	(Def. 4.19)	linear	$D^{\pm}_m(M^f, br_P)$	$\mathbf{D^P_i}$
total preference order $\preceq$	$M^{pl\preceq}$	(Def. 4.22)	exponential	$D^{\pm}_m(M^{pl\preceq}, br_P, br_H)$	$\mathbf{D^P_i}$
preference order $\preceq$	$M^{\preceq}$	(Def. 4.23)	linear	$D^{\pm}_{m,t_{max}}(M^{\preceq}, br_P, br_H)$	$\mathbf{\Pi^P_i}$

Table 4.2: Overview of the presented approaches to select filtered and most-preferred diagnoses. For the given filters and preference orders (first column), the transformation approaches (second column) are proven to be correct. Size is the size of the resulting transformed MCS in terms of bridge rules of M; complexity is given wrt. context complexity $\mathbf{\Sigma^P_i}$ of the transformed MCS (complexity depends also on hardness of f and $\preceq$).

diagnoses additional guessing is necessary. The second approach allows perfect observation and requires no guessing, which is the reason why we focused on this approach for realising the selection of most-preferred diagnoses.

Both approaches also need some more involved notions of diagnosis, namely diagnoses where some bridge rules are protected and diagnoses where some bridge rules are considered of higher priority than the rest (i.e., some kind of lexicographic order on bridge rules). An analysis of the computational complexity of these notions was given, which showed that the notion of minimal diagnosis with protected bridge rules has the same complexity as the notion of minimal diagnosis. The notion of prioritised-minimal diagnosis also has the same complexity, but it is not sufficient to select most-preferred diagnoses. The notion to do so, called an mpm-diagnosis, has higher complexity than the notion of subset-minimal diagnosis, but we also showed that the problem of selecting most-preferred diagnoses is computationally as hard as an mpm-diagnosis. Hence, our approach is worst-case optimal from a complexity theoretic point of view.

In Table 4.2 we give an overview of the meta-reasoning techniques developed and their respective overall complexities given some preference order or filter. Note that the definitions of the presented transformations all state which "interface" has to be followed by the implementation of a filter or a preference order. We also give examples that show how Answer-Set Programming can be used to realise these interfaces in general. But of course, any formalism may be used to do so.

CHAPTER 5

Inconsistency in Managed Multi-Context Systems

5.1 Introduction

This chapter is dedicated to the incorporation of legacy solutions for local inconsistency management, i.e., inconsistency management that is tailored to specific knowledge-representation formalisms. For many such formalisms there already exist ways to deal with inconsistency, e.g. various kinds of belief revision operators for propositional logic, updates for logic programming, paraconsistent logics, etc. All of these methods address inconsistency in one specific formalism or a specific class of logics, hence in a Multi-Context System these ways to deal with inconsistency are tightly connected to certain instances of abstract logics, i.e., they are tied to contexts. Additionally, most of these methods of dealing with inconsistency assume to have a "global" perspective, i.e., they require full access to all knowledge that they are applied to. Therefore they must be connected directly with a context and its corresponding knowledge base, where they may assume this global perspective on the local context.

On the other hand, the approach must be very general to allows all kinds of ways to deal with inconsistency in a knowledge-representation formalism. In [32] we therefore proposed a new, more general form of MCS where such additional operations on knowledge bases can be freely defined; this is akin to the management functionality of database systems. We call the additional component *context manager* and the generalised systems *managed Multi-Context Systems* (mMCS). Each context manager may deal with inconsistency or apply other modifications to a context. Depending on applicable bridge rules, this manager can modify the knowledge base and even the semantics (i.e., the acceptability function) of the context. The premier use of context managers in case of this thesis is, however, that they allow to employ legacy techniques for inconsistency management. Since our results on inconsistency also hold for mMCS in general, we do not restrict mMCS here and instead introduce them in their full generality,

Our contributions in this chapter therefore are

(1) the concept of managed Multi-Context Systems, which allow bridge rules to not only add elements to the knowledge base of a context, but to trigger any operation on the knowledge base. For instance, rather than simply *adding* a formula ϕ, we may want to *revise* the KB with ϕ to avoid inconsistency in the context's belief set (cf. [105]); and, we may want to modify only certain parts of KB such that ϕ is entailed.

(2) a short survey on how the manager of a context can be used: for the full range of SQL operations on a relational database; for belief revision; for updates of logic programs; for capturing the framework of argumentation context systems.

(3) an investigation of the effects of local inconsistency management on overall consistency of the mMCS. We show that the reasons of inconsistency are always rooted in some cyclic information flow in an mMCS where the local consistency of each context is ensured by the context manager.

(4) we show that mMCS can capture MCS and vice versa by incorporating the functionality of context managers into the respective acceptability functions. The latter is also used to show that the computational complexity of deciding whether an mMCS is consistent is of the same complexity as deciding whether the corresponding MCS (with internalised management functionality) is consistent.

The outline of this chapter is as follows: we introduce the mMCS framework in Section 5.2, Section 5.3 discusses sample instances. The the aspects of inconsistency management in mMCS are addressed in Section 5.4, and complexity issues in Section 5.6.

5.2 Managed Multi-Context Systems

Multi-Context Systems allow us to increase the knowledge base of a context using information from other contexts, but not to operate on it in other ways. Such other operations might be: removal of information; revision with new information, or other complex operations like view-updates of databases; program updates of logic programs; modifications of argumentation frameworks, etc. Notably, these operations are realised by legacy systems, but the MCS framework can not cope with this functionality in a principled way.

To enable such functionality with a clear distinction between the knowledge base and additional operations on it, we introduce the managed Multi-Context Systems framework. The latter extends Multi-Context Systems such that contexts come with an additional managing function that evaluates the aforementioned operations, in analogy to the distinction of a database (DB) and a database management system (DBMS).

Example 5.1. *Consider a pharmaceutical company producing drugs. A drug effect database C_1 holds information about what is the remedy of an illness caused by certain bacteria, also treatments known to be ineffective are stored. To maximise efficiency, the company wants to know all kinds of illness that can be cured by their drugs. A public health RDF-triple store C_2 is queried on illness caused by bacteria for which C_1 already holds a remedy. Furthermore, probable influence of the drugs on other bacteria is derived which enables more focused clinical* *trials, i.e., only those effects are tested in the later trial where a likely effect was found. This is realised using an ontology about bacteria C_3 and a third party reasoner using a logic program C_4 to derive likely drug effects on other bacteria. To avoid that the third-party reasoner becomes inconsistent, the bridge rules of C_4 do not add information, instead they trigger an update of the employed logic program. To facilitate the information exchange, C_1 uses a view restricted to the necessary information and since information derived from C_4 is also returned through this view, it must be updateable. Furthermore, the third-party offers several available semantics for their reasoner, including stable model and well-founded semantics. The overall system in the end is the mMCS $M_{ph} = (C_1, C_2, C_3, C_4)$ given in Example 5.4 and depicted in Figure 5.1.*

To accommodate flexible semantics of contexts, we first extend the notion of abstract logic to one which has several semantics to choose from. This allows, e.g., that a logic program is evaluated using well-founded semantics instead of stable-model semantics based on input from other contexts, or switching from classical to paraconsistent semantics.

Definition 5.1. *A* logic suite $LS = (\mathbf{KB}_{LS}, \mathbf{BS}_{LS}, \boldsymbol{\mathcal{ACC}}_{LS})$ *consists of the set* $\mathbf{BS}_{LS}$ *of possible belief sets, the set* $\mathbf{KB}_{LS}$ *of well-formed knowledge-bases, and a nonempty set* $\boldsymbol{\mathcal{ACC}}_{LS}$ *of possible semantics of LS, i.e,* $\mathbf{ACC} \in \boldsymbol{\mathcal{ACC}}_{LS}$ *implies* $\mathbf{ACC} : \mathbf{KB}_{LS} \to 2^{\mathbf{BS}_{LS}}$.

For a logic suite LS, let $\Phi_{LS} = \{s \in kb \mid kb \in \mathbf{KB}_{LS}\}$ be the set of formulas occurring in its knowledge bases.

Example 5.2 (continued)**.** *C_1 is a relational database which contains the information that penicillin remedies pneumonia caused by the bacteria streptococcus pneumoniae and also that azithromycin remedies Legionair's disease caused by legionella pneumophila; effectiveness of both remedies is backed by clinical trials and therefore evident. Also, it contains information that penicillin is ineffective against legionella pneumophila. C_1 is based on the abstract logic $L_\Sigma^{DB} = (\mathbf{KB}_{DB}, \mathbf{BS}_{DB}, \mathbf{ACC}_{DB})$ of Example 2.7 with Σ suitably chosen to accommodate the following knowledge base kb_1 of C_1.*

$$\begin{aligned} kb_1 = \{&treatment(penicil, str\_pneu, pneu, evidence),\\ &treatment(azith, leg\_pneu, leg, evidence),\\ &ineffective(penicil, leg\_pneu)\} \end{aligned}$$

The logic suite LS_1 employed by C_1 is $LS_1 = (\mathbf{KB}_{DB}, \mathbf{BS}_{DB}, \{\mathbf{ACC}_{DB}\})$, i.e., it equals L_Σ^{DB} with its acceptability function $\mathbf{ACC}_{DB}$ being the only one of LS_1.

C_2 is an RDF-triple store containing information that streptococcus pneumoniae causes meningitis and legionella pneumonphila causes atypical pneumonia. Formally, the knowledge bases of C_2 is:

$$\begin{aligned} kb_2 = \{&str\_pneu\ rdf{:}causes\ men,\\ &leg\_pneu\ rdf{:}causes\ atyp\_pneu\}. \end{aligned}$$

We assume that the logic suite $LS_2 = (\mathbf{KB}_{RDF}, \mathbf{BS}_{RDF}, \{\mathbf{ACC}_{RDF}\})$ and its semantics are suitably chosen, e.g., based on [81] with $\mathbf{KB}_{RDF}$ and $\mathbf{BS}_{RDF}$ being a set of ground RDF *triples (i.e., RDF triples without blank nodes), and* $\mathbf{ACC}_{RDF}$ *being simple entailment, i.e.,* $\mathbf{ACC}_{RDF}(kb) = \{kb\}$*.*

The bacteria DL ontology C_3 *is based on the logic* $L_\Sigma^{\mathcal{ALC}} = (\mathbf{KB}, \mathbf{BS}, \mathbf{ACC})$ *(with* $\mathcal{A}$ *suitably chosen), but differently from* $L_\Sigma^{\mathcal{ALC}}$*, here we are not interested in atomic A-Box axioms, but in concept subsumption. Formally, the logic suite* $LS_3 = (\mathbf{KB}_3, \mathbf{BS}_3, \{\mathbf{ACC}_3\})$ *where* $\mathbf{KB}_3 = \mathbf{KB}$*,* $\mathbf{BS}_3$ *is the powerset of atomic concept subsumption, i.e.,* $\mathbf{BS}_3 = 2^Q$*,* $Q = \{C \sqsubseteq D \mid C, D \text{ are atomic concepts}\}$*, and* $\mathbf{ACC}_3(kb) = \{S\}$ *where* S *contains all concept subsumptions under* kb*. The knowledge base* kb_3 *of* C_3 *contains the information that streptococcus pneumoniae is a bacterium and legionella pneumophilae also is a bacterium;* kb_3 *is as follows:*

$$kb_3 = \{str\_pneu \sqsubseteq bact,\\ leg\_pneu \sqsubseteq bact\}.$$

C_4 *is a logic program accessing the ontology* C_3 *on bacteriological relations to deduce probable effects on related bacteria.* C_1 *also holds information on ineffective drugs which should be incorporated into* C_4 *by means of a logic program update. This allows one to deduce probable effects on bacteria except for cases where there is already negative evidence. The context* C_4 *employs a generalised logic program to derive further possible drug effects. The context uses a logic similar to* L_Σ^{asp} *of Example 2.5 where default-negated atoms may appear in the head of a rule.*

Formally, $LS_4 = (\mathbf{KB}_4, \mathbf{BS}_4, \{\mathbf{ACC}_{ASP}, \mathbf{ACC}_{WF}\})$ *where* $\mathbf{KB}_4$ *is the powerset of rules of form* $L \leftarrow L_1, \ldots, L_k$ *where* $L, L_1, \ldots, L_k$ *are literals over* Σ*,* $\mathbf{BS}_4$ *is the set of Herbrand interpretations over* Σ*,* $\mathbf{ACC}_{ASP}$ *is the semantics of stable models of generalised logic programs (cf. [2]), and* $\mathbf{ACC}_{WF}$ *is the well-founded semantics of logic programs (cf. [125]).*

Finally, the knowledge base kb_4 *of* C_4 *states that: if* A *and* B *are instances of the same concept* C*, and* X *is effective against* A*, then* X *is also effective against* B*;*

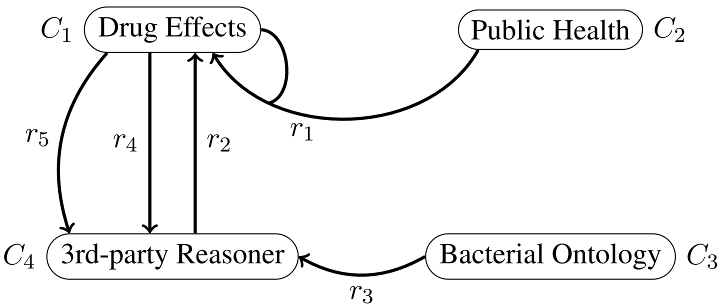
$$kb_4 = \{effective(X, B) \leftarrow effective(X, A), isa(A, C), isa(B, C).\}$$

Regarding the bridge rules of this system, we note that C_1 *requires view updates to be executed while* C_4 *simply imports some information from the ontology* C_3 *and uses information from* C_1 *to revise its program with. Hence,* C_4 *requires two types of operations executed on its applicable bridge rules, namely addition and revision.*

In MCS the head formulas in the bridge rules of a context are all treated the same, i.e., these formulas are added to the knowledge base of the context. Here, we want to extend bridge rules such that their head formulas may be treated in different ways, e.g., some the head formulas of some bridge rules are simply added to the knowledge base, while for some other bridge rules, the knowledge base is revised with their head formulas. This also enables bridge rules to "trigger" more than one operation. To distinguish these operations, we introduce the notion of a management base as follows.

Definition 5.2. *A* management base *is a set of operation names (briefly, operations)* OP*.*

Intuitively, a management base is the set of commands that can be executed on formulas, e.g., addition, revision with formulas. For a logic suite LS and a management base OP, let



r_1: $(1:\mathit{insert}(\mathit{treatment}(X,B,I,\mathit{likely}))) \leftarrow (1:\mathit{effective}\,(X,B))\,,$
$(2:B\;\mathit{rdf}\!:\!\mathit{causes}\;I)\,.$
r_2: $(1:\mathit{insert}(\mathit{effective}(X,B))) \leftarrow (4:\mathit{effective}\,(X,B))\,.$
r_3: $(4:\mathit{add}(\mathit{isa}(X,Y))) \leftarrow (3:\,(X \sqsubseteq Y))\,.$
r_4: $(4:\mathit{add}(\mathit{effective}(X,B))) \leftarrow (1:\mathit{effective}\,(X,B))\,.$
r_5: $(4:\mathit{upd}(\mathit{not\ effective}(X,B))) \leftarrow (1:\mathit{ineffective}\,(X,B))\,.$

$$kb_1 = \{str\_pneu\ \mathit{rdf}\!:\!causes\ men,\ leg\_pneu\ \mathit{rdf}\!:\!causes\ atyp\_pneu\}$$

$$kb_2 = \{blood\_marker, xray\_pneumonia\}$$

$$kb_3 = \{str\_pneu \sqsubseteq bact,\ leg\_pneu \sqsubseteq bact\,\}$$

$$kb_4 = \{\mathit{effective}(X,B) \leftarrow \mathit{effective}(X,A), \mathit{isa}(A,C), \mathit{isa}(B,C).\}$$

Figure 5.1: The mMCS M_{ph} from Examples 5.1–5.4 with contexts C_i, knowledge bases kb_i, and schematic bridge rules $r_1, \ldots, r_5$.

$\Phi_{LS}^{OP} = \{o(s) \mid o \in OP, s \in \Phi_{LS}\}$ be the set of operational statements that can be built from OP and Φ_{LS}.

The semantics of such statements is given by a management function. A management function maps a set of operational statements and a knowledge base to pairs of a modified knowledge base and a semantics. It allows to not only add formulas to a context, but to specify any desired operations to be applied on formulas and a context.

Definition 5.3. *A* management function *over a logic suite LS and a management base OP is a function $mng : 2^{\Phi_{LS}^{OP}} \times \mathbf{KB}_{LS} \to 2^{(\mathbf{KB}_{LS} \times \boldsymbol{\mathcal{ACC}}_{LS})} \setminus \{\emptyset\}$.*

Note that the management function returns a set of pairs of modified knowledge-base and selected semantics. This allows for a management function to offer more than one result which may be used, e.g., in model-based revision where each resulting knowledge-base corresponds to a model of the revised formula.

The bridge rules of a context C_i of an mMCS are now of form (2.1) as for MCS, but with the head expression s being an operational statement for the management function mng_i of C_i. All notions regarding bridge rules carry over directly from MCS to mMCS. Hence, for OP_i being the management base of C_i, it holds for every $r \in br_i$ that $\varphi(r) \in \Phi^{OP_i}_{LS_i}$. The notion of an L^k bridge rule in mMCS is defined analogous to MCS, with the only difference that L^k bridge rules are now defined over a sequence $L = (LS_1, \ldots, LS_n)$ of logic suites instead of a sequence $L = (L_1, \ldots, L_n)$ of abstract logics.

Example 5.3 (continued)**.** *The data of C_1 is offered to C_4 through a view "$effective$", which contains only the information about which drug effects what bacteria, i.e., $(X, B) \in effective$ iff $(X, B, \_, evidence) \in treatment$. To insert into that view, C_1 uses an operation "$insert$", i.e., $OP_1 = \{insert\}$ and the management function mng_1 is defined such that tuples inserted into the view $effective$ are transformed to $treatment$ tuples. Formally,*

$$mng_1(O, kb_1) = \{(kb'_1 \cup N \cup V, \mathbf{ACC}_1)\}$$

$$\begin{aligned} \textit{where } kb'_1 &= kb_1 \cup \{effective(X, B) \mid treatment(X, \_, B, evidence) \in kb_1\} \\ N &= \{treatment(X, B, I, E) \mid insert(treatment(X, B, I, E)) \in O\} \\ V &= \{treatment(X, B, I, estimate) \mid insert(effective(X, B)) \in O, \\ &\qquad\qquad treatment(\_, B, I, \_) \in kb_1 \cup N\} \end{aligned}$$

Notice that kb'_1 serves to materialise the view regarding $effective$.

Since C_2 and C_3 are not intended to have any bridge rules, their management bases are irrelevant, i.e., we pick $OP_2 = OP_3 = \emptyset$ and their management functions do not modify their original respective knowledge base. Hence mng_2 and mng_3 are any functions such that $mng_2(\emptyset, kb) = \{(kb, \mathbf{ACC}_{RDF})\}$ and $mng_3(\emptyset, kb) = \{(kb, \mathbf{ACC})\}$ hold.

For C_4, the operations are $OP_4 = \{upd, add\}$ and mng_4 adds all formulas in add as facts and the resulting program is updated according to formulas in upd by the method given in [2], which avoids inference of conflicting information (see Example 5.5 for details). Thus, mng_4 always selects as acceptability function $\mathbf{ACC}_{ASP}$ the stable model semantics of dynamic logic programs.

The context C_1 contains two types of bridge rules: the first connect existing information about effects on bacteria with information from C_2 about illness caused by the same bacteria. The incorporated information is further marked as only being likely, but not clinically tested. The second type of bridge rules incorporates probable drug effects from C_4.

$$\begin{aligned} br_1 = \{\ &(1\!:insert(treatment(X, B, I, likely))) \leftarrow (1\!:effective\,(X, B))\,, (2\!:B\ rdf\!:\!causes\ I)\,. \\ &(1\!:insert(effective(X, B))) \leftarrow (4\!:effective\,(X, B))\,.\} \end{aligned}$$

Contexts C_2 and C_3 have no bridge rules, only C_4 gets bridge rules

$$\begin{aligned} br_4 = \{ \quad &(4\!:add(isa(X, Y))) \leftarrow (3\!:(X \sqsubseteq Y))\,. \\ &(4\!:add(effective(X, B))) \leftarrow (1\!:effective\,(X, B))\,. \\ &(4\!:upd(not\ effective(X, B))) \leftarrow (1\!:ineffective\,(X, B))\,.\}, \end{aligned}$$

which, from top to bottom, import ontological information from C_3, import drug effects from C_1, and update the resulting logic program with ineffectiveness information from C_1. The updating *ensures that there is no contradicting effectiveness derived by C_4 in such a way that known ineffectiveness takes precedence over derived information.*

Here we use for readability and succinctness schematic bridge rules with variables (upper case letters and _) which range over associated sets of constants; they stand for all respective instances, which are obtainable by value substitution.

A managed MCS (mMCS) is then an MCS where each context additionally comes with a management function and bridge rule heads are operational statements.

Definition 5.4. *A* managed Multi-Context System M *is a collection* $M = (C_1, \ldots, C_n)$ *of managed contexts where, for* $1 \leq i \leq n$*, each* managed context C_i *is a quintuple* $C_i = (LS_i, kb_i, br_i, OP_i, mng_i)$ *such that*

- $LS_i = (\mathbf{BS}_{LS_i}, \mathbf{KB}_{LS_i}, \boldsymbol{\mathcal{ACC}}_{LS_i})$ *is a logic suite,*
- $kb_i \in \mathbf{KB}_{LS_i}$ *is a knowledge base,*
- br_i *is a set of* L^i *bridge rules over* $L = (LS_1, \ldots, LS_n)$*,*
- OP_i *is a management base, and*
- mng_i *is a management function over* LS_i *and* OP_i*.*

As for ordinary MCS, a belief state $S = (S_1, \ldots, S_n)$ of M is a belief set for every context, i.e., $S_i \in \mathbf{BS}_{LS_i}$. Again, by $app_i(S, M) = \{\varphi(r) \mid r \in br_i(M), S \models r\}$ we denote the set of applicable heads of bridge rules, which is a set of operational statements.

The semantics of mMCS is also defined in terms of equilibria as follows.

Definition 5.5. *Let* $M = (C_1, \ldots, C_n)$ *be an mMCS. A belief state* $S = (S_1, \ldots, S_n)$ *is an equilibrium of* M *iff for every* $1 \leq i \leq n$ *there exists some* $(kb'_i, \mathbf{ACC}_{LS_i}) \in mng_i(app_i(S, M), kb_i)$ *such that* $S_i \in \mathbf{ACC}_{LS_i}(kb'_i)$*.*

The equilibrium semantics of ordinary MCS contains two steps: applicability of bridge rules and acceptability of belief sets under the resulting knowledge base. The equilibrium semantics of mMCS adds between those two steps another one for managing the context.

Example 5.4. *The mMCS* $M_{ph} = (C_1, C_2, C_3, C_4)$ *of Example 5.3 has one equilibrium* $S = (S_1, S_2, S_3, S_4)$ *where, omitting for brevity atoms of the form* $not\ a$ *in* S_4*, it holds that:*

$$
\begin{aligned}
S_1 = \{\, & \mathit{treatment}(\mathit{penicil}, \mathit{str\_pneu}, \mathit{men}, \mathit{likely}),\\
& \mathit{treatment}(\mathit{azith}, \mathit{leg\_pneu}, \mathit{atyp\_pneu}, \mathit{likely}),\\
& \mathit{treatment}(\mathit{azith}, \mathit{str\_pneu}, \mathit{pneu}, \mathit{estimate}),\\
& \mathit{treatment}(\mathit{azith}, \mathit{str\_pneu}, \mathit{men}, \mathit{estimate}),\\
& \mathit{effective}(\mathit{azith}, \mathit{str\_pneu}), \mathit{effective}(\mathit{azith}, \mathit{leg\_pneu}),\\
& \mathit{effective}(\mathit{penicil}, \mathit{str\_pneu})\} \cup kb_1,
\end{aligned}
$$

$$
\begin{aligned}
S_2 &= kb_2, \\
S_3 &= \{\, str\_pneu \sqsubseteq bact, \\
&\qquad leg\_pneu \sqsubseteq bact\}, \textit{ and} \\
S_4 &= \{\, \mathit{effective}(penicil, str\_pneu), \\
&\qquad \mathit{effective}(azith, str\_pneu), \\
&\qquad \mathit{effective}(azith, leg\_pneu), \\
&\qquad isa(str\_pneu, bact), \\
&\qquad isa(leg\_pneu, bact)\}.
\end{aligned}
$$

We extend some notions from MCS to mMCS: given an mMCS M, the set of all its bridge rules is denoted by $br(M)$, $cf(r)$ is the condition-free version of r, i.e., $cf(r)$ is $head(r) \leftarrow .$, $\mathrm{EQ}(M)$ denotes the set of all equilibria of M, and for any set R of bridge rules is $cf(R) = \{cf(r) \mid r \in R\}$. Furthermore, $M[R]$ denotes the mMCS M where all bridge rules are replaced by those bridge rules in R (assuming that R is compatible with M).

5.3 Sample Instantiations

We consider instantiations of our framework, first discussing relational databases, logic programs, and belief revision. Second, we capture argumentation context systems by mMCS.

Relational Databases. For relational databases, our running example already shows how a management function is used to realise view-updates. Many other operations on databases may be realised using managed contexts. In fact, e.g., the SQL language, Σ_{SQL}, can be accommodated: a context whose management base is built upon Σ_{SQL} and a management function mng_{SQL} which realises the SQL semantics. This allows us to use SQL in an mMCS. Observe that the implementation of mng_{SQL} is rather trivial, as existing implementations of SQL can be used via suitable interfaces, e.g., MySQL, Oracle DB, etc. In our running example, the respective view statement is:

```
CREATE VIEW eff AS
       SELECT drug, bacteria FROM treat
       WHERE credibility = evd;
```

To realise ordered sequences of SQL statements, one may use time stamps in bridge rules handled by mng_{SQL}.

Belief Revision. Change of logical theories and knowledge bases is a long-standing area in logic and AI. Central operations on beliefs are $expansion$, $contraction$, $revision$ and $update$ (see [105] for an excellent survey).

Let L be a logic with the set $\mathcal{L}$ of formulas and semantics $\mathbf{ACC}_L$, and let $rev : 2^{\mathcal{L}} \times \mathcal{L} \rightarrow 2^{\mathcal{L}}$ be a revision operator for theories in L. We may define a management function mng_{rev} for the management base $\{revise\}$, e.g., as follows:

$$mng_{rev}(O, kb) = \{(rev(kb, \{\phi_1 \wedge \ldots \wedge \phi_n \mid revise(\phi_i) \in O\}), \mathbf{ACC}_L)\}.$$

Here, multiple revisions ($n \geq 2$) are handled by conjunction; other realisations (e.g., iteration) exist.

Logic Programming. Various extensions of logic programs have been proposed, e.g., updates of logic programs [2] which we use in our running example, debugging-support [26], or meta-reasoning support. Many of them are realised using meta-programming (viewing the program to update as data), i.e., they transform a logic program P_e and additional input I, into a logic program P_t, such that solutions to the problem given by P_e and I are obtained from P_t (without altering the semantics). In the mMCS framework, we can achieve this directly using a management function mng such that, for a program P_e and operational statements O encoding I, $mng(O, P_e) = \{(P_t, ACC_{LP})\}$ where P_t is assembled from P_e and O, and ACC_{LP} is the employed semantics of logic programs.

Example 5.5. *Suppose C_4 of the mMCS in Example 5.4 uses the update semantics of [2], i.e., the semantics of upd is given by the respective program transformation. For the operational statement $upd(not\ effective(penicil, leg\_pneu))$ which is applicable in the belief state S of Example 5.4, the relevant rules are*

$$not\ effective(penicil, leg\_pneu) \leftarrow . \tag{5.1}$$

$$effective(penicil, leg\_pneu) \leftarrow effective(penicil, bact), isa(str\_pneu, bact), isa(leg\_pneu, bact). \tag{5.2}$$

The ground instance (5.2) *is rejected (not contained in the transformed program), intuitively because it is in conflict with the more recent information represented by* (5.1)*. Therefore, the stable model does not contain $effective(penicil, leg\_pneu)$.*

The ability of the management function to choose among different semantics allows one to flexibly select a suitable logic program semantics depending on the belief state. E.g., one may enforce for a program in context C_i that paraconsistent semantics as in [115] is used if it has an inconsistent belief set, and answer-set semantics otherwise. If the context employs answer-set semantics as originally defined in [72], where an inconsistent logic program has the single answer-set consisting of all literals, then a self-referential bridge rule may be used to detect whether the program is inconsistent. Hence the management function need not even be able to detect that answer-set semantics yields inconsistency, since this can be done using a bridge rule whose operation in the head causes the management function to select one of the available semantics. It is achieved by self-referential bridge rules $semantics(parAS) \leftarrow (i:\bot).$ and $semantics(AS) \leftarrow not(i:\bot).$, where $\bot$ encodes inconsistency, $OP_i = \{semantics\}$, and

$$mng_i(kb, O) = \begin{cases} \{(kb, \mathbf{ACC}_{parAS})\} & \text{if } semantics(parAS) \in O \\ \{(kb, \mathbf{ACC}_{AS})\} & \text{otherwise.} \end{cases}$$

Argumentation Context Systems. Argumentation context systems (ACS) [30] are a homogeneous framework for distributed, abstract group argumentation where all reasoning components are Dung argumentation frameworks (AFs) [47]. ACS are similar to mMCS in that they utilize bridge rules for information exchange about arguments and each component of an ACS is equipped with a mediator similar to a context manager. Even more so since bridge rules not only extend context AFs, they may invalidate arguments or attack relations, they may select a semantics, and they provide ways of resolving inconsistent information resulting the applicability of bridge rules whose head formulas possibly contradict each other.

We now recall AFs and ACS as given in [30]. An *argumentation framework* (AF) is a pair $\mathcal{A} = (A, attacks)$ where A is a set of *arguments* and $attacks$ is a binary relation on A. An argument $a \in A$ is *acceptable* with respect to a set $E \subseteq A$ if for each $b \in A$ with $attacks(b, a)$ exists some $b' \in E$ with $attacks(b', b)$, i.e., a is acceptable wrt. E if every attacker in A is attcked by some argument $b' \in E$. A set $E \subseteq A$ is *conflict-free* if there are no arguments $a, b \in E$ with $attacks(a, b)$; E is *admissible* if it is conflict-free and each argument $a \in E$ is acceptable wrt. E.

There is a number of different semantics for AFs, where the basic ones are given by *preferred extensions*, *stable extensions* and *grounded extensions*. A set $E \subseteq A$ is a preferred extensions of $\mathcal{A} = (A, attacks)$ if E is a maximal (wrt. $\subseteq$) admissible set of $\mathcal{A}$; $E \subseteq A$ is a stable extensions of $\mathcal{A}$ if E is conflict-free and every argument outside E is attacked by an argument in E, i.e., for all $b \in A \setminus E$ exists $b' \in E$ such that $attacks(b', b)$ holds; finally, $E \subseteq A$ is a grounded extensions if it is the least fixpoint of the operator $F_{\mathcal{A}} : 2^A \rightarrow 2^A$ where $F_{\mathcal{A}}(E) = \{a \in A \mid a \text{ is acceptable wrt. } E\}$. A logic suite to capture argumentation frameworks and the above semantics is $L^{AF}_{\Sigma} = (\mathbf{KB}_{AF}, \mathbf{BS}_{AF}, \{\mathbf{ACC}_{pref}, \mathbf{ACC}_{stab}, \mathbf{ACC}_{gr}\})$ where $\mathbf{KB}_{AF}$ is the powerset over $\{attacks(a, b) \mid a, b \in \Sigma\}$, i.e., $\mathbf{KB}_{AF}$ consists of all possible attack relations over Σ, $\mathbf{BS}_{AF} = 2^{\Sigma}$ is the set of possible extensions, and $\mathbf{ACC}_s(kb) = \{E \in \mathbf{BS}_{AF} \mid E \text{ is an } s\text{-extension of } kb\}$ where $s \in \{pref, stab, grnd\}$ and an s-extension is the respective preferred, stable, or ground extension.

ACS allow an AF to be modified according to information from other AFs. These modifications are given by so-called *context expressions*, where for a set A of arguments and a set of values V the following expressions are possible ($a, b \in A$, $v, v' \in V$): $\mathbf{arg}(a)$, $\overline{\mathbf{arg}}(a)$, $\mathbf{att}(a, b)$, $\overline{\mathbf{att}}(a, b)$, $a > b$, $\mathbf{val}(a, v)$, $v > v'$, $\mathbf{mode}(r)$, $\mathbf{sem}(s)$ where $r \in \{skep, cred\}$ is a reasoning mode and $s \in \{pref, stab, grnd\}$ is a semantics. Notice that the usage of values V allows to model value-based argumentation-frameworks.

A set CE of context expressions induces a *preference order* $>_{CE}$ defined as the smallest transitive relation such that $a >_{CE} b$ holds if either $a > b \in CE$ or $\mathbf{val}(a, v_1) \in CE$, $\mathbf{val}(b, v_2) \in CE$, and (v_1, v_2) is in the transitive closure of $\{(v, v') \mid v > v' \in CE\}$. Based on $>_{CE}$ and the contents of CE one can decide whether a set of context expressions CE is *consistent*; for more details see [30].

Given an AF $\mathcal{A} = (A, attacks)$ and a set of context expressions CE, the CE-modification of $\mathcal{A}$ is the argumentation framework $\mathcal{A}^{CE} = (A^{CE}, attacks^{CE})$ where $A^{CE} = A \cup \{\mathbf{def}\}$ such that $\mathbf{def} \notin A$, and $attacks^{CE}$ is the smallest relation satisfying

1. if $\mathbf{att}(a, b) \in CE$ then $(a, b) \in attacks^{CE}$,
2. if $(a, b) \in attacks$, $\mathbf{att}(a, b) \notin CE$, and $b \not>_{CE} a$, then $(a, b) \in attacks^{CE}$,

3. if $\overline{\mathbf{arg}}(a) \in CE$ or $(\mathbf{arg}(b) \in CE \wedge (a,b) \in \mathit{attacks}^{CE})$ then $(\mathbf{def}, a) \in \mathit{attacks}^{CE}$.

The CE-modification basically applies all modifications as given in CE respecting preferences and removal of arguments, where the latter is achieved by the addition of a new argument $\mathbf{def}$ that invalidates all "removed" arguments by attacking them.

To express information exchange between AFs, bridge rules similar to those of MCS are used. In the following, we call such a bridge rule an *ACS-rule*, whose general form is

$$s \leftarrow p_1, \ldots, p_j, \mathbf{not}\ p_{j+1}, \ldots, \mathbf{not}\ p_m.$$

where s is a context expression and $p_1, \ldots, p_m$ are arguments of a specific parent AF. Note that an ACS-rule only considers arguments from one AF, though multiple rules may consider several AFs.

An *argumentation context systems* (ACS) is a sequence $\mathcal{F} = (\mathcal{M}_1, \ldots, \mathcal{M}_n)$ of *modules* $\mathcal{M}_i = (\mathcal{A}_i, \mathit{Med}_i)$, $1 \leq i \leq n$ where $\mathcal{A}_i$ is an argumentation framework and Med_i is a mediator over the AFs $\mathcal{A}_1, \ldots, \mathcal{A}_n$. A *mediator* $\mathit{Med}_i = (E_i, R_1, \ldots, R_{i-1}, R_{i+1}, \ldots, R_k, \mathit{choice})$ is a tuple comprised of a set E_i of context expressions, a set of ACS-rules R_j, $j \neq i$, for every other AF of the ACS, and an inconsistency-handling method choice. There are four inconsistency-handling methods for choice which are based on [27]: some details of this inconsistency handling are given in [30]; for this work it is sufficient to know that each of these methods allows to select from a sequence $F = (F_1, \ldots, F_n)$ of formulas a consistent subset F' of $F_1 \cup \ldots \cup F_n$.

The semantics of an ACS is given in terms of *acceptable states*, where a state $\mathcal{S}$ assigns each module $\mathcal{M}_i = (\mathcal{A}_i, \mathit{Med}_i)$ a set of acceptable arguments $S_i \subseteq A_i$ of $\mathcal{A}_i = (A_i, \mathit{attacks}_i)$ and a set CE_i of context expressions for $\mathcal{A}_i$. A state is acceptable, if each S_i is an acceptable CE_i-extension for $\mathcal{A}_i$ and each CE_i is an acceptable context for Med_i with respect to all A_j ($1 \leq j \leq n$). We first state what makes S_i an acceptable CE_i-extension for $\mathcal{A}_i$ and then we state what makes CE_i an acceptable set of context expressions for Med_i.

First, given a consistent set of context expressions CE_i, a set $S_i \in A_i$ is an acceptable CE_i-extension either if $\mathbf{mode}(\mathit{cred}) \in CE_i$ and $S \cup \{\mathbf{def}\}$ is an s-extensions of $\mathcal{A}_i^{CE_i}$, or if $\mathbf{mode}(\mathit{skep}) \in CE_i$ and $S \cup \{\mathbf{def}\}$ is the intersection of all s-extensions of $\mathcal{A}_i^{CE_i}$, where $\mathbf{sem}(s) \in CE_i$ holds. Second, CE_i is an acceptable set of context expressions for the mediator $\mathit{Med}_i = (E_i, R_1, \ldots, R_n, \mathit{choice})$ wrt. a state $\mathcal{S} = ((S_1, CE_1), \ldots, (S_n, CE_n))$ if CE_i is a choice-preferred (hence consistent) set of context expressions for the sequence $F = (E_1, R_1(S_1), \ldots, R_n(S_n))$ where for all $1 \leq i \leq n$, $R_i(S_i)$ is the set of heads of applicable ACS-rules, i.e.,

$$R_i(S_i) = \big\{ h \mid h \leftarrow p_1, \ldots, p_j, \mathbf{not}\ p_{j+1}, \ldots, \mathbf{not}\ p_m. \in R_i, \\ \{p_1, \ldots, p_j\} \subseteq S_i, \{p_{j+1}, \ldots, p_m\} \cap S_i = \emptyset \big\}.$$

Note that different from MCS and mMCS, the bridge rules of an ACS have no context identifier but the context they refer to is implicit, since each module has one set of bridge rules for any other module.

We may capture an ACS $\mathcal{F} = (\mathcal{M}_1, \ldots, \mathcal{M}_1)$ by an mMCS $M^{\mathcal{F}} = (C_1, \ldots, C_n)$ based on logic suites L_{Σ}^{AF} and context managers that simulate the mediators. For every context expression of the ACS there exists a corresponding operation name and the bridge rules of $M^{\mathcal{F}}$ simulate the ACS-rules of $\mathcal{F}$. A one-to-one correspondence between equilibria (and their witnessing knowledge-bases) of $M^{\mathcal{F}}$ and the acceptable states of $\mathcal{F}$ can be shown.

Definition 5.6. *Given an ACS $\mathcal{F} = (\mathcal{M}_1, \ldots, \mathcal{M}_n)$ with modules $\mathcal{M}_i = (\mathcal{A}_i, Med_i)$, AFs $\mathcal{A}_i = (A_i, attacks_i)$, and mediators $Med_i = (E_i, R_{i,1}, \ldots, R_{i,n}, choice_i)$ for every $1 \leq i \leq n$, the* corresponding mMCS *$M^{\mathcal{F}} = (C_1, \ldots, C_n)$ is as follows.*

For $1 \leq i \leq n$, the context $C_i = (LS_i, kb_i, br_i, OP_i, mng_i)$ is such that

- *the logic suite is $LS_i = L^{AF}_{\Sigma} = (\mathbf{KB}_i, \mathbf{BS}_i, \boldsymbol{\mathcal{ACC}}_i)$ with $\Sigma = A_i \cup \{\mathbf{def}\}$,*
- *the knowledge base $kb_i = \{(a, a') \mid attacks(a, a') \in attacks_i\}$ contains the attack relation from the AF $\mathcal{A}_i$,*
- *the management base OP_i contains every context expression of CE_i indexed from 1 to n to identify the AF where the expression originates from, i.e, $OP_i = \{s_j \mid s \in CE_i, 1 \leq j \leq n\}$.*

The set of bridge rules br_i of context C_i contains for every ACS-rule $r \in R_{i,\ell}$, $1 \leq \ell \leq n$, of form (5.3) *the following bridge rule:*

$$(i : s_j(attacks(\mathbf{def}, \mathbf{def}))) \leftarrow (j : p_1), \ldots, (j : p_j), \mathbf{not}\ (j : p_{j+1}), \ldots, \mathbf{not}\ (j : p_m).$$

where $s_j \in OP_i$ is the operational name corresponding to the context expression s indexed by the context identifier j where the context expression originates from; also note that the knowledge-base formula $attacks(\mathbf{def}, \mathbf{def})$ serves the sole purpose of turning the context expression s into an operational statement. We design the management function such that the attack relation is ignored.

In slight abuse of notation, we denote by $R_j(O)$ the set of context expressions originating from j, with $1 \leq j \leq n$, that occur in a set O of operational statements, i.e., $R_j(O) = \{s \mid s_j(attacks(\mathbf{def}, \mathbf{def})) \in O\}$. Note that $R_j(O)$ actually ignores the knowledge-base formula and only yields operation names. Finally, the management function mng_i is such that for every set O of operational statements and every $kb \in \mathbf{KB}_i$ we have that

$$\begin{aligned} mng_i(O, kb) = \big\{(kb', \mathbf{ACC}_{s,m}) \mid\ & CE \textit{ is a } choice_i\textit{-preferred set of context expressions for} \\ & \mathcal{F} = (E_i, R_1(O), \ldots, R_n(O)), \\ & \mathbf{sem}(s) \in CE, \mathbf{mode}(m) \in CE, \mathbf{ACC}_{s,m} \in \boldsymbol{\mathcal{ACC}}_i \\ & kb' = \{attacks(a, b) \mid (\mathcal{A}_i)^{CE} = (A_i, att), (a, b) \in att\}\big\}. \end{aligned}$$

Given an ACS $\mathcal{F} = (\mathcal{M}_1, \ldots, \mathcal{M}_n)$ and its corresponding mMCS $M^{\mathcal{F}}$, we observe that for any equilibrium $(S_1, \ldots, S_n) \in \text{EQ}(M^{\mathcal{F}})$ there exist sets $CE_1, \ldots, CE_n$ of context expressions such that $((S_1, CE_1), \ldots, (S_n, CE_n))$ is an acceptable state of $\mathcal{F}$ and for any acceptable state $((S_1, CE_1), \ldots, (S_n, CE_n))$ of $\mathcal{F}$ it holds that $(S_1, \ldots, S_n)$ is an equilibrium of $M^{\mathcal{F}}$. In the realisation here, there is no direct one-to-one correspondence, because the context expressions $CE_1, \ldots, CE_n$ are implicitly kept inside the management functions of $M^{\mathcal{F}}$.

Note that it is also possible to shift the context expressions and corresponding modifications of an AF from the management function to the logic suite of the context in order to obtain the context expressions of acceptable states of the ACS. A knowledge base of a context then not only represents an AF $\mathcal{A}$ but also a set CE of context expressions; furthermore, the acceptability function $\mathbf{ACC}$ accepts a set of arguments A, if A is acceptable for $\mathcal{A}^{CE}$ and the respective reasoning mode and semantics that is specified in CE. By that, the choosing of a semantics, however, also is shifted from the management function to the context. We therefore feel that the presented realisation of ACS in mMCS by Definition 5.6 is more appealing.

5.4 Inconsistency Management

Different forms of inconsistency can arise in mMCS:

1. *Nonexistence of equilibria*: in the previous chapters, we said that an MCS M is inconsistent if $\text{EQ}(M) = \emptyset$, i.e., these chapters deal with nonexistence of equilibria in MCS, the underlying notions and many results can be directly extended to mMCS.

2. *Local inconsistency:* even if equilibria exist, they may contain inconsistent belief sets. This presupposes an adequate notion of consistency (for belief sets and sets of formulas). In most context logics such a notion exists or is easily defined.

3. *Operator inconsistency:* the operations in the heads of applicable bridge rules are conflicting, e.g., operations like $add(p)$ and $delete(p)$, or $revise(p)$ and $revise(\neg p)$, which might require the management function to yield knowledge bases that are consistent and at the same time contain p and contain $\neg p$.

Handling inconsistencies of type 2 and 3 is one of the motivations that led to the development of mMCS. For type 1 inconsistencies the techniques of Chapter 3 and from [52, 54] can be adapted easily.

Local Consistency

Local consistency requires an adequate notion of consistency for every employed logic suite. We give some examples of such notions, which may be used to decide whether a knowledge-base or a belief set of a context is (locally) consistent.

- Classical propositional logic using the abstract logic L^{pl}_{Σ} or any logic suite based on the same set $\mathbf{KB}$ of knowledge bases and belief sets $\mathbf{BS}$: a belief set $bs \in \mathbf{BS}$ is inconsistent, if there exists a formula ψ such that $\psi \in bs$ and $\neg\psi \in bs$ both hold; a knowledge base $kb \in \mathbf{KB}$ is consistent if every $bs \in \mathbf{ACC}(kb)$ is consistent, where $\mathbf{ACC} \in \mathcal{ACC}$ is any acceptability function employed by the logic suite.

  Note that for many logics, deciding whether a given knowledge base kb is consistent effectively amounts to checking whether all belief sets acceptable under kb contain both a literal and its negation.

- Description logics based on the abstract logic $L_{\Sigma}^{\mathcal{ALC}}$ or any logic suite using the same set $\mathbf{KB}$ of knowledge bases and belief sets $\mathbf{BS}$: a knowledge base $kb \in \mathbf{KB}$ (which contains A- and T-Box axioms) is consistent if there exists an interpretation that is both a model of all A-Box axioms of kb and of all T-Box axioms of kb (cf. [4]). Provided that the belief sets of $L_{\Sigma}^{\mathcal{ALC}}$ contain only atomic A-Box axioms (without negation), any belief set of $L_{\Sigma}^{\mathcal{ALC}}$ is consistent.

  Note that the logic suite employed in the mMCS of Example 5.4 uses belief sets $bs \in \mathbf{BS}$ to represent concept subsumption and $kb \in \mathbf{KB}$ may contain only T-Box axioms. Here, one may require for kb being considered consistent that there exists a model I of kb such that I satisfies all concepts occurring in kb. For $bs \in \mathbf{BS}$ that is acceptable for kb under the given semantics, i.e., $\mathbf{ACC}(kb) = \{bs\}$, one may require for bs being consistent that kb is consistent, i.e., it satisfies all occurring concepts.

- Answer-set programs based on $L_{\Sigma}^{asp} = (\mathbf{KB}, \mathbf{BS}, \mathbf{ACC})$: a knowledge base $kb \in \mathbf{KB}$ is consistent, if there exists an answer set of kb, i.e., kb is consistent if $\mathbf{ACC}(kb) \neq \emptyset$. A belief set $bs \in \mathbf{BS}$ always is consistent.

  Note that following [72], an inconsistent logic program has exactly one answer-set, namely the set $\{a, \neg a \mid a \in \Sigma\}$. If this definition of answer set is used for an abstract logic, then there previous set corresponds to the only inconsistent belief set and kb is inconsistent, if the aforementioned set is its single answer-set.

We now demonstrate how local consistency can be achieved by using adequate managers, given suitable notions of consistency as exemplified above.

Definition 5.7. *We call a management function mng* local consistency (lc-) ensuring[1] *if, for each set O of operational statements and each KB kb, in every pair $(kb', \mathbf{ACC}) \in mng(O, kb)$ the KB kb' is consistent. Furthermore, an mMCS M is* locally consistent *if in each equilibrium $S = (S_1, \ldots, S_n)$ of M, all S_i are consistent belief sets.*

In the remainder of this section, we assume that all acceptability functions of a logic suite reasonably fit the consistency notion in the sense that, if an acceptability function is applied to a consistent knowledge base, then all accepted belief sets are also consistent. Formally, given a logic suite $LS = (\mathbf{KB}, \mathbf{BS}, \mathcal{ACC})$, we assume the following holds:

> for all $\mathbf{ACC} \in \mathcal{ACC}$, if $kb \in \mathbf{KB}$ is consistent, then every $bs \in \mathbf{BS}$ with $bs \in \mathbf{ACC}(kb)$ is consistent.

Proposition 5.1. *Let M be an mMCS such that all management functions are lc-ensuring. Then M is locally consistent.*

Proof. Let $M = (C_1, \ldots, C_n)$ be an mMCS such that each for every context C_i its management function mng_i is lc-ensuring. Towards contradiction, assume that M is not locally consistent, i.e., there exists an equilibrium $S = (S_1, \ldots, S_n)$ such that for some belief set S_j with $1 \leq j \leq n$ it

[1] In [32] this property is called *local consistency (lc-) preserving*. We call this property lc-ensuring here to better account for the fact that the original knowledge-base may be inconsistent.

holds that S_j is not consistent. Since S is an equilibrium, it holds that $S_j \in \mathbf{ACC}(kb')$ for some $(\mathbf{ACC}, kb') \in mng_j(app_j(S, M), kb_j)$. Since mng_i is lc-ensuring, it holds that kb' is consistent. Since acceptability functions reasonably fit the consistency notion and since $S_j \in \mathbf{ACC}(kb')$ holds, it follows that S_j is consistent; this contradicts that S_j is not consistent. Therefore, no such equilibrium S exists and it holds that M is locally consistent. □

How to define lc-ensuring managers? To simplify matters we assume all contexts are based on propositional logic with classical consistency and semantics, given by $\mathbf{ACC}_{pl}$, and consider a single operator add with the obvious meaning. We proceed by: (1) selecting a base revision operator rev satisfying consistency preservation (revising a propositional KB with a consistent formula always results in a consistent KB); (2) picking maximal consistent subsets of the formulas to be added. Let $F_O = \{p \mid add(p) \in O\}$ and let $MC(F_O)$ be the set of maximal consistent subsets of F_O. Now define

$$mng(O, kb) = \{(rev\,(kb, \textstyle\bigwedge F)\,, \mathbf{ACC}_{pl}) \mid F \in MC(F_O)\}\,.$$

This management function is obviously lc-ensuring. Further refinements, e.g., based on additional preferences among bridge rules, are straightforward.

Global Consistency

We can directly extend the notions of Chapter 3 to mMCS to obtain diagnoses and explanations. A *diagnosis* of an inconsistent mMCS M is a pair (D_1, D_2) of sets of bridge rules $D_1, D_2 \subseteq br(M)$ such that $M[D_1, D_2] = M[br(M) \setminus D_1 \cup cf(D_2)]$ is consistent, i.e., $M[D_1, D_2] \not\models \bot$ which is equivalent to $\textsc{Eq}(M[D_1, D_2]) \neq \emptyset$. We denote by $D^{\pm}(M)$ the set of all such diagnoses and by $D^{\pm}_m(M)$ the set of all pointwise subset-minimal diagnoses. Observe that the definition of a diagnosis of an mMCS is the same as of a diagnosis of an MCS. This is because the necessary notions of MCS directly carry over to mMCS.

Explanations of inconsistency in mMCS are also defined in exactly the same way as for MCS. Formally, given an mMCS M, an *explanation* of M is a pair (E_1, E_2) of sets $E_1, E_2 \subseteq br(M)$ such that for all (R_1, R_2) with $E_1 \subseteq R_1 \subseteq br(M)$ and $R_2 \subseteq br(M) \setminus E_2$, it holds that $M[R_1 \cup cf(R_2)] \models \bot$. $E^{\pm}(M)$ denotes the set of all inconsistency explanations of M, and $E^{\pm}_m(M)$ the set of all pointwise subset-minimal ones.

Example 5.6. *Suppose to improve the mMCS M_{ph} from Example 5.3 by adding a further bridge rule r_1 to ensure that $water$ is considered ineffective, even if C_4 does not derive this. The bridge rule r_1 is:*

$$(1 : insert(ineffective(water, leg\_pneu))) \leftarrow \mathbf{not}\ (4 : ineffective(water, leg\_pneu)).$$

The resulting mMCS M'_{ph} is inconsistent: let t denote $ineffective(water, leg\_pneu)$ and let r_2 be the rule of form $(4 : upd(t) \leftarrow (1 : t)$ in of context C_4; now if $t \notin S_4$ holds, then $t \in S_1$ must hold by r_1, implying $t \in S_4$ holds by r_2, a contradiction; on the other hand, if $t \in S_4$ holds then $t \notin S_1$ follows, which implies that $t \notin S_4$ holds, again a contradiction.

Then M'_{ph} has one minimal explanation, $(\{r_1, r_2\}, \{r_1, r_2\})$, and four minimal diagnoses:

$$D^{\pm}_{m}(M'_{ph}) = \big\{(\{r_1\}, \emptyset), (\{r_2\}, \emptyset),\\ (\emptyset, \{r_1\}), (\emptyset, \{r_2\})\big\}.$$

One of the basic functions of context managers is to ensure an acceptable belief set of a context regardless of its applicable operational statements.

Definition 5.8. *We call a context C_i with knowledge base kb_i in an mMCS M* totally coherent *if for every belief state S of M some $(kb', \mathbf{ACC}_i) \in mng_i(app_i(S), kb_i)$ exists such that $\mathbf{ACC}_i(kb') \neq \emptyset$; and C_i* totally incoherent *if no belief state S fulfils the previous condition.*

Note that any context with an lc-ensuring management function is totally coherent; the opposite need not be the case.

Example 5.7. *All contexts of M_{ph} from Example 5.4 and M'_{ph} from Example 5.6 are totally coherent. Indeed, for C_2 and C_3 this holds trivially, as they have no bridge rules. Similarly, for C_1 each insertion of tuples yields a knowledge base with an acceptable belief set (as $\mathit{effective}$ and $\mathit{ineffective}$ may share tuples, the belief set may be regarded to be inconsistent). For C_4 we observe that the logic program has no stable model only if $\mathit{effective}(X, B)$ and $\mathit{not\ effective}(X, B)$ are derived for some (X, B). The atom $\mathit{not\ effective}(X, B)$ however, can only hold for atoms added through the upd operation whose semantics guarantees that $\mathit{effective}(X, B)$ is no longer derivable. Therefore C_4 has always an acceptable belief set.*

Note that M'_{ph} is inconsistent, although all of its contexts are totally coherent.

Total coherence cannot prevent inconsistency of the whole mMCS caused by cyclic information flow. On the other hand, context managers can ensure the existence of diagnoses. To guarantee the existence of a diagnosis for an MCS M, [52] requires that $M[\emptyset]$ is consistent. For mMCS we can replace this premise by the considerably weaker assumption that no context is totally incoherent.

Proposition 5.2. *Let M be an inconsistent mMCS. Then $D^{\pm}(M) \neq \emptyset$ if no context of M is totally incoherent.*

Proof. Let $M = (C_1, \dots, C_n)$ for some $n \geq 1$. Since no context is totally incoherent, it holds for every $1 \leq i \leq n$ that there exists exists a belief state $S^i = (S^i_1, \dots, S^i_n)$ of M such that some $(kb'_i, \mathbf{ACC}_i) \in mng_i(app_i(S^i), kb_i)$ exists with $\mathbf{ACC}_i(kb') \neq \emptyset$. Since $\mathbf{ACC}_i(kb')$ is not empty, let $S_i \in \mathbf{ACC}_i(kb')$ be one accepted belief set and let $S^w = (S_1, \dots, S_n)$ be the belief state where each S_i is such an accepted belief set, i.e. for all $1 \leq i \leq n$ is $S_i \in \mathbf{ACC}_i(kb'_i)$ with $(kb'_i, \mathbf{ACC}_i) \in mng_i(app_i(S^i), kb_i)$. Intuitively, each belief set S_i of S^w is acceptable if the right set R_i of operational statements is given to mng_i. We now craft a diagnosis which ensures that R_i results from the modified system as follows: delete all bridge rules and add those bridge rules in condition-free form whose operational statement in the head occurs in R_i. Because all bridge rules of the resulting mMCS are condition-free, all those bride rules are applicable also in S^w, making it an equilibrium.

Formally: recall that $app_i(S^i) = \{\varphi(r) \mid r \in br_i(M), S^i \models r\}$, and consider the set $R_i = \{r \in br_i(M) \mid S^i \models r\}$ of bridge rules applicable in S^i. Note that $br_i(M[br(M), R_i]) = cf(R_i)$ and that $app_i(S^i) = \{\varphi(r) \mid r \in cf(R_i)\}$, because $\varphi(r) = \varphi(cf(r))$ holds for all bridge rules r. Also note that for any belief state S' it holds that $\{\varphi(r) \mid r \in br_i(M[br(M), R_i]), S' \models r\} = \{\varphi(r) \mid r \in cf(R_i)\} = app_i(S^i)$, because for any $r \in cf(R_i)$ it holds that $body(r) = \emptyset$ and thus $S' \models r$.

Let $R^\cup = R_1 \cup \ldots \cup R_n$ be the union of all such R_i and observe that $br_i(M[br(M), R^\cup]) = br_i(M[br(M), R_i])$ for all $1 \leq i \leq n$. Since all bridge rules in $M[br(M), R^\cup]$ are unconditional, it holds that $\{\varphi(r) \mid r \in app(br_i(M[br(M), R^\cup]), S^w)\} = app_i(S^i)$, hence it holds for all $1 \leq i \leq n$ that $S_i \in \mathbf{ACC}_i(kb'_i)$ where $(kb'_i, \mathbf{ACC}_i) \in mng_i(\{\varphi(r) \mid r \in app(br_i(M[br(M), R^\cup]), S^w)\}, kb_i)$. In other words, S^w is an equilibrium of $M[br(M), R^\cup]$; hence S^w is a witnessing equilibrium of $(br(M), R^\cup) \in D^\pm(M)$. □

Note that the converse of Proposition 5.2 is not true, i.e., a totally incoherent mMCS M may have a diagnosis. The reason here is that total incoherence is defined with respect to all possible sets of applicable bridge rules and not with respect to all possible sets of knowledge-base formulas added by bridge rules. There can be more sets of possible knowledge-base formulas than there might be sets of heads of applicable bridge rules (if no modifications of the bridge rules are taken into account). Since a diagnosis is able to enforce any possible sets of knowledge-base formulas, there still can be diagnoses for a totally incoherent context. The following example shows that behaviour in more detail.

Example 5.8. *Let $M = (C_1)$ be an mMCS where C_1 is based on the abstract logic for ASP $L_\Sigma^{asp} = (\mathbf{KB}, \mathbf{BS}, \mathbf{ACC})$. The logic suite of C_1 is then $LS_1 = (\mathbf{KB}, \mathbf{BS}, \{\mathbf{ACC}\})$. The management base $OP_1 = \{add\}$, and the management function mng_1 adds all statements, i.e., for $kb \in \mathbf{KB}$, $O \subseteq \Phi_{LS_1}^{OP_1}$ is $mng_1(O, kb) = \{(kb \cup \{s \mid add(s) \in O\}, \mathbf{ACC})\}$. Furthermore, the knowledge base and bridge rules of C_1 are:*

$$kb_1 = \{\bot \leftarrow a, b. \quad c.\}$$
$$br_1 = \{\, (1 : add(a.)) \leftarrow \top. \quad (1 : add(b.)) \leftarrow \top.\}$$

Any belief state $S = (S_1)$ of M is such that $S_1 \subseteq \{a, b, c\}$ and it holds for all such S_1 that $app_1(S_1) = \{add(a.), add(b.)\}$ since both bridge rules of C_1 are applicable in any belief state. Hence for any belief state S it holds that $mng_1(app_1(S, M), kb_1) = \{(kb_1 \cup \{a. \quad b.\}, \mathbf{ACC})\}$. By the semantics of ASP it then follows that $\mathbf{ACC}(kb_1 \cup \{a. \quad b.\}) = \emptyset$. Since this holds for all belief states, C_1 is totally incoherent. Nevertheless, $(br_1(M), \emptyset)$ is a diagnosis, since $M[\emptyset]$ has an equilibrium, namely $S = (\{c\})$, because $app_1(S, M[\emptyset]) = \emptyset$, hence $mng_1(app_1(S, M[\emptyset]), kb_1) = \{(kb_1, \mathbf{ACC})\}$ and $\mathbf{ACC}(kb_1) = \{\{c\}\}$. Therefore, $(br_i(M), \emptyset) \in D^\pm(M)$ holds although every context of M is totally incoherent.

Note that forbidding the use of $\top$ in the body of a bridge rule would not be sufficient to remedy this issue, since C_1 is also totally incoherent if the body of both bridge rules is $(1 : c)$.

We can strengthen the notion of coherence for contexts further, by requiring that the context is coherent not only for all possible belief sets, but also for all possible sets of knowledge-base formulas added by bridge rules. This ensures coherence, even under modification of bridge rules.

Definition 5.9. *We call a context C_i with knowledge base kb_i in an mMCS M* omni-coherent *if for every set $H \subseteq \{\varphi(r) \mid r \in br_i(M)\}$ of operational statements of bridge rules, some $(kb', \mathbf{ACC}_i) \in mng_i(H, kb_i)$ exists such that $\mathbf{ACC}_i(kb') \neq \emptyset$. We call C_i* omni-incoherent *if no such $H \subseteq \{\varphi(r) \mid r \in br_i(M)\}$ fulfils the previous condition.*

Observe that if a context is omni-coherent, then it also is totally coherent, because for all belief states S it holds that $app_i(S, M) \subseteq \{\varphi(r) \mid r \in br_i(M)\}$. For omni-incoherent contexts the previous proposition now holds in both directions.

Proposition 5.3. *Let M be an inconsistent mMCS. Then $D^{\pm}(M) \neq \emptyset$ iff no context of M is omni-incoherent.*

Proof. "$\Rightarrow$": Let $D^{\pm}(M) \neq \emptyset$ hold. We have to show that no context of M is omni-incoherent. Towards contradiction, assume that there exists $1 \leq i \leq n$ such that context C_i is omni-incoherent. Let $(D_1, D_2) \in D^{\pm}(M)$ and $S \in \text{EQ}(M[D_1, D_2])$ where $S = (S_1, \ldots, S_n)$ and let $H = app_i(S, M[D_1, D_2])$. Since $\{\varphi(r) \mid r \in br_i(M[D_1, D_2])\} \subseteq \{\varphi(r) \mid r \in br_i(M)\}$, it follows by the omni-incoherence of C_i that there exists no $(kb', \mathbf{ACC}_i) \in mng_i(app_i(S, M[D_1, D_2]), kb_i)$ such that $\mathbf{ACC}_i(kb') \neq \emptyset$.
Hence $S_i \notin \mathbf{ACC}_i(kb')$ holds for all $(kb', \mathbf{ACC}_i) \in mng_i(app_i(S, M[D_1, D_2]), kb_i)$. This contradicts that that $S \in \text{EQ}(M[D_1, D_2])$. Therefore the assumption that there exists an omni-incoherent C_i is wrong and it follows that no context of M is omni-incoherent.

"$\Leftarrow$": We can directly re-use the proof of Proposition 5.2 to construct a diagnosis of M as follows.

Let $M = (C_1, \ldots, C_n)$, $n \geq 1$. Since no context is omni-incoherent, it holds for every $1 \leq i \leq n$ that there exists some belief state $S^i = (S^i_1, \ldots, S^i_n)$ of M such that some $(kb'_i, \mathbf{ACC}_i) \in mng_i(app_i(S^i, M), kb_i)$ exists with $\mathbf{ACC}_i(kb') \neq \emptyset$. Since $\mathbf{ACC}_i(kb')$ is not empty, let $S_i \in \mathbf{ACC}_i(kb')$ be an arbitrary accepted belief set and let $S^w = (S_1, \ldots, S_n)$ be the belief state where each S_i is such an accepted belief set w.r.t. the belief state S^i, i.e. for all $1 \leq i \leq n$ is $S_i \in \mathbf{ACC}_i(kb'_i)$ with $(kb'_i, \mathbf{ACC}_i) \in mng_i(app_i(S^i, M), kb_i)$.

Recall that $app_i(S^i, M) = \{\varphi(r) \mid r \in br_i(M), S^i \models r\}$, and consider the set $R_i = \{r \in br_i(M) \mid S^i \models r\}$ of bridge rules applicable in S^i. Note that $br_i(M[br(M), R_i]) = cf(R_i)$ and that $app_i(S^i, M) = \{\varphi(r) \mid r \in cf(R_i)\}$, because $\varphi(r) = \varphi(cf(r))$ holds for all bridge rules r. Also note that for any belief state S' it holds that $\{\varphi(r) \mid r \in br_i(M[br(M), R_i]), S' \models r\} = \{\varphi(r) \mid r \in cf(R_i)\} = app_i(S^i, M)$, because for any $r \in cf(R_i)$ it holds that $body(r) = \emptyset$ and thus $S' \models r$.

Let $R^{\cup} = R_1 \cup \cdots \cup R_n$ be the union of all such R_i and observe that $br_i(M[br(M), R^{\cup}]) = br_i(M[br(M), R_i])$ for all $1 \leq i \leq n$. Since all bridge rules in $M[br(M), R^{\cup}]$ are unconditional, it holds that $app_i(S^w, M[br(M), R^{\cup}]) = app_i(S^i, M)$, hence it holds for all $1 \leq i \leq n$ that $S_i \in \mathbf{ACC}_i(kb'_i)$ where $(kb'_i, \mathbf{ACC}_i) \in mng_i(app_i(S^w, M[br(M), R^{\cup}]), kb_i)$. In other words, S^w is an equilibrium of $M[br(M), R^{\cup}]$; hence S^w is a witnessing equilibrium of $(br(M), R^{\cup}) \in D^{\pm}(M)$. □

One of the main observations regarding inconsistency and managed contexts is that acyclic and thus in particular hierarchic mMCS with totally coherent contexts are always consistent.

Proposition 5.4. *Any acyclic mMCS with totally coherent contexts has an equilibrium.*

Proof. Let M be an acyclic mMCS with totally coherent contexts. Since M is acyclic, there exists a topological ordering $C_{\ell_1}, \dots, C_{\ell_n}$ of the contexts of M such that $(c:p) \in \mathit{body}^{\pm}(r)$, $r \in \mathit{br}_i(M)$, $C_c = C_{\ell_j}$, and $C_i = C_{\ell_k}$ implies that $j < k$, i.e., bridge rules of context C_i only refer to beliefs of contexts that occur earlier in the ordering.

Using this topological ordering, we construct in the following a series $S^1, \dots, S^n$ of belief states where each $S^i = (S^i_1, \dots, S^i_n)$ is such that (1) $S^i_{\ell_k} = S^{i-1}_{\ell_k}$ for all $i > k$ and (2) for all C_{ℓ_k} it holds that $S^k_{\ell_k} \in \mathbf{ACC}'(kb')$ where $(kb', \mathbf{ACC}') \in \mathit{mng}_{\ell_k}(\mathit{app}_{\ell_k}(S^k, M), kb_{\ell_k})$. Intuitively, the topologically first context C_{ℓ_1} accepts the belief set $S^1_{\ell_1}$ of the belief state S^1 and the belief set $S^i_{\ell_1}$ of all later belief states S^i, $1 \leq i \leq n$. The topologically second context accepts the belief set $S^2_{\ell_2}$ of the second belief state S^2 and the belief set $S^i_{\ell_2}$ of all later belief states S^i, $2 \leq i \leq n$. Unitl the last context C_{ℓ_n} accepts the belief set $S^n_{\ell_n}$ of the belief state S^n and since all contexts accept S^n it holds that $S^n \in \mathrm{EQ}(M)$.

Let $S^0 = (S^0_1, \dots, S^0_n) = (\emptyset, \dots, \emptyset)$, we then define S^k for $1 \leq k \leq n$ inductively as follows. Let kb_{ℓ_k} be the knowledge base of C_{ℓ_k}; now consider $\mathit{mng}_{\ell_k}(\mathit{app}_{\ell_k}(S^{k-1}), kb_{\ell_k})$ and observe that by total coherence of C_{ℓ_k} it holds that there exists $(kb', \mathbf{ACC}') \in \mathit{mng}_{\ell_k}(\mathit{app}_{\ell_k}(S^{k-1}), kb_{\ell_k})$ such that there exists $S_{\ell_k} \in \mathbf{ACC}'(kb')$. Then, we define

$$S^k = (S^{k-1}_1, \dots, S^{k-1}_{\ell_k - 1}, S_{\ell_k}, S^{k-1}_{\ell_k + 1}, \dots, S^{k-1}_n).$$

Observe that for every $1 \leq \ell_k \leq n$ it holds that $\mathit{app}_{\ell_k}(S^k) = \mathit{app}_{\ell_k}(S^n)$ since the bridge rules of C_{ℓ_k} only refer to beliefs from contexts C_{ℓ_j} with $j < k$ and $S^j_{\ell_j} = S^{j+1}_{\ell_j} = \dots = S^n_{\ell_j}$. Hence, $S^n_{\ell_k} = S^k_{\ell_k} \in \mathbf{ACC}'(kb')$ with $(kb', \mathbf{ACC}') \in \mathit{mng}_{\ell_k}(\mathit{app}_{\ell_k}(S^n), kb_{\ell_k}) = \mathit{mng}_{\ell_k}(\mathit{app}_{\ell_k}(S^k), kb_{\ell_k})$. Since this holds for all $1 \leq k \leq n$, it follows that $S^n \in \mathrm{EQ}(M)$ holds, which proves the statement.

□

Since any omni-coherent context also is totally coherent, the above result also extends to such mMCS.

Corollary 5.1. *Any acyclic mMCS with omni-coherent contexts has an equilibrium.*

Our main contribution to inconsistency in mMCS with omni-coherent contexts is the observation of Theorem 5.1 that any minimal inconsistency explanation of such an mMCS contains a cycle and all bridge rules of such an explanation are cycle-reaching. The examples following the theorem also demonstrate that some probable refinements of the theorem do not hold, i.e., the characterisation is quite precise.

Before proceeding, we need to introduce some notations regarding cycles. Let $M = (C_1, \dots, C_n)$ be an mMCS. Then the directed graph *induced* by M is $G^M = (V, E)$ with vertices $V = \{C_1, \dots, C_n\}$ and edges $E = \{e(r) \mid r \in \mathit{br}(M)\}$ where $e(r) = \{(C_i, C_j) \mid r \in \mathit{br}_j(M), (i : p) \in \mathit{body}^{\pm}(r)\}$. Intuitively, G^M contains an edge from C_i to C_j whenever there exists a bridge rule of C_j whose body contains a belief from C_i.[2] A path p in $G^M = (V, E)$ is any sequence $p = (e_1, \dots, e_k)$ of edges with $e_1, \dots, e_k \in E$ and $e_1 = (v_1, v_2), \dots, e_k = (v_k, v_{k+1})$. A path $p = (e_1, \dots, e_k)$ in G^M is a *cycle*, if $e_1 = (v_1, v_2)$ and $e_k = (v_k, v_1)$ hold. A sequence $(r_1, \dots, r_k)$ of bridge rules *corresponds* to a path $p = (e_1, \dots, e_k)$ if $e_1 \in e(r_1), \dots, e_k \in e(r_k)$ all hold. A sequence $(r_1, \dots, r_k)$ of bridge rules is a *cycle* if there exists a corresponding path $p = (e_1, \dots, e_k)$ and p is a cycle.

Given a cycle $p = (e_1, \dots, e_k)$, we say a bridge rule r reaches p in G^M, if there exists a path $p' = (e'_1, \dots, e'_m, e_1)$ in G^M such that $e'_1 \in e(r)$ holds. A bridge rule $r \in br(M)$ is *cycle-reaching* in G^M if there exists a cycle p in G^M such that r reaches p in G^M.

Observe that the modifications considered by an explanation of an mMCS M yield an mMCS M' such that G^M contains all edges of $G^{M'}$. Formally, let $R \subseteq br(M) \cup cf(br(M))$ hold, let $G^M = (V, E)$ and let $G^{M[R]} = (V', E')$. Then it holds that $E' \subseteq E$, i.e., $G^{M[R]}$ contains no more edges than G^M. The reason is that bridge rules in $cf(br(M))$ do not refer to any beliefs from other contexts, i.e., for any $r \in cf(br(M))$ it holds that $body(r) = \emptyset$, thus $C_b(r) = \emptyset$, and consequently $e(r) = \emptyset$. From that it also follows that if a bridge rule $r \in br(M)$ is not cycle-reaching in G^M, then r also is not cycle-reaching in $G^{M[R]}$ for any $R \subseteq br(M) \cup cf(br(M))$.

Theorem 5.1. *Let M be an inconsistent mMCS with omni-coherent contexts. Then for every minimal explanation $(E_1, E_2) \in E^{\pm}_m(M)$ there exists a cycle $cyc = (r_1, \dots, r_k)$ such that $\{r_1, \dots, r_k\} \subseteq E_1$ holds and every $r \in E_1$ is cycle-reaching in G^M.*

Proof. Let $M = (C_1, \dots, C_n)$ be an inconsistent mMCS with omni-coherent contexts and let $(E_1, E_2) \in E^{\pm}_m(M)$. Towards contradiction, assume that there exists no cycle $(r_1, \dots, r_k)$ such that $\{r_1, \dots, r_k\} \subseteq E_1$. Since $(E_1, E_2) \in E^{\pm}_m(M)$, it follows for all $E_1 \subseteq R_1 \subseteq br(M)$ and $R_2 \subseteq br(M) \setminus E_2$ that $M[R_1 \cup cf(R_2)] \models \bot$ holds. Consider $R_1 = E_1$ and $R_2 = \emptyset$ and observe that $M[R_1 \cup cf(R_2)] = M[E_1]$ is an acyclic mMCS with omni-coherent contexts. Hence by Corollary 5.1 it follows that $M[E_1]$ has an equilibrium, i.e., $M[R_1 \cup cf(R_2)] \not\models \bot$. This contradicts that $(E_1, E_2) \in E^{\pm}_m(M)$ holds. Hence there exists a cycle $cyc = (r_1, \dots, r_k)$ such that $\{r_1, \dots, r_k\} \subseteq E_1$.

It remains to show that every $r \in E_1$ is cycle-reaching in G^M. Towards contradiction, assume there is $r \in E_1$ such that r is not cycle-reaching in G^M. In the following we show that given such r it holds that $(E_1 \setminus \{r\}, E_2)$ also is an explanation, contradicting the minimality of (E_1, E_2). We proceed in three steps. In the first step, we show that considering any R_1, R_2 with $E_1 \subseteq R_1 \subseteq br(M)$ and $R_2 \subseteq br(M) \setminus E_2$ it holds for the resulting $M[R_1 \cup cf(R_2)]$ that in every belief state S one of the contexts that is not reachable by r is causing S to be not an equilibrium, i.e., the cause of inconsistency always is some context that is not reachable by r. In the second step, we show that removing r has no influence on applicable bridge rules of these contexts not reachable by r. In the third step we combine the two observations to derive that for all relevant bridge rules R_1, R_2 and all belief states S it holds that $M[R_1 \setminus \{r\} \cup cf(R_2)]$ is inconsistent, which is equivalent to showing that $(E_1 \setminus \{r\}, E_2) \in E^{\pm}(M)$ holds.

In the following, let $V^+(r) \subseteq V$ be the set of nodes of $G^M = (V, E)$ that are reachable from the node where r belongs to, i.e., $V^+(r)$ is the set of nodes reachable in G^M from the node in the

[2]The direction of edges in G^M is the same as in all illustrations throughout this thesis. Note, however, that this is exactly the opposite direction as used in the import closures of [5, 6].

singleton set $\{v \in V \mid (v', v) \in e(r)\}$. Since V is the set of contexts of M, $V^+(r)$ is a subset of these contexts. Furthermore, let A be the set of indices of nodes/contexts reachable from r and let B be the set of indices of nodes/contexts not reachable from r, formally:

$$A = \{i \in \mathbb{N} \mid C_i \in V^+(r)\}$$
$$B = \{i \in \mathbb{N} \mid C_i \in V \setminus V^+(r)\} = \{1, \dots, n\} \setminus A$$

We now show that for every belief state $S = (S_1, \dots, S_n)$ there exists some $i \in B$ such that there exists no $(\mathbf{ACC}_i, kb'_i) \in mng_i(app_i(S, M[R_1 \cup cf(R_2)]), kb_i)$ with $S_i \in \mathbf{ACC}_i(kb'_i)$. Informally, we show that every belief state is not acceptable by some context of B. Towards contradiction, assume that there exists $S^0 = (S^0_1, \dots, S^0_n)$ such that for every $i \in B$ it holds that there exists $(\mathbf{ACC}_i, kb'_i) \in mng_i(app_i(S^0, M[R_1 \cup cf(R_2)]), kb_i)$ with $S_i \in \mathbf{ACC}_i(kb'_i)$.

Since r is not cycle-reaching, it holds that the resulting induced sub-graph is acyclic. Hence, there exists a topological ordering $to : \{C_j \mid j \in A\} \to \{1, \dots, k\}$ of the contexts of $V^+(r)$ such that $to(C_i) \leq to(C_j)$ implies that the bridge rules of C_i do not use beliefs of C_j for any $i, j \in A$. Formally, $to(C_i) = \ell$ implies that for every context C_j with $to(C_j) \geq \ell$ it holds that $j \notin \bigcup_{r \in br_i(M)} C_b(r)$. Note that $\bigcup_{r \in br_i(M)} C_b(r)$ is a superset of $\bigcup_{r \in br_i(M[R_1 \cup cf(R_2)])} C_b(r)$ since $R_1 \subseteq br(M)$ holds and unconditional bridge rules refer to no other contexts. Hence, the topological ordering to also works for $M[R_1 \cup cf(R_2)]$ where $E_1 \subseteq R_1 \subseteq br(M)$, $R_2 \subseteq br(M) \setminus E_2$ are arbitrary.

Using to, we then inductively construct a sequence of belief states. Let j be such that $to(C_j) = 1$ and observe that since C_j is omni-coherent, it holds that there exists a belief set $S^1_j \in \mathbf{ACC}_j(kb'_j)$ with $(\mathbf{ACC}_j, kb'_j) \in mng_j(app_j(S^0, M[R_1 \cup cf(R_2)]), kb_j)$. Now define $S^1 = (S^1_1, \dots, S^n_1)$ such that for all $1 \leq i \leq n$ with $i \neq j$ it holds that $S^1_i = S^0_i$ and S^1_j is the belief set above. Notice that $app_j(S^1, M[R_1 \cup cf(R_2)]) = app_j(S^0, M[R_1 \cup cf(R_2)])$ by construction of S^1. Consequently, it holds that $S^1_j \in \mathbf{ACC}_j(kb'_j)$ with $(\mathbf{ACC}_j, kb'_j) \in mng_j(app_j(S^1, M[R_1 \cup cf(R_2)]), kb_j)$. Furthermore, note that for all $i \in B$ it holds that $app_i(S^0, M[R_1 \cup cf(R_2)]) = app_i(S^1, M[R_1 \cup cf(R_2)])$, because by definition of $V^+(r)$ no context of B has a bridge rule that refers to any context of A. Since it further holds for all $j \in B$ that $S^1_j = S^0_j$, it therefore holds that there exists $(\mathbf{ACC}_i, kb'_i) \in mng_i(app_i(S^1, M[R_1 \cup cf(R_2)]), kb_i)$ with $S^1_i \in \mathbf{ACC}_i(kb'_i)$.

Let $j \in \{2, \dots, k\}$ and let C_i be such that $to(C_i) = j$. Then the belief state $S^j = (S^j_1, \dots, S^j_n)$ is such that for all $1 \leq \ell \leq n$ with $\ell \neq i$ it holds that $S^j_\ell = S^{j-1}_\ell$ and S^j_i is a belief set $S^j_i \in \mathbf{ACC}_i(kb_i)$ with $(\mathbf{ACC}_i, kb'_i) \in mng_i(app_i(S^{j-1}, M[R_1 \cup cf(R_2)]), kb_i)$. Note that such an S^j_i exists since C_i is omni-coherent and since r is not cycle-reaching and C_i is reachable from r, it holds that C_i has no bridge rule referring to itself. Hence, it holds that $app_i(S^{j-1}, M[R_1 \cup cf(R_2)]) = app_i(S^j, M[R_1 \cup cf(R_2)])$. Consequently, it holds that there exists $(\mathbf{ACC}_i, kb'_i) \in mng_i(app_i(S^j, M[R_1 \cup cf(R_2)]), kb_i)$ such that $S^j_i \in \mathbf{ACC}_i(kb'_i)$ holds.

Consider $S^k = (S^k_1, \dots, S^k_n)$ and observe that for all $i \in B$ it holds that $S^k_i = S^{k-1}_i = \dots = S^0_i$. Hence it holds that $app_i(S^k, M[R_1 \cup cf(R_2)]) = app_i(S^0, M[R_1 \cup cf(R_2)])$ and therefore it holds that there exists $(\mathbf{ACC}_i, kb'_i) \in mng_i(app_i(S^1, M[R_1 \cup cf(R_2)]), kb_i)$ with $S^1_i \in \mathbf{ACC}_i(kb'_i)$. It furthermore holds for all $j \in A$ that there exists $(\mathbf{ACC}_j, kb'_j) \in$

$mng_j(app_j(S^k, M[R_1 \cup cf(R_2)]), kb_j)$ with $S_j^k \in \mathbf{ACC}_j(kb'_j)$, because $S_j^k = S_j^\ell$ for $to(C_j) = \ell$ and as argued in the inductive step above it holds that $(\mathbf{ACC}_j, kb'_j) \in mng_j(app_j(S^k, M[R_1 \cup cf(R_2)]), kb_j)$ with $S_j^k \in \mathbf{ACC}_j(kb'_j)$. In summary, it holds for all $1 \leq i \leq n$ that there exists $(\mathbf{ACC}_i, kb'_i) \in mng_i(app_i(S^k, M[R_1 \cup cf(R_2)]), kb_i)$ with $S_i^k \in \mathbf{ACC}_i(kb'_i)$, i.e., it holds that $S^k \in \textsc{Eq}(M[R_1 \cup cf(R_2)])$. This contradicts that $M[R_1 \cup cf(R_2)] \models \bot$ and therefore also contradicts that $(E_1, E_2) \in E_m^{\pm}(M)$ holds. Hence the assumption that there exists such a belief state S^0 is false, and it holds for every belief state $S = (S_1, \ldots, S_n)$ that there exists $i \in B$ such that there exists no $(\mathbf{ACC}_i, kb'_i) \in mng_i(app_i(S, M[R_1 \cup cf(R_2)]), kb_i)$ with $S_i \in \mathbf{ACC}_i(kb'_i)$.

We now show that for all $R \subseteq br(M) \cup cf(br(M))$, for all belief states $S = (S_1, \ldots, S_n)$ and for all $i \in B$ it holds that $app_i(S, M[R]) = app_i(S, M[R \setminus \{r\}])$. Recall that B is the set of indices of those contexts that are not in $V^+(r)$. Since r belongs to a context in $V^+(r)$, it thus holds that $br_i(M[R]) = br_i(M[R \setminus \{r\}])$. It thus follows that $app_i(S, M[R]) = app_i(S, M[R \setminus \{r\}])$.

Summarising, we know that for every belief state $S = (S_1, \ldots, S_n)$ there exists some $i \in B$ such that no $(\mathbf{ACC}_i, kb'_i) \in mng_i(app_i(S, M[R_1 \cup cf(R_2)]), kb_i)$ exists with $S_i \in \mathbf{ACC}_i(kb'_i)$. By the above, it also holds that there exists no $(\mathbf{ACC}_i, kb'_i) \in mng_i(app_i(S, M[R_1 \setminus \{r\} \cup cf(R_2)]), kb_i)$ with $S_i \in \mathbf{ACC}_i(kb'_i)$. Hence S is not an equilibrium of $M[R_1 \setminus \{r\} \cup cf(R_2)]$ and since this holds for all belief states S, it follows that $M[R_1 \setminus \{r\} \cup cf(R_2)] \models \bot$.

Furthermore, since this holds for all R_1, R_2 with $E_1 \subseteq R_1 \subseteq br(M)$ and $R_2 \subseteq br(M) \setminus E_2$, it follows that $(E_1 \setminus \{r\}, E_2) \in E^{\pm}(M)$ holds. This clearly contradicts that $(E_1, E_2) \in E_m^{\pm}(M)$ holds. Therefore, the assumption that there exists an $r \in E_1$ such that r is not cycle-reaching in G^M is false; thus all $r \in E_1$ are cycle-reaching in G^M, which concludes our proof. □

Note, that not every cycle causes inconsistency, and that due to potential non-monotonicity inside contexts the number of negative literals occuring in the bridge rules of a cycle is not relevant for determining whether a cycle will cause inconsistency.

One possible sharpening of the above theorem could be that those bridge rules which are not part of a cycle are directly connected to the cycle, i.e., given an mMCS M and $(E_1, E_2) \in E_m^{\pm}(M)$ such that $r \in E_1$ is not part of a cycle in E_1, then one could expect that r is cycle-reaching in $G^{M[E_1]}$. As the following example shows, however, this need not be the case.

Example 5.9. *Consider an mMCS $M = (C_1, C_2, C_3, C_4, C_5)$ where all contexts use ASP, i.e., $LS_1 = \ldots = LS_5$ with $LS_1 = (\mathbf{KB}_{ASP}, \mathbf{BS}_{ASP}, \{\mathbf{ACC}_{ASP}\})$ stemming from the abstract logic $L_\Sigma^{asp} = (\mathbf{KB}_{ASP}, \mathbf{BS}_{ASP}, \mathbf{ACC}_{ASP})$. All contexts use the same management function and operational base, i.e., for all $1 \leq i \leq 5$ holds $O_i = \{add\}$ and the management functions simply add all head formulas, i.e., $mng_i(O, kb) = \{(kb \cup \{s \mid add(s) \in O\})\}$. The knowledge bases and bridge rules of M are as follows:*

$$
\begin{aligned}
kb_1 &= \{a \leftarrow not\, b.\} & br_1 &= \{r_1 : & (1 : add(b)) &\leftarrow (2 : b).\} \\
kb_2 &= \{b \leftarrow a, c.\} & br_2 &= \{r_2 : & (2 : add(a)) &\leftarrow (1 : a). \\
& & & \quad r_2' : & (2 : add(c)) &\leftarrow (3 : c).\} \\
kb_3 &= \{c \leftarrow not\, d.\} & br_3 &= \{r_3 : & (3 : add(d)) &\leftarrow (4 : d).\} \\
kb_4 &= \{d \leftarrow not\, e.\} & br_4 &= \{r_4 : & (4 : add(e)) &\leftarrow (5 : e).\}
\end{aligned}
$$

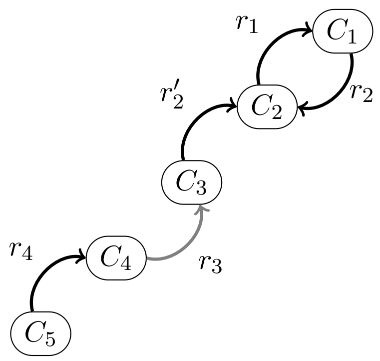


Figure 5.2: The MCS of Example 5.9 whose minimal explanation $(E_1, E_2) = (\{r_1, r_2, r'_2, r_4\}, \{r_1, r_2, r_3\})$ contains in E_1 all bridge rules except r_3. Bridge rule r_4 is cycle-reaching in G^M, but not in $G^{M[E_1]}$.

$$kb_5 = \{e.\} \qquad\qquad br_5 = \{\}$$

The mMCS is also depicted in Figure 5.2. Observe that all contexts of M are omni-coherent, since for any set of operational statements occurring at a context of M, the knowledge base resulting from the management function has an answer set.

Intuitively, the cycle (r_1, r_2) causes inconsistency, if c is present at C_2. Since e is present at C_5, r_4 causes d to be absent in C_4 and r_3 is not applicable. Thus c is derived in C_3 and r'_2 adds c to C_2.

A minimal explanation of M is $(E_1, E_2) = (\{r_1, r_2, r'_2, r_4\}, \{r_1, r_2, r_3\})$ where (r_1, r_2) is a cycle in G^M and bridge rules $r'_2, r_4 \in E_1$ are cycle-reaching in G^M. Removing r_4 from E_1 would result in $M[R_1 \cup cf(R_2)]$ being consistent, for $R_1 = \{r_1, r_2, r_3, r_4\}$ and $R_2 = \emptyset$, hence $(E_1 \setminus \{r_4\}, E_2)$ does not constitute an explanation. One can check that removing any other bridge rules from (E_1, E_2) also does not constitute an explanation, i.e., $(E_1, E_2) \in E^{\pm}_m(M)$ holds.

Therefore there is a minimal explanation (E_1, E_2) for M such that there exists $r \in E_1$ which is not contained in a cycle and also not directly connected to the cycle in $G^{M[E_1]}$.

One could also expect that for any $(E_1, E_2) \in E^{\pm}_m(M)$ of an mMCS M, it holds that E_1 contains at most one cycle. The following example shows that this, however, need not be the case.

Example 5.10. *Consider the mMCS $M = (C_1, C_2, C_3, C_4, C_5)$ using the same logic suites and context managers as the mMCS in Example 5.9, i.e., all contexts of M use ASP. The knowledge bases and bridge rules are as follows:*

$$
\begin{array}{lll}
kb_1 = \{a \vee a'.\} & br_1 = \{\} & \\
kb_2 = \{b \leftarrow a, c.\} & br_2 = \{r_2: & (2 : add(a)) \leftarrow (1 : a). \\
 & \qquad r'_2: & (2 : add(c)) \leftarrow (3 : c).\} \\
kb_3 = \{c \leftarrow not\, b.\} & br_3 = \{r_3: & (3 : add(b)) \leftarrow (2 : b).\} \\
kb_4 = \{b' \leftarrow a', c'.\} & br_2 = \{r_4: & (4 : add(a')) \leftarrow (1 : a'). \\
 & \qquad r'_4: & (4 : add(c')) \leftarrow (5 : c').\}
\end{array}
$$

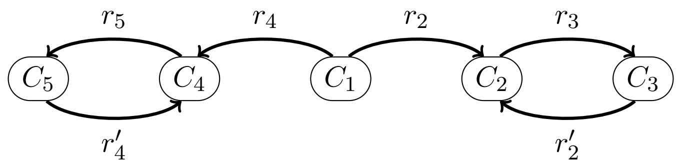


Figure 5.3: The MCS of Example 5.10 whose minimal explanation $(E_1, E_2) = (\{r_2, r'_2, r_3, r_4, r'_4, r_5\}, \{r'_2, r_3, r'_4, r_5\})$ contains two cycles (r_3, r'_2) and (r_5, r'_4).

$$kb_5 = \{c' \leftarrow not\, b'.\} \qquad br_3 = \{r_5 : \quad (5 : add(b')) \leftarrow (4 : b').\}$$

The resulting mMCS is depicted in Figure 5.3. Observe that all contexts of M are omni-coherent, for the same reason as in Example 5.9.

Intuitively, C_1 has two answer sets, one containing a the other containing a'. For the one containing a, the cyclic information flow through C_2 and C_3 causes inconsistency by the cycle (r'_2, r_3). Analogous for the answer set containing a', the cycle (r'_4, r_5) causes inconsistency. M is inconsistent and one can check that $(E_1, E_2) = (\{r_2, r'_2, r_3, r_4, r'_4, r_5\}, \{r'_2, r_3, r'_4, r_5\})$ is an explanation. Indeed, (E_1, E_2) is a minimal explanation since any removal of one bridge rule makes the resulting mMCS consistent. Since E_1 contains two cycles, this shows that even for omni-coherent mMCS there are minimal explanations containing more than one cycle.

5.5 Expressiveness of MCS and mMCS

In this section we show that each MCS can be cast into an mMCS where the management base of each context contains a single operation name for addition and every context manager simply adds all applicable bridge rules. We then show that each mMCS can be cast into an MCS where the acceptability function of each context is internally using the functionality of the context manager, i.e., we show that an acceptability function of an MCS may hide a context manager. This establishes that MCS and mMCS have the same expressivity.

An ordinary context $C_i = (L_i, kb_i, br_i)$ with logic $L_i = (\mathbf{BS}_{L_i}, \mathbf{KB}_{L_i}, \mathbf{ACC}_{L_i})$ can be turned quite easily into a managed context $C'_i = (LS_i{}', kb_i, br'_i, OP_i, add_i)$ over $LS_i = (\mathbf{BS}_{L_i}, \mathbf{KB}_{L_i}, \{\mathbf{ACC}_{L_i}\})$, where $OP_i = \{add_i\}$ and add_i interprets $add(f)$ as addition of f, i.e., $add_i(O, kb_i) = \{(kb_i \cup \{s \mid add(s) \in O\}, \mathbf{ACC}_{L_i})\}$ and $br'_i = \{(i : add(s)) \leftarrow body(r). \mid r \in br_i \wedge \varphi(r) = s\}$. We call C'_i the *management version* of C_i and for convenience we identify both. Managed MCS thus generalise ordinary MCS.

Proposition 5.5. *Let $M = (C_1, \ldots, C_n)$ be an MCS and $M' = (C'_1, \ldots, C'_m)$ the mMCS where each C'_i is the management version of C_i. Then S is an equilibrium of M iff S is an equilibrium of M'.*

Proof. Note that for every $1 \leq i \leq n$ it holds that $r \in br_i$ holds iff $(i : add(s)) \leftarrow body(r). \in br'_i$ holds. Therefore for any belief state S holds that $S \models r$ iff $S \models (i : add(\varphi(r))) \leftarrow$

$body(r)$. By that and from the definition of br'_i it follows that $\{\varphi(r) \mid r \in br'_i, S|\!\!\rightsquigarrow r\} = \{add(\varphi(r)) \mid r \in br_i, S|\!\!\rightsquigarrow r\} = \{add(s) \mid s \in app_i(S, M)\}$, where $1 \leq i \leq n$ holds. Since $add_i(O, kb_i) = \{(kb_i \cup \{s \mid add(s) \in O\}, \mathbf{ACC}_{L_i})\}$, it holds for all $1 \leq i \leq n$ that $add_i(\{\varphi(r) \mid r \in br'_i, S|\!\!\rightsquigarrow r\}, kb_i) = \{(kb_i \cup app_i(S, M), \mathbf{ACC}_{L_i})\}$. W.l.o.g. let $S = (S_1, \ldots, S_n)$. Then it holds for all $1 \leq i \leq n$ that $S_i \in \mathbf{ACC}_{L_i}(kb_i \cup app_i(S, M))$ holds iff there exists $(kb'_i, \mathbf{ACC}'_i) \in add_i(kb_i, app_i(S, M'))$ with $S_i \in \mathbf{ACC}'_i(kb'_i)$. In other words, S is an equilibrium of M iff S is an equilibrium of M'. □

The idea to show that MCS can simulate mMCS is that the management function and acceptability function of the managed context are combined into a new acceptability function which then serves in the ordinary context. Since the acceptability function of an ordinary MCS only receives a knowledge base as input, the operational statements of the bridge rules of the mMCS must be allowed as knowledge-base elements. Furthermore, measures have to be taken to ensure that these additional knowledge-base elements cannot be confused with knowledge-base elements of the original context of the mMCS. We thus construct an abstract logic whose knowledge bases are composed of knowledge-base elements and operational statements that are preceded by a new symbol not occurring in the original knowledge base to distinguish the operational statements from knowledge-base elements.

Given a logic suite $LS = (\mathbf{KB}, \mathbf{BS}, \boldsymbol{\mathcal{ACC}})$ and a management base OP, the *overlap-free combination* of LS and OP is the set $\mathbf{KB}^{OP}_{LS} = \{kb \cup \{ns(o) \mid o \in O\} \mid kb \in \mathbf{KB}, O \subseteq \Phi^{OP}_{LS}\}$ where ns is a new symbol not occurring in Φ_{LS}, i.e., ns occurs in no knowledge base of $\mathbf{KB}$. Notice that for any knowledge-base $kb \in \mathbf{KB}$ and set $o \subseteq OP$ of operational statements, there exists exactly one corresponding set in $\mathbf{KB}^{OP}_{LS}$.

In the following, we denote the set of all operational statements preceded by the new symbol by $ns(\Phi^{OP}_{LS}) = \{ns(o) \mid o \in \Phi^{OP}_{LS}\}$. Since ns is a symbol not occurring in any knowledge base, it is possible for any $kb \in \mathbf{KB}^{OP}_{LS}$ to compute the corresponding knowledge base $kb \setminus ns(\Phi^{OP}_{LS})$ of $\mathbf{KB}$ and the corresponding set of operational statements $kb \cap ns(\Phi^{OP}_{LS})$ in time linear in the size of kb.

Let $M = (C_1, \ldots, C_n)$ be an mMCS and let $C_i = (LS_i, kb_i, br_i, OP_i, mng_i)$ be a managed context with logic suite $LS_i = (\mathbf{KB}, \mathbf{BS}, \boldsymbol{\mathcal{ACC}})$, $1 \leq i \leq n$. The corresponding ordinary context is $C'_i = (L'_i, kb_i, br'_i)$ where the abstract logic $L'_i = (\mathbf{KB}^{OP_i}_{LS_i}, \mathbf{BS}, \mathbf{ACC}_i)$ is based on the overlap-free combination of LS_i and OP_i, the same set of belief sets as for LS_i, and an acceptability function $\mathbf{ACC}_i : \mathbf{KB}^{OP_i}_{LS_i} \to \mathbf{BS}$ as follows:

$$\mathbf{ACC}_i(kb) = \{S_i \cap OUT_i \mid S_i \in \mathbf{ACC}'(kb'),$$
$$(kb', \mathbf{ACC}') \in mng_i(\{o \mid ns(o) \in kb\}, kb \setminus ns(\Phi^{OP}_{LS})\}$$

The knowledge base of C'_i is the same as of C_i, while the head-formula of each bridge rule is wrapped by the new symbol ns, i.e., $br'_i = \{(i : ns(\varphi(r))) \leftarrow body(r). \mid r \in br_i\}$. The operational statements in the head of bridge rules of C_i are turned into ordinary knowledge-base elements of C'_i.

Definition 5.10. *Given an mMCS* $M = (C_1, \ldots, C_n)$*, the* corresponding ordinary MCS *is* $M' = (C'_1, \ldots, C'_n)$ *where for each* $1 \leq i \leq n$*,* C'_i *is the ordinary context corresponding to the managed context* C_i*.*

As the following proposition shows, the output-projected equilibria of the managed MCS are the same as the equilibria of the corresponding ordinary MCS. Output-projected belief states in mMCS are similar to those of MCS. For a context C_i of an mMCS $M = (C_1, \dots, C_n)$, let OUT_i be the set of all beliefs p occurring in the body of some bridge rule $r \in br(M)$, i.e., $OUT_i = \{p \mid (i:p) \in \mathit{body}^{\pm}(r), r \in br(M)\}$. Then, the *output-projection* S^o of a belief state $S = (S_1, \dots, S_n)$ of M is the belief state $S^o = (S^o_1, \dots, S^o_n)$, where $S^o_i = S_i \cap OUT_i$, for $1 \leq i \leq n$.

Proposition 5.6. *Given an mMCS $M = (C_1, \dots, C_n)$, its corresponding ordinary MCS $M = (C'_1, \dots, C'_n)$, and an output-projected belief state S of M. Then, $S \in \mathrm{EQ}^o(M)$ holds iff $S \in \mathrm{EQ}(M')$ holds.*

Proof. First observe that for all $1 \leq i \leq n$ holds that $OUT_i(M) = OUT_i(M')$ since the bodies of all bridge rules are the same in both systems; hence we only write OUT_i in the following.

For the following observations, let $T = (T_1, \dots, T_n)$ be a belief state of M, and let $S = (S_1, \dots, S_n) = T^o$ be the output-projected belief state wrt. T. Thus, it holds that $S_i = T_i \cap OUT_i$ for all $1 \leq i \leq n$ and it holds that T is a belief state of M'.

Let $1 \leq i \leq n$ be arbitrary and observe that $app_i(T, M) = app_i(S, M)$ and $app_i(T, M') = app_i(S, M')$, since S is the output-projection of T and $OUT_i(M) = OUT_i(M')$. Furthermore, $app_i(S, M') = \{ns(o) \mid o \in app_i(T, M)\} = ns(app_i(T, M))$, because there exists $r \in br_i(M)$ iff there exists $r' \in br_i(M')$ such that $body(r) = body(r')$ and $\varphi(r') = ns(\varphi(r))$. Since $kb_i \cap ns(\Phi^{OP_i}_{LS_i} = \emptyset$, it holds that $\{o \mid ns(o) \in kb_i \cup app_i(S, M')\} = app_i(T, M)$ and it holds that $(kb_i \cup app_i(S, M')) \setminus ns(\Phi^{OP}_{LS}) = kb_i$.

In summary, the following holds:

$$
\begin{aligned}
&T_i \cap OUT_i \in \mathbf{ACC}'_i(kb'_i) \text{ holds for some } (kb'_i, \mathbf{ACC}'_i) \in mng(app_i(T, M), kb_i)\\
&\text{iff } \exists (kb', \mathbf{ACC}') \in mng_i(app_i(T, M), kb_i) : T_i \cap OUT_i \in \mathbf{ACC}'(kb')\\
&\text{iff } \exists (kb', \mathbf{ACC}') \in mng_i(\{o \mid ns(o) \in kb_i \cup app_i(S, M')\}, kb_i) : T_i \cap OUT_i \in \mathbf{ACC}'(kb')\\
&\text{iff } \exists (kb', \mathbf{ACC}') \in mng_i(\{o \mid ns(o) \in kb_i \cup app_i(S, M')\}, (kb_i \cup app_i(S, M')) \setminus ns(\Phi^{OP}_{LS})):\\
&\qquad T_i \cap OUT_i \in \mathbf{ACC}'(kb')\\
&\qquad\qquad \text{iff } S_i \in \mathbf{ACC}_i(kb_i \cup app_i(S, M')).
\end{aligned}
$$

Since i was arbitrarily chosen, it holds for all $1 \leq i \leq n$ that $T_i \cap OUT_i \in \mathbf{ACC}'_i(kb'_i)$ holds for some $(kb'_i, \mathbf{ACC}'_i) \in mng(app_i(T, M), kb_i)$ iff $S_i \in \mathbf{ACC}_i(kb_i \cup app_i(S, M'))$ holds. Hence, $T \in \mathrm{EQ}(M)$ holds iff $S \in \mathrm{EQ}(M')$ holds for any belief state T of M such that $S = T^o$. □

Although this proposition only considers output-projected equilibria of the mMCS, the following section shows that these are sufficient to check the consistency of an mMCS (cf. Proposition 5.7). Also note that the notion of an ordinary context C'_i corresponding to the managed context C_i of an mMCS can be changed easily to consider belief sets without output-projection.

Here, $\mathbf{ACC}_i$ of C'_i then is as follows:

$$\mathbf{ACC}_i(kb) = \{S_i \mid S_i \in \mathbf{ACC}'(kb'),$$
$$(kb', \mathbf{ACC}') \in mng_i(\{o \mid ns(o) \in kb\}, kb \setminus ns(\Phi_{LS}^{OP})\}.$$

It is easy to prove that the equilibria of the resulting MCS directly correspond to those of the mMCS.

5.6 Computational Complexity

We consider here the consistency problem $\mathcal{CONS}_{mng}(M)$ in mMCS, i.e., given an mMCS $M = (C_1, \ldots, C_n)$, decide whether it has some equilibrium. For consistency checking, we can concentrate on output-projections of equilibria:

Proposition 5.7. *An mMCS $M = (C_1, \ldots, C_n)$ is consistent iff some output-projected belief state $S' = (S'_1, \ldots, S'_n)$, exists such that, for all $1 \leq i \leq n$, $S'_i \in \{S_i \cap OUT_i \mid S_i \in \mathbf{ACC}_i(kb'_i) \wedge (kb'_i, \mathbf{ACC}_i) \in mng_i(app_i(S', M), kb_i)\}$.*

Proof. By definition, an mMCS M is consistent iff there exists some $S \in \text{EQ}(M)$.

We now show that there exists $S \in \text{EQ}(M)$ iff there exists an output-projected belief state $S' = (S'_1, \ldots, S'_n)$ such that, for all $1 \leq i \leq n$ holds $S'_i \in \{S_i \cap OUT_i \mid S_i \in \mathbf{ACC}_i(kb'_i) \wedge (kb'_i, \mathbf{ACC}_i) \in mng_i(app_i(S', M), kb_i)\}$.

Recall that by definition, it holds for all $1 \leq c \leq n$ and beliefs p that $p \in OUT_c$ holds iff there exists a bridge rule $r \in br(M)$ with $(c : p) \in body^{\pm}(r)$. Therefore it holds for all belief states $S = (S_1, \ldots, S_n)$ and $S' = (S'_1, \ldots, S'_n) = (S_1 \cap OUT_1, \ldots, S_n \cap OUT_n)$ that for every $r \in br(M)$ and for all $(c : p) \in body^{\pm}(r)$ it holds that: $p \in S_c$ holds iff $p \in S'_c$ holds. Consequently, $S \models r$ holds iff $S' \models r$ holds.

"$\Rightarrow$": Let $S = (S_1, \ldots, S_n) \in \text{EQ}(M)$ and consider $S' = (S_1 \cap OUT_1, \ldots, S_n \cap OUT_n) = (S'_1, \ldots, S'_n)$ such that for all $1 \leq i \leq n$ it holds that $S'_i = S_i \cap OUT_i$. Obviously, S' is an output-projected belief state and it holds for any $r \in br(M)$ that $S \models r$ iff $S' \models r$. It thus follows that $app_i(S, M) = app_i(S', M)$ and hence $mng_i(app_i(S', M), kb_i) = mng_i(app_i(S, M), kb_i)$. Since $S \in \text{EQ}(M)$ holds, it follows for all $1 \leq i \leq n$ that $S_i \in \mathbf{ACC}_i(kb'_i)$ holds for some $(\mathbf{ACC}_i, kb'_i) \in mng_i(app_i(S, M), kb_i)$; consequently it also holds that $S_i \in \mathbf{ACC}_i(kb'_i)$ holds for some $(\mathbf{ACC}_i, kb'_i) \in mng_i(app_i(S', M), kb_i)$. Consequently, S' is such that for all $1 \leq i \leq n$ holds $S'_i \in \{S_i \cap OUT_i \mid S_i \in \mathbf{ACC}_i(kb'_i) \wedge (kb'_i, \mathbf{ACC}_i) \in mng_i(app_i(S', M), kb_i)\}$.

"$\Leftarrow$": Let $S' = (S'_1, \ldots, S'_n)$ be such that for all $1 \leq i \leq n$ it holds that $S'_i \in \{S_i \cap OUT_i \mid S_i \in \mathbf{ACC}_i(kb'_i) \wedge (kb'_i, \mathbf{ACC}_i) \in mng_i(app_i(S', M), kb_i)\}$. Consider $S = (S_1, \ldots, S_n)$ such that for all $1 \leq i \leq n$ it holds that $S_i \in \mathbf{ACC}_i(kb'_i) \wedge (kb'_i, \mathbf{ACC}_i) \in mng_i(app_i(S', M), kb_i)$ and $S_i \cap OUT_i = S'_i$. Observe that such S_i exists by the definition of S'. It remains to show that for all $1 \leq i \leq n$ holds $S_i \in \mathbf{ACC}_i(kb'_i) \wedge (kb'_i, \mathbf{ACC}_i) \in mng_i(app_i(S, M), kb_i)$. Since $S_i \cap OUT_i = S'_i$, it holds that $S \models r$ iff $S' \models r$ for all $r \in br(M)$. Hence $app_i(S, M) = app_i(S', M)$ and the statement follows; consequently, it holds that $S \in \text{EQ}(M)$. $\square$

Generalising the notion of context complexity $\mathcal{CC}(M)$ from MCS to mMCS, let the *context complexity* of C_i be the complexity of the following problem:

$\mathcal{CC}_{mng}(M)$	**P**	$\mathbf{\Sigma_i^P}$	$\mathbf{\Delta_{i+1}^P}$	**PSPACE**	**EXPTIME**
$\mathcal{CONS}_{mng}(M)$	**NP**	$\mathbf{\Sigma_i^P}$	$\mathbf{\Sigma_{i+1}^P}$	**PSPACE**	**EXPTIME**

Table 5.1: Complexity $\mathcal{CONS}_{mng}(M)$ of recognising whether an mMCS M is consistent with respect to context complexity $\mathcal{CC}_{mng}(M)$, where $i \geq 1$ and entries denote membership.

($\mathcal{CC}_{mng}$) Decide, given a set O_i of operator statements and $S'_i \subseteq OUT_i$, whether there exist some $(kb'_i, \mathbf{ACC}_i) \in mng_i(O_i, kb_i)$ and $S_i \in \mathbf{ACC}_i(kb'_i)$ such that $S'_i = S_i \cap OUT_i$.

Here, C_i is explicitly represented by kb_i and br_i, and the logic suite is implicit, i.e., an oracle decides existence of S_i. The *context complexity* $\mathcal{CC}_{mng}(M)$ of an mMCS M is a (smallest) upper bound for the context complexity classes of all C_i. Depending on $\mathcal{CC}_{mng}(M)$, the complexity of consistency checking for some complexity classes is shown in Table 5.1, where entries denote membership results, resp. completeness results if CC is hard for some C_i, $1 \leq i \leq n$.

These results are all direct consequences of the complexity results of ordinary MCS, since we can use Proposition 5.6 to simulate an mMCS using an MCS. Observe that, given a managed context C_i and its corresponding ordinary context C'_i, the definition of the acceptability function of C'_i is equal to the notion of context complexity of C_i, hence the context complexity of C'_i is the same as for C_i. If $\mathcal{CC}_{mng}(M)$ is $\mathbf{C}$ for some complexity class $\mathbf{C}$, then deciding for given S'_i, O_i whether some $(kb'_i, \mathbf{ACC}_i) \in mng_i(O_i, kb_i)$ exists such that $S_i \in \mathbf{ACC}_i(kb'_i)$ and $S'_i = S_i \cap OUT_i$ hold is possible in $\mathbf{C}$, i.e., there exists an algorithm deciding this in $\mathbf{C}$.

We can use this to construct an algorithm realising the acceptability function of an ordinary context, whose complexity then is also in $\mathbf{C}$.

The following proposition shows that using Proposition 5.6 allows to translate an mMCS M into an MCS M' such that their output-projected equilibria are the same and the context complexity of both systems also is the same; effectively, this is a reduction from $\mathcal{CONS}_{mng}(M)$ to $\mathcal{CONS}(M')$.

Proposition 5.8. *The complexity of $\mathcal{CONS}_{mng}(M)$ in Table 5.1 for an mMCS M is the same as the complexity of $\mathcal{CONS}(M')$ for the corresponding MCS M'.*

Proof. Let $M = (C_1, \dots, C_n)$ be an mMCS and let $M' = (C'_1, \dots, C'_n)$ be the corresponding ordinary MCS. First observe, that the size of M' is the same as of M, since for any $1 \leq i \leq n$ holds that those elements of C_i that are explicitly represented, knowledge-base kb_i and bridge rules $br_i(M)$, are of the same size as those elements of C'_i that are explicitly represented, knowledge-base kb_i and bridge rules $br_i(M')$. This holds, because the knowledge base of C_i and C'_i is both times kb_i and the bridge rules of M' are the same as for M, except that the operational statements in the heads of $br_i(M)$ are wrapped by a new symbol in $br_i(M')$; hence $|br_i(M)| = |br_i(M')|$.

Consider for any $1 \leq i \leq n$, the managed context C_i and the corresponding ordinary context C'_i. Let $\mathcal{CC}_{mng}(M) \leq \mathbf{C}$ hold for some complexity class $\mathbf{C}$. Then given a set O_i of operational statements and an output-projected belief set S'_i, deciding whether some $(kb'_i, \mathbf{ACC}'_i) \in mng_i(O_i, kb_i)$ and $S_i \in \mathbf{ACC}'_i(kb'_i)$ exists such that $S'_i = S_i \cap OUT_i$, is in $\mathbf{C}$.

Likewise, for the context complexity $\mathcal{CC}(M')$ of an ordinary MCS it holds that $\mathcal{CC}(M')$ is the complexity of deciding whether for a given $H \subseteq \{\varphi(r) \mid r \in br_i(M')\}$ and an output-projected belief set S_i there exists $T_i \in \mathbf{ACC}_i(kb_i \cup H)$ such that $S_i \cap OUT_i = T_i$.

In the following, we assume that $\mathbf{C} \geq \mathbf{P}$. For deciding whether $T_i \in \mathbf{ACC}_i(kb_i \cup H)$ holds, we recall the definition of $\mathbf{ACC}_i$:

$$\mathbf{ACC}_i(kb) = \{S \cap OUT_i \mid \exists (kb', \mathbf{ACC}') \in mng_i(\{o \mid ns(o) \in kb\}, kb \setminus ns(\Phi^{OP}_{LS})) : \\ S \in \mathbf{ACC}'(kb')\}$$

Note that the computation of $O_i = \{o \mid ns(o) \in kb\}$ and $kb_i = kb \setminus ns(\Phi^{OP}_{LS})$ is possible in time linear in the size of kb, while the decision whether there exists $(kb', \mathbf{ACC}') \in mng_i(H, kb_i)$ such that $S \in \mathbf{ACC}'(kb')$ is possible in $\mathbf{C}$, because this decision is exactly the context complexity of C_i in the mMCS M, and $\mathcal{CC}_{mng}(M) \leq \mathbf{C}$. Hence, the complexity of deciding whether $T_i \in \mathbf{ACC}_i(kb_i \cup H)$ holds is $\mathbf{C}$. Since this holds for all $1 \leq i \leq n$, it holds that the context complexity $\mathcal{CC}(M') \leq \mathbf{C}$; consequently $\mathcal{CC}(M') = \mathcal{CC}_{mng}(M)$.

Since $\mathcal{CC}(M') = \mathcal{CC}_{mng}(M)$ for $\mathcal{CC}_{mng}(M) \geq \mathbf{P}$, and by Proposition 5.6 it holds that $S \in \mathrm{EQ}^o(M)$ holds iff $S \in \mathrm{EQ}(M')$ holds, and by Proposition 5.7 it holds that $S \in \mathrm{EQ}^o(M)$ holds iff there exists an equilibrium $T \in \mathrm{EQ}(M)$ such that $S = T^o$, it holds that deciding $\mathcal{CONS}_{mng}(M)$ is possibly by deciding $\mathcal{CONS}(M')$. Hence, we have a polynomial-time reduction from $\mathcal{CONS}_{mng}(M)$ to $\mathcal{CONS}(M)$. □

Using simple insert/delete management, an example of $\mathcal{CC}_{mng}(M)$ in $\mathbf{P}$ would be an mMCS built on defeasible logic (cf. [101]), and one for $\mathbf{NP}$ (resp., $\mathbf{\Sigma^P_2}$) using normal (disjunctive) answer set programs. Argumentation context systems [30] provide examples of mMCS with context complexity in $\mathbf{\Delta^P_3}$; examples for $\mathbf{PSPACE}$ and $\mathbf{EXPTIME}$ can be found, e.g., among modal and description logics. Such contexts also have respective hard instances.

Problem $\mathcal{CC}_{mng}$ intuitively consists of two subproblems: (MC) compute some $(kb'_i, \mathbf{ACC}_i) \in mng_i(O_i, kb_i)$ and (EC) decide whether $S_i \in \mathbf{ACC}_i(kb'_i)$ exists s.t. $S'_i = S_i \cap OUT_i$. However, it makes sense to analyse consistency depending on $\mathcal{CC}_{mng}$: often MC is solvable in polynomial time (perhaps non-deterministically and/or with the help of an oracle) or polynomial space, but kb'_i may become exponentially large (e.g., using a KB update or revision operator), nevertheless its explicit construction is avoidable for solving EC. If the output of MC remains polynomial, then the complexity of $\mathcal{CC}_{mng}$ can suitably be characterised in terms of MC (e.g., maximal consistent subsets of a propositional theory) and EC. We leave more detailed results for future work.

5.7 Summary

In this chapter we presented managed Multi-Context Systems (mMCS), where contexts are enhanced by a context manager which allows that the knowledge base of the context is modified depending on applicable bridge rules and a semantics for reasoning is selected. Context managers can apply belief revision for classical logics, update logic programs according to some update operators, ensure that the context is consistent, and much more. Thus they allow that legacy methods of managing inconsistency are brought to the Multi-Context Systems framework in such a way that each context of an mMCS uses the best fitting method of local inconsistency management.

We gave some sample instantiations of mMCS showing that they can capture operations on relational databases, belief revision, logic program updates, and the framework of argumentation context systems. The main issue addressed, however, is the influence of context managers that ensure the existence of locally acceptable belief sets (i.e., managers that guarantee the consistency of a context). Most importantly, it turns out that acyclic mMCS using such context managers are always consistent and for cyclic mMCS it holds that the source of inconsistency always is some cyclic information flow.

An investigation of the expressiveness of mMCS showed that they are not more expressive than MCS, since we showed how to translate an mMCS into an MCS and vice versa. This is in line with our results on computational complexity, which show that deciding whether a given mMCS has an equilibrium is of the same complexity as deciding whether the corresponding MCS has an equilibrium.

CHAPTER 6

Related Work

Non-monotonicity in MCS was introduced in [114] and then further developed in [29, 35] to eventually allow heterogeneous as well as nonmonotonic systems, and in particular nonmonotonic MCS [29] as considered in this thesis (cf. [31] for a more comprehensive account of work related to MCS). However, issues arising from inconsistency of such systems have been largely disregarded.

Inconsistency in MCS

A remarkable exception, and thus most closely related to ours, is [18, 19], where inconsistency in a homogeneous MCS setting is addressed. The approach is to consider defeasible bridge rules for inconsistency removal, i.e., a rule is applicable only if its conclusion does not cause inconsistency. This concept is described in terms of an argumentation semantics in [17]. The decision which bridge rules to ignore is based, for every context, on a *strict total order* of all contexts. The set of rules that are ignored thus corresponds to a unique deletion-only diagnosis whose declarative description is more involved compared to our notion, but which is polynomially computable. Note however, that the second component of diagnoses, i.e., rules that are forced to be applicable, have no counterpart in the defeasible MCS inconsistency management approach. Furthermore, the strict total order over contexts forces the user to make (perhaps unwanted) decisions at design time; alternative orders would require a redesign and separate evaluation. Our approach avoids this and can be refined to respect various kinds of orderings and preferences (cf. Chapter 4). Furthermore, the management component of mMCS can also enfore a strict total order when importing information from other contexts, i.e., if for such a managed context two bridge rules with conflicting head formulas are applicable, then the context manager only adds the formula of the strictly preferred context. A more detailed investigation, however, is required to determine in how far this is sufficient to capture the approach of [19].

The framework of managed Multi-Context Systems has been used as a basis for systems which take time into account. In [76] *evolving Multi-Context Systems* (eMCS) are introduced where discrete time steps are considered and distinct by sets of observations at each step. There are two types of contexts in an eMCS, ordinary ones like in mMCS and observation contexts, whose knowledge bases change at each time step. The underlying idea is that an observation context represents some kind of sensor whose data is used by the other contexts. Furthermore, there are two types of bridge rules, one whose operational statement is considered only for the current time step, and one which has a permanent effect. The semantics of eMCS is given in terms of sequences of belief states under a given sequence of observations: the sequence of belief states is incremental in the sense that each belief state takes the current observations and current bridge rules into account as well as all permanent bridge rules from belief states earlier in that sequence.

In [77] eMCS are further extended by evolving bridge rules resulting in an MCS framework called beMCS. In beMCS observation sequences not only effect the knowledge bases of certain contexts, but also the bridge rules of the ordinary contexts. Depending on these observations, some bridge rules are then disabled following some update semantics similar to that of logic programming updates. Some of the notions and results of inconsistency management in mMCS also carry over to eMCS and beMCS. Most notably it holds for a beMCS (eMCS) with totally coherent contexts, that it is consistent if it is acyclic, i.e., it admits an evolving equilibrium.

Reactive Multi-Context Systems (rMCS) (cf. [28,33,34,61]) have been developed in parallel to eMCS. They capture sensor readings and discrete time steps similarly as eMCS, although sensors are technically not represented as contexts. They behave like observation contexts in eMCS. Reactive MCS are also based on mMCS and their semantics is given in terms of a so-called run, which again is very similar to an evolving equilibrium. It is also shown how contradicting sensor data can be treated by timestamping each observation and using the management function of contexts for inconsistency management. The advantage of such an approach is that various tailored management functions can be used at the same time to resolve inconsistent sensor data. Nevertheless, rMCS can address issues which cannot be addressed by (static) mMCS, e.g., short response times to emergencies.

Addressing inconsistency in rMCS and eMCS, one can lift the notion of diagnosis to a sequence of observations, where for a sequence of length n, a lifted diagnosis is a sequence of n ordinary diagnoses, one for every time step, such that every of these steps admits a (static) equilibrium, i.e., the whole sequence of observations admits a run. Since the belief states at different time steps possibly depend on each other, further conditions may account for identifying preferred or minimal diagnoses. Intuitively, given a sequence of observations, the (seemingly superflous) removal of a bridge rule at time t might allow to keep two other bridge rules at time $t + 1$. So, diagnoses for rMCS and eMCS can be minimal with respect to each time step, or minimal with respect to a whole sequence of observations. Intuitively, the latter seems to be the better choice since rMCS and eMCS deliberately consider inductively built semantics while the former kind of minimality ignores that time steps occur one after another and that beliefs of an earlier step can influence a latter one. Regarding explanations, it seems possible to define an explanation dual to the concept of diagnosis of the latter kind (considering the inductive influence of time steps). More work on this is required, but rMCS and eMCS are outside the scope of this thesis.

The study of equilibria in Multi-Context Systems recently has been continued in [124] where the notions of grounded equilibrium and (ordinary) equilibrium are generalised to *supported equilibrium semantics*. Various strengths of supports are possible to obtain a range of equilibrium semantics; grounded equilibria and ordinary equilibria then are instances of certain supported equilbria. Of special interest here is the notion of support which in principle enables a notion of diagnosis that also considers modifications of knowledge bases to restore global consistency. Due to the recency of this work, the notion of support has not been addressed in this thesis and the topic remains for future work.

A stream-based approach at the semantics of MCS is taken in [62] where so-called asynchronous Multi-Context Systems (aMCS) are introduced. In aMCS each context is assigned input- and output-streams which contain computed pieces of information. Similar to rMCS (and eMCS) discrete time steps are considered, some contexts are designated for sensory input, and the semantics of such a system is given in terms of a run, which describes the states of the system. Different from rMCS and eMCS, however, a run in aMCS allows for partially computed belief sets and information to be exchanged before a full belief set (or full belief state) is computed.

Another formalism for homogenous contextualized reasoning that incorporates a form of inconsistency tolerance is the *Contextualized Knowledge Repository* (CKR) approach [119]. It is similar to the MCS approach of formalizing context-dependent knowledge, i.e., a CKR is a set of contexts where each context is a description logic. Contexts are assigned a vector of dimensional values that specify which kind of knowledge is contained in the context, e.g. $\{location = Italy, time = 2014\}$ states that the knowledge contained in the context is about Italy in the year 2014. Another context with broader knowledge on Europe in 2014 may be assigned the vector $\{location = Europe, time = 2014\}$. When MCS use bridge rules to refer to knowledge from another context, the CKR approach extends each description logic used in a context to allow direct referring to another context by a dimensional vector as qualifier. For example a DL concept for hot Italian cities might be $HotItalianCity \sqsubseteq HotRegion \sqcap City_{\{location=Italy\}}$. The semantics of CKR then ensures that the interpretation of individuals, concepts, and roles matches across contexts, i.e., a CKR model is a local model for each context (similar as a belief state in MCS is a belief set for each context) and the local models agree on the interpretation of common knowledge, which is different from MCS where belief sets need not agree on common symbols. It is furthermore possible to specify in a DL-like meta-language the coverage of topics among contexts, e.g., that the context about Europe is more general than the one about Italy. A CKR model also guarantees that the knowledge is then interpreted accordingly in both contexts.

A CKR is inconsistency tolerant in the sense that if some context is inconsistent (i.e., if its local model is the one with empty domain), then this inconsistency does not propagate to other unrelated contexts. The details are more intricate and given certain circumstances, inconsistency in one context may even turn the whole CKR inconsistent. In our terms, the inconsistency notion of CKR is more closely described by what we call local inconsistency in Chapter 5. In this sense, MCS are also inconsistency tolerant, since local inconsistency does not imply that no equilibrium exists. For example, consider a context C_i with an inconsistent knowledge-base using L^{pl}_{Σ} and $\Sigma = \{a\}$: C_i accepts the belief set $S = \{a, \neg a\}$ and this belief set occurs in an equilibrium of the MCS. So the MCS is not inconsistent while one of its contexts is (locally) inconsistent. In contrast to CKR, our approach also allows to restore consistency by modifying the interlinking of contexts.

Similar in vein to CKR systems are *Modular Ontologies*, i.e., a framework (cf. [63]) where description logic modules utilize and realize a set of interfaces in such a way that if a module is locally inconsistent, then by the notion of an "epistemic hole" it is ensured that the inconsistency does not spread to other modules. The interfaces are connected by bridge rules for Distributed Description Logic (DDL) [23]. Consistent query answering in a module is achieved by using the maximal consistent set of interfaces utilized by this module only, therefore whole interfaces will be ignored if they would cause any inconsistency in the module. Again, in addition to addressing a more general setting in terms of heterogeneity, our work considers potential modifications of bridge rules that allow to go beyond simple masking of inconsistent parts of the system in order to analyze inconsistency and potentially restore consistency.

Conceptually close to homogeneous forms of MCS are *Federated Databases*, a distributed formalism for autonomous, cooperating, linked databases [80]: each database has in addition to its local database schema an export schema describing which of its data may be used by other databases and an import schema describing what data from other databases it uses. Data is considered to describe "objects" (abstract or real-world entities), which can be exported and imported using a decentralized negotiation between two databases. Notably, [121] is a survey that, in addition to autonomy (access granting and revoking), is taking up on issues of heterogeneity, however mostly referring to the integration of different query languages of database systems. Global transaction management guarantees the consistency of the federated database, but due to the autonomy and concurrency of the employed database systems this task is possible only by enforcing serious restrictions. Existing approaches handle incoherence in a database-typical manner of cascading or rejecting local or distributed constraints. For instance, several protocols for global integrity constraint enforcement are presented in [79]. Since global transactions are unavailable without imposing serious restrictions, these protocols for constraint enforcement are defined upon quiescent states of the system, i.e., when it is at rest. For these states, designated constraint managers at each database ensure that no constraints are violated by querying each other according to the protocol and the constraints. Hence, inconsistency in federated databases is addressed at the level of the (individual) databases rather than their interlinking. Even though resorting to SQL and stratified Datalog allows for non-monotonicity, the possibility of instability in a distributed database system—due to cyclic dependencies—has not been addressed in the literature. Our work would be suitable to deal with such situations, given that federated databases can be described as MCS with stratified (mostly monotonic) contexts including constraints, and with positive bridge rules. Ordinary MCS on the other hand, cannot deal with the iterative nature of federated databases, but reactive or evolving MCS could.

Concerning the complexity results we established for diagnoses of MCS, we remark that they are related to respective results in abduction: by associating abducible hypotheses with bridge rules, due to the non-monotonicity of the system, recognition of diagnoses corresponds to *cancellation abduction problems*. The latter have been shown to be $\mathbf{NP}$-complete in [36] under the assumption of a tractable underlying theory (i.e., for $\mathbf{P}$ contexts in our terminology).

In [128] simplified MCS are presented as important for practical matter. In such MCS all bridge rules are of the form $(j:a) \leftarrow (i:a)$; such MCS are less expressive, as complex rule bodies and negation (thus nonmonotonic information flow) are not supported. In the presence of context managers, i.e., in mMCS, an analogous restriction does not impair expressivity: using, e.g., bridge rules $(j:\mathit{beliefs}(i,a)) \leftarrow (i:a)$. with designated operations *beliefs*, the management function mng_j can emulate any bridge rule r by $mng_j(O, kb_j) = \{(kb_j \cup H, \mathbf{ACC}_j)\}$ where

$H = \{\varphi(r) \mid \forall(i\,{:}\,p) \in \mathit{body}^+(r) : \mathit{beliefs}(i,p) \in O, \forall(i\,{:}\,p) \in \mathit{body}^-(r) : \mathit{beliefs}(i,p) \notin O\}$, i.e. mng_j adds the head formula $\varphi(r)$ if all positive literals of r are present in the form of their respective operational statement and all negative literals are absent. In fact, every MCS M can be easily transformed into such an mMCS M' having the same equilibria and all bridge rules being of the above form.

Broader Context

In a broader context, we have explored the relationship of our work to approaches and methods for inconsistency management in knowledge bases, grouped into debugging techniques (e.g., for Prolog [107, 108] and ASP [70, 102]), repairing methods (for instance based on abductive reasoning [84], discrimination among fusion rules [82], or policies for subquery propagation in peer-to-peer systems [11]), consistent query answering (e.g., over ontologies [90], propositional knowledge bases in peer-to-peer systems [20], etc.), and paraconsistent reasoning (applying, e.g., syntactic [15], logic-based [116], or domain-specific [66] methods).

We classify and discuss this literature according to the following basic approaches:

- *debugging* techniques serve the purpose of diagnosing information systems, aiming at identifying sources of unexpected and in most cases unintended computation outcomes, and at explaining the latter;
- *repairing* techniques modify the content of knowledge bases in order to restore consistency, in particular when new information is incorporated into a knowledge base, or when several knowledge bases are integrated into a single one;
- *consistent query answering* virtually repairs a knowledge base or system, often by ignoring a minimal subset of beliefs or subsystems, and operates on the resulting (virtual) consistent system (i.e., no knowledge is permanently removed);
- *paraconsistent reasoning* accepts contradictory knowledge and, rather than repairing or ignoring (parts of) the information, a more tolerant mode of reasoning is applied that handles also inconsistent pieces of knowledge in a non-trivial way.

Different from most approaches to inconsistency management, the aim of this thesis is not to provide a fixed set of methods for automatically restoring inconsistency, but to provide the following: a useful theoretical framework for analyzing inconsistency, methods to reason about inconsistency and its possible resolutions (using a user-selected formalism), and the incorporation of legacy inconsistency management techniques.

Debugging in Logic Programming

Debugging in logic programming, i.e., finding out why some logic program has no answer or has an unexpected answer, is remotely related to the problem considered in this thesis given that bridge rules look and behave similar to rules in logic programming. A major difference is that in MCS we take contexts with an opaque content into account. In logic programming, the presence of an atom in a model of a program directly depends on the firing of rules, which in turn directly depends on the presence or absence of other atoms in the bodies; in the MCS framework, which allows to capture arbitrary logics by abstract belief set functions, there is in general no visible link between the firing of bridge rules (adding information to a context) and beliefs accepted by the context.

Prolog Debugging

A framework for debugging Prolog programs was developed in [120]. It relies strongly on the operational specifics of Prolog and consists of a diagnosis and a bug-correction component, where three basic types of errors are considered: (i) termination with incorrect output, (ii) termination with missing output, and (iii) nontermination. For the latter, the approach identifies rules that behave unexpectedly by tracing procedure calls and querying the user whether the procedure call at hand of the form $\langle procedure, input, output\rangle$ is correct. A similar goal is achieved in [109], where the user should not tell whether such a triple is wrong, but point to a wrong subterm of a procedure call; for that, the implementation builds on a modified unification algorithm that keeps track of the origins of subterms. This is further refined in [106], where the different types of bugs are treated uniformly and by the use of a heuristics the number of questions to the user is reduced.

In comparison, our notion of inconsistency diagnosis roughly corresponds to type (i) and (ii) errors: in a diagnosis (D_1, D_2), D_1 contains bridge rules whose head belief is “incorrect”, while D_2 contains bridge rules whose head belief is “missing”. As for (iii), nontermination is not an issue for MCS since no infinite recursion can emerge (modulo computations inside contexts). Furthermore, our approach is fully declarative, without operational attachment adherent to Prolog, and it does not require user input; on the other hand, it only covers consistency and no further aspects. Nonetheless, it is possible to mimic behaviour under user input to some extent by using the meta-reasoning encoding of Chapter 4 such that the observation context enforces the user input, i.e., the user input is interpreted as a filter f and the filter-encoding M^f is used.

A purely declarative perspective on Prolog debugging is taken in [98], based on the formal semantics of extended programs under SLDNF resolution. Again two types of errors are considered, so called “wrong clause instances” (wrong solutions) and “uncovered atoms” (missing solutions). To pinpoint the origin of such errors, the user must specify the intended interpretation of the program, by repeatedly answering queries about the behaviour of the rules.

In [108] a connection between logic program debugging and abductive diagnosis is investigated. It considers extended logic programs (with strong and default negation) under closed-world assumption (CWA). Based on revisables, i.e., a subset R of the set of literals $not\, L$ assumed true by CWA, and the notion of supported sets $SS(L)$ of a literal L, the removal sets of L are defined as the hitting sets of $SS(L)$ restricted to R; the ones of the literal $\perp$ indicate how to obtain a non-contradictory program. Using a transformed program P_1 of P and information about wrong and missing solutions in P, so called minimal revising assumptions (MRAs) of P_1 are computed in an iterative manner which identify the reasons for wrong and missing solutions. For programs P that model diagnostic problems, minimal solutions can be obtained from the MRAs.

The ideas and notions in [98, 106] are merged in [107, 108] for normal logic programs with constraint rules under well-founded semantics. Referring to them, a diagnosis for a set U of literals (corresponding to a partially known desired model) is a pair $D = \langle Unc, InR\rangle$ where Unc are uncovered atoms and InR are incorrect rules of P, such that U is contained in the well-founded model (WFM) of the program P' that results from P by removing all incorrect rules and adding all uncovered atoms. In case of a single minimal diagnosis, the bug in the program is pinpointed precisely; otherwise, the user is asked which diagnosis corresponds to the intended interpretation. This leads to an iterative debugging algorithm that only asks disambiguating queries, i.e., it asks about a subset of the intended interpretation and adds the answer to U. Our notion of inconsistency diagnosis, where $D = (D_1, D_2)$ is a diagnosis iff $M[br(M) \setminus D_1 \cup cf(D_2)] \not\models \bot$ resembles this notion for $U = \emptyset$, since a diagnosis of an MCS only asks for the existence of an equilibrium and not for certain beliefs to be present or absent; the underlying semantics of MCS is however very different from WFM. Furthermore, there is no counterpart of our inconsistency explanations, nor have refined diagnoses been considered.

ASP Debugging

Answer-set Programming (ASP) is as a rule-based paradigm related to MCS, yet more under grounded equilibrium semantics, which imposes a minimality condition on equilibria [29]; in fact, answer-set programs can be modeled as particular MCS with monotonic rules and nonmonotonic bridge rules.

The declarative debugging of answer-set programs was approached by [123] for programs that have no cycles of odd length (where constraints are still allowed); in subsequent works, tagging [26], meta-programming for ground [70] and non-ground programs [102], and establishing procedural techniques (breakpoints, step-wise execution) [103] have been considered. The idea is that an expected answer-set E and an (erroneous) ASP program P are transformed into a program T whose answer-sets explain why E is not an answer-set of P. Explanations cover that an instantiation of some rule in P is not satisfied by E, as well as the presence of unfounded loops (i.e., lack of foundedness). The latter could be of interest for developing a diagnosis of MCS under grounded equilibria semantics; this remains for future work. On the other hand, the procedural techniques seem to be less promising, as MCS lack rule chaining due to context logics.

A different approach to debug answer-set programs is given in [7], where A-Prolog (an ASP-based language) is extended by *consistency-restoring* (CR) rules of the form

$$r: \quad h_1 \, or \ldots or \, h_k \stackrel{+}{\leftarrow} l_1, \ldots, l_m, not\, l_{m+1}, \ldots, not\, l_n.$$

which intuitively reads as: if $l_1, \ldots, l_m$ are accepted beliefs while $l_{m+1}, \ldots, l_n$ are not, then one of $h_1, \ldots, h_k$ “may possibly” be believed to remove inconsistency. In addition, a preference relation on the rules may be provided. The semantics of CR rules is defined via a translation to abductive logic programs, i.e., logic programs where certain atoms are abducibles (cf. [87]). In answer sets of such programs, a minimal set of abducibles may be assumed to be true without further justification.

Disregarding possible rule preferences, a logic program P with CR rules CR can be embedded to a MCS $M = (C_1)$, where the single context C_1 is over disjunctive logic programs, such that the answer sets of P with CR correspond to the witnessing equilibria of the minimal diagnoses (D_1, D_2) of M. In more detail, C_1 has the knowledge base $kb_1 = P \cup \{cr(r) \mid r \in CR\}$ and bridge rules $br_1 = \{(c_1 : ab(r)) \leftarrow \bot. \mid r \in CR\}$, where $ab(r)$ are fresh atoms, for each r as above, and

$$cr(r) = \quad h_1 \vee \ldots \vee h_k \leftarrow ab(r), l_1, \ldots, l_m, not\, l_{m+1}, \ldots, not\, l_n;$$

informally, unconditional firing of a bridge rule simulates the corresponding CR rule; note that $D_1 = \emptyset$.

Content-based Methods

The methods and approaches underlying research issues and works presented in this subsection exhibit more foundational differences to our notions of diagnosis and explanation. Therefore, we will mostly discuss them on a more general level, pointing to some seminal works and survey articles for more extensive coverage of the relevant literature. Note however, that the framework of managed Multi-Context Systems introduced in Chapter 5 enables MCS to employ the same inconsistency management methods local at each context that we describe in the sequel, i.e., every approach below can be used in an mMCS as a management function of a managed context employing a suitable logic.

Repair Approaches in Integrating Information

A lot of work on inconsistency management has been concentrating on the repair of data during merging, incorporating, or integrating data from different sources. In contrast to our work, in such approaches usually the mappings that relate data of different knowledge bases are fixed, while the contents of the knowledge bases are subject to change in order to restore consistency. This subsumes approaches that do not actually modify original data but modify it virtually (i.e., a view), or operate on a copy.

Belief revision and *belief merging* are well understood problems, in particular for classical propositional theories [88, 105]. They address how to incorporate a new belief into an existing knowledge base, respectively how to combine knowledge bases, such that the resulting knowledge base is consistent. In this regard, our approach is more related to belief merging than to belief revision. A major difference to belief merging is, however, that MCS connect heterogeneous knowledge bases in a decentralized fashion (compared to a centralized merge of uniform knowledge bases), and that selective information exchange among knowledge bases is possible via bridge rules in complex topologies. Furthermore, our work concentrates on changing the mappings between these components in case of conflict, while belief merging strives for modified contents (i.e., knowledge base). Nevertheless, mMCS allow the use of belief revision locally at suitable contexts to guarantee local consistency.

Abductive reasoning is often applied to identify pieces of information that need to be changed in order to repair a logical theory or knowledge base, cf. [84, 99, 127]. In particular, in [84] abduction is applied to repair theories in (nonmonotonic) logic based on notions of 'explanation' and 'anti-explanation'. Given an autoepistemic theory K and a set Γ of abducible formulas, one removes the formulas of a set $O \subseteq \Gamma$, and adds the formulas of a set $I \subseteq \Gamma$, to entail (resp. not entail) an observation F; i.e., $(K \cup I) \setminus O \models F$ (explanation), resp. $(K \cup I) \setminus O \not\models F$ (anti-explanation). A repair of an inconsistent theory K is given by an anti-explanation of $F = \bot$.

Given an inconsistent theory K and a set of hypotheses Γ, we can establish a correspondence between diagnoses using the following MCS M_K and anti-explanations for $\bot$. $M_K = (C_1)$ is an MCS whose only context $C_1 = (L^K, K \setminus \Gamma, br^K)$ is defined over a suitable (autoepistemic) logic $L^K = (\mathbf{KB}^K, \mathbf{BS}^K, \mathbf{ACC}^K)$ where $\mathbf{KB}^K$ is the set of all well-formed autoepistemic theories, $\mathbf{BS}^K$ is the set of all possible autoepistemic theories, and $\mathbf{ACC}^K$ maps each theory to the set of its stable and consistent expansions. The bridge rules br^K of C_1 are $br^K = \{(1\!:\!\phi) \leftarrow \top. \mid \phi \in K \cap \Gamma\} \cup \{(1\!:\!\phi) \leftarrow \bot. \mid \phi \in \Gamma \setminus K\}$, i.e., every $\phi \in K \cap \Gamma$ is added by a bridge rule and every $\phi \in \Gamma \setminus K$ is not added by a bridge rule. A diagnosis of M_K which removes a bridge rule of the former kind removes the corresponding formula ϕ from the theory K, and a bridge rule made condition-free is adding the respective formula.

Diagnoses and anti-explanations now correspond as follows. Let $D_1, D_2 \subseteq br(M_K)$ be such that for all $r \in D_1$ holds $\mathit{body}(r) = \top$ and for all $r \in D_2$ it holds that $\mathit{body}(r) = \bot$; then $(D_1, D_2) \in D^{\pm}(M_K)$ implies $(K \cup \{\varphi(r) \mid r \in D_2\}) \setminus \{\varphi(r) \mid r \in D_1\} \not\models \bot$, i.e., I, O with $I = \{\varphi(r) \mid r \in D_2\}$ and $O = \{\varphi(r) \mid r \in D_1\}$ is an anti-explanation of $\bot$ with respect to K. Furthermore, if $I \cap O = \emptyset$ and $(K \cup I) \setminus O \not\models \bot$ hold, then $(D_1, D_2) \in D^{\pm}(M_K)$ where $D_1 = \{r \in br^K \mid \varphi(r) \in O \wedge \mathit{body}(r) = \{\top\}\}$ and $D_2 = \{r \in br^K \mid \varphi(r) \in I \wedge \mathit{body}(r) = \{\bot\}\}$ hold, i.e., every anti-explanation corresponds to a diagnosis, given that the anti-explanation is not at the same time adding and removing the same formula $\phi \in \Gamma$.

As regards our notion of explanation for an inconsistent MCS, it has no counterpart in the approach of [84], since our explanations yield an inconsistent MCS while the notions of explanation and anti-explanation of [84] require that the resulting theory $(K \cup I) \setminus O$ is consistent, i.e., no theory K and sets I, O exist such that $(K \cup I) \setminus O$ is consistent and $(K \cup I) \setminus O \models \bot$ both hold.

Information integration approaches (see, e.g., [38, 45, 92, 93]) wrap several information sources and materialize the information into one global schema. Two main tasks are required to do so; the first is called schema matching to match the underlying schemas of the data (e.g., the attribute 'address' in one schema corresponds to the attribute 'location' in another). Schema matching is either rule-based (often using hand-crafted rules) or based on machine learning techniques to automatically deduce matches after sufficient training examples have been given. The second task of information integration then is to match the data, e.g., identify (relational) tuples from different sources that describe the same real-world entity.

Differences exist in whether the global schema is expressed as a view in terms of the local schemata (global-as-view approaches), or vice versa (local-as-view). The relevant relationships are represented as mappings, which often are specified by database queries; inconsistencies are resolved by modifying the materialized information, thus again by changing contents. However, since the original information sources are not altered, one might consider it closer in spirit to our approach than belief merging. Inconsistency management in information integration systems, and in particular the global-as-view approach, may be regarded as implicit change of mappings, by discarding tuples and/or generating missing tuples. Naturally, this corresponds to deactivating bridge rules and forcing bridge rules to fire, respectively. Different from MCS however, information integration approaches rely on hierarchical, acyclic system topologies. On the other hand, they apply a more expressive mapping formalism compared to bridge rules in MCS.

Peer-to-peer data integration systems [40] allow for a dynamically changing architecture of a data integration scenario in which peers can enter or leave the system anytime.

An automatic approach for reasoning with inconsistent knowledge in a peer-to-peer system is presented in [20] where peers use propositional languages but may utilize varying consequence relations like classical logic or nonmonotonic logic. Each peer is assigned a global preference value and each formula is assigned a rank according to the support of the formula, i.e., if knowledge from other peers is used, then the priority value of the most preferred peer supporting the formula gives an upper bound to the rank of the formula. Each formula and its support is used to construct an abstract argumentation framework (cf. [47]) whose preferred extensions designate those formulas that are "distributed entailed" by the system. Due to this, a formula ϕ and its negation $\neg\phi$ might both be entailed by the same system.

In principle, preference orders over diagnoses as introduced in Chapter 4 of an MCS can be used to simulate the ranking of formulas (occurring in the head of bridge rules), but due to the lack of a notion of support in contexts in general, this approach is limited to contexts where such a notion can be defined and successfully incorporated into the preference order on diagnoses. The witnessing equilibria of preferred diagnoses then would result in sets similar to the preferred extensions of the above argumentation framework.

Note that the distributed entailment is similar in nature to techniques of consistent query answering, and inconsistency handling in peer-to-peer systems often uses approaches similar to consistent query answering. We therefore discuss some peer-to-peer systems in the respective subsection below.

Ontology mapping [41] and the related tasks of ontology alignment, merging, and integration aim at (re-)using ontologies in a suitable combination. To this end, mappings between concepts, roles, and individuals are identified to denote the same entity or related entities in different ontologies. Automatic, statistical, and machine learning-based methods are used to 'discover' suitable mappings. They may introduce inconsistency in the (global or local) view on the resulting ontology, even if each individual ontology is consistent. Consistency is achieved by either fully disregarding a mapping if it would add an inconsistency, by preventing the spreading of inconsistency from one inconsistent ontology to another (cf. [24]), or by applying an evolutionary approach at modifying/mapping ontologies (cf. [100, 122]). Heterogeneity in ontology mapping usually refers to different nomenclatures prevailing in different ontologies, or to ontologies in different yet closely related formalisms (e.g., different description logics). In contrast, in MCS heterogeneity refers to combining systems based on different logical formalisms, which may be related by need not share any relationship in general.

Using context managers of mMCS, however, it is possible to employ ontology mapping and corresponding tools in mMCS to achieve the same mapping results between contexts based on ontologies, while at the same time these ontologies can interact with other contexts using formalisms totally different from ontologies. Also note that our approaches of preference-based inconsistency resolution allow a more fine-grained and specific resolution of inconsistency, given that mappings are expressed via bridge rules.

To summarize, the main difference between our work and the contributions to these rather diverse settings of information integrating — and in particular the issue of achieving integrity in doing so — is that our approach allows to locally re-use these approaches using context managers, but on a global perspective our notions of diagnosis and explanation consider modifying the 'mapping', i.e., the interlinking, rather than the data. While the importance of maintaining and repairing mappings has been recognized [45], major breakthroughs are still missing.

Consistent Query Answering

The approaches considered in this section do not actually modify data to repair an inconsistent system, but virtually consider possible repairs in order to return consistent answers to queries. As this includes (partial) ignorance of information (and thus inconsistency) for the sake of reasoning on a consistent system, the approaches may be regarded as in between repairing and paraconsistent reasoning.

The term *consistent query answering (CQA)* has been coined in the database area where various settings (wrt. integrity constraints and operations for repair) have been considered [3, 12, 13]. In general, the consistent answers to a query are those that result from every possible repair of the inconsistent database, where a repair is a minimal modification of the database such that all constraints are satisfied (cf. [13]). Given a database instance D (i.e., a set of tuples) over some schema $\mathcal{S}$ that violates a set of integrity constraints IC, a repair is a database instance D' over $\mathcal{S}$ that satisfies IC and which makes $\Delta(D, D') = (D \setminus D') \cup (D' \setminus D)$ (the symmetric set difference) minimal. The repair D' hence can be seen as a modification of D where some tuples are removed ($D \setminus D'$) and some tuples are added ($D' \setminus D$), which is comparable to the minimal diagnosis of an inconsistent MCS.

Indeed, let D be a database instance over $\mathcal{S}$ and let IC be some integrity constraints. We design a corresponding MCS $M^{D,IC} = (C_1)$ as follows: $L_1 = (\mathbf{KB}_1, \mathbf{BS}_1, \mathbf{ACC}_1)$ is a logic where $\mathbf{KB}_1$ contains all database instances over $\mathcal{S}$, $\mathbf{BS}_1 = \mathbf{KB}_1$, and $\mathbf{ACC}_1(kb) = \{kb\}$ holds iff kb satisfies all constraints of IC; the context $C_1 = (L_1, \emptyset, br_1)$ and every possible tuples over $\mathcal{S}$ appears in the head of some bridge rule of C_1, such that without modified bridge rules the tuples of D are added while all other tuples are not added. Formally, $br_1 = \{(1\colon a) \leftarrow \top. \mid a \in D\} \cup \{(1\colon a) \leftarrow \bot. \mid a \notin D, a \in \bigcup \mathbf{KB}_1\}$. Then $M^{D,IC}$ has an equilibrium $S = (D)$ iff D satisfies all integrity constraints of IC. Furthermore, every minimal diagnosis $(D_1, D_2) \in D^{\pm}_m(M^{D,IC})$ corresponds to a repair D' of D with $D' = D \setminus \{\varphi(r) \mid r \in D_1\} \cup \{\varphi(r) \mid r \in D_2\}$.

Further note that minimal deletion-diagnosis introduced in Chapter 3 correspond to a form of repairs where only the removal of tuples is considered. Notably, this kind of repairs is sufficient for the case of denial constraints (including key constraints, functional dependencies, etc.), it is sufficient to restrict the attention to tuple deletions for obtaining repairs and answering queries consistently.

Consistent query answering on top of equilibria admitted by minimal diagnoses then is possible, but beyond the scope of this work. On the other hand, context managers of mMCS may directly realize the computation of repairs. Selecting the consistent query answers, however, then requires skeptical reasoning over all knowledge bases returned by the context manager while the semantics introduced for mMCS is geared towards credulous reasoning, i.e., a belief state $(S_1, \ldots, S_n)$ is an equilibrium of an mMCS if for every $1 \leq i \leq n$ there *exists some* knowledge base returned by the context manager that accepts the S_i, but for CQA it the requirement for an equilibrium would be that for every $1 \leq i \leq n$ and *all* knowledge bases returned by the context manager (i.e., for all repairs), S_i is accepted.

Irrespective of that, CQA can be used in mMCS if the context manager returns as modified knowledge base one which encodes all possible repairs. Also note that several notions of repairs have been introduced in CQA (cf. [13]), each defining repair as minimal modification but with different understanding of minimality (e.g., cardinality-based minimality versus subset-based minimality) or modification (e.g., change of attribute values). Using suitable context managers and acceptability functions these techniques may be employed in mMCS.

In general, CQA might be regarded as an approach that automatically applies minimal (deletion-)diagnoses to suppress inconsistent information for answering queries over inconsistent relational databases. Despite this similarity to our work, the differences apart from heterogeneity are that diagnoses and explanations address the interlinking of knowledge bases rather than their content and they aim at making inconsistencies amenable to analysis, explicitly hinting at problems that should be investigated, rather than treating them implicitly for the sake of providing consistent answers.

CQA techniques have also been extended to description logic ontologies, e.g., in [90, 91], where the taxonomy part (TBox) is considered to be consistent but the data part (ABox) may possibly be inconsistent. Consistent answers to queries are then obtained on maximal consistent subsets of the data wrt. the taxonomy part (and potential further constraints).

Other approaches (but similar in nature) have been applied to answering queries in peer-to-peer data integration settings. The approach in [39] ignores inconsistent components and imports beliefs from other contexts only if they do not cause inconsistency; hence mappings between peers are changed such that only maximally consistent sets of beliefs are imported. Besides the conceptual difference to MCS regarding the system architecture (dynamic vs. static), our approach explains inconsistency by pointing out mappings that must be changed to achieve consistency. Furthermore, its does not aim at suggesting fixes to the system, and in particular not by ignoring entire contexts or beliefs held by a minority among them.

Paraconsistent Approaches

Paraconsistent reasoning approaches (see, e.g., [14, 83]) aim upfront at ignoring or tolerating inconsistency in knowledge bases, providing means to reason on them without knowledge explosion, i.e., without justifying arbitrary beliefs (*ex falso quodlibet*); thus, they do not focus on eliminating inconsistency. Nevertheless, in addition to keeping information systems operable in case of inconsistency, paraconsistent reasoning may, similar to our aim, also serve the purpose of analysing inconsistency.

Taking again a very general perspective, in particular disregarding heterogeneity and even the fact that our techniques apply to the interlinking of information, syntactic approaches such as [15] would be closest to our approach. The authors of [15] essentially restrict theories to the intersection of maximal consistent subsets of formulae as a basis for drawing paraconsistent conclusions. However, while minimal deletion-diagnoses might be viewed as corresponding to maximal consistent subsets, our approach does not prescribe a particular reasoning mode upon them (like considering the system obtained by their intersection). Moreover, our notions of diagnosis and explanation provide more fine-grained structures for analysis than just considering deletion diagnosis, and they deal with nonmonotonic behavior.

The methods that are applied in logic-based approaches to paraconsistent reasoning are completely orthogonal to our techniques. The most prominent representatives resort to *many-valued logics* in order to deal with inconsistency (cf. [9, 110]). The same applies to *paraconsistent logic programming* [21] (see e.g. [50] for more references and recent works), which therefore also are elusive from a detailed comparison. Nevertheless, developing model-based techniques for paraconsistent reasoning from inconsistent MCS is an interesting topic for future research. In this regard, [116] can be inspiring, where trust on information sources on the web has been modeled using an extension of Belnap's four-valued logic [9] and bridge-rule like constructions based on external predicates govern the information flow.

We conclude this section with a pointer to Gabbay and Hunter [67] who argued strongly for *managing inconsistency*, in contrast to avoiding, removing, or ignoring it. Their point is that an inconsistent system requires actions to be taken, and in order to do so, different issues must be respected that require a variety of methods. Notably they also developed a corresponding framework in a relational database setting [68, 69]. In this spirit, we consider the notions of diagnosis and inconsistency explanation for MCS as providing a foundational basis for developing methods for more specific tasks on top in order to manage inconsistency of the system. Employing our approaches at meta-reasoning on diagnoses of an inconsistent MCS and using context managers allows on the global and local level to analyse inconsistency and subsequently execute actions deemed sufficient for resolving inconsistency.

CHAPTER 7

Conclusion

In this work we have investigated inconsistency management in the Multi-Context Systems framework. We have addressed three main questions: how to identify the reasons of inconsistency, how to select most preferred resolutions of inconsistency, and how to enable in MCS the use of legacy solutions to inconsistency management locally at each employed formalisms.

To answer the first question (cf. Chapter 3), we have introduced the notion of diagnosis to characterize resolutions of inconsistency and explanations to identify the reasons of inconsistency and separate multiple inconsistencies. Diagnoses and explanations are pairs of sets of bridge rules indicating which bridge rules have to be removed and which have to be condition-free (such that they add information unconditionally). We also investigated refined notions where more fine-grained modifications to bridge rules are considered and have shown that the ordinary notions are sufficient to capture the refined notions.

Depending on the topology of the information exchange (so-called splitting sets) of an MCS, we showed that diagnoses and explanations of the whole MCS may be constructed by a combining diagnoses and explanations of parts of the MCS, i.e., we gave conditions that allow for a modular computation of diagnoses and explanations. Finally, we also gave an encoding in HEX (an extension of answer-set programming) to compute all explanations of inconsistency in an MCS.

The second question, i.e., selecting most preferred resolutions of inconsistency (i.e., diagnoses), was addressed in Chapter 4. Since it is impossible to decide for all MCS what comprises a preferred diagnosis, the preference on diagnoses for a given MCS must be expressed in some formalism. Our main contribution here is to not confine the user of an MCS to a specific preference formalism, but to develop a technique for meta-reasoning about diagnoses in MCS. By that, the user of an MCS may employ any formalism that can be used in an MCS to reason about preferred diagnoses. We presented several transformations and enhanced notions of diagnosis to allow the selection of most preferred diagnoses. Furthermore, we also presented transformations which allow the filtering of unwanted diagnoses, i.e., we not just support the comparison of diagnoses, but also the dismissal of diagnoses that fail requirements specified by the user of an MCS.

In the course of this, we have introduced two transformation-based approaches to meta-reasoning, where the first approach allows for bridge rules of the original MCS to be untouched in some cases and meta-reasoning is added on top of the existing MCS. The disadvantage of this approach is that meta-reasoning is imperfect, i.e., there are circumstances where two different diagnoses cannot be discerned. The second approach allows perfect observation but requires all bridge rules to be modified. These modifications are minimal and the original MCS can be reconstructed easily from the transformed one. We showed for the first approach to meta-reasoning that so-called deletion-parsimonious filters can be applied and according diagnoses can be selected correctly. Based on the second approach, we showed that filters in general can be selected correctly. In both cases, the notion of a diagnosis where some bridge rules are protected from modification is employed.

Using the second approach to meta-reasoning, we introduced two transformations to realize the selection of most preferred diagnoses. The first transformation, called "plain encoding", was shown to be correct for total preference orders using a further enhanced notion of diagnosis where some bridge rules are prioritized over the others. The second transformation, called "clone encoding", was shown to be correct for preference orders in general using another enhancement of the diagnosis notion, called mpm-diagnosis (i.e., subset-minimal prioritized-minimal diagnosis). Note that the plain encoding introduces exponentially many bridge rules while the clone encoding duplicates every context of the original MCS.

We also have investigated the computational complexity of the introduced notions: the complexity of subset-minimal diagnoses with protected bridge rules and prioritized diagnoses is the same as the complexity of ordinary subset-minimal diagnoses; the complexity of recognizing an mpm-diagnosis, however, is higher than the complexity of the aforementioned diagnoses, specifically it is one step up in the polynomial hierarchy. Nevertheless, we also showed that the complexity of selecting most-preferred diagnoses is already at the second level of the polynomial hierarchy for MCS where context complexity is polynomial-time and the given preference order also is polynomial-time. This shows that the clone encoding (which is linear in the size of the original MCS) combined with the notion of mpm-diagnosis indeed is worst-case optimal in this case.

Chapter 5 addressed the third main question, i.e., how to enable MCS to take advantage of existing inconsistency management techniques that have been developed for specific formalisms. To accomodate for such inconsistency managers, we extended the MCS framework such that each context is accompanied by a context manager resulting in managed Multi-Context Systems (mMCS). A context manager receives as input the heads of applicable bridge rules and the knowledge base of the context and it returns a modified knowledge base. Since the context manager may return any possible knowledge base, this allows any form of modification to be applied to the context. We have presented several useful sample instances, but the most interesting one is inconsistency management where the context manager applies belief revision to a classical logic, or logic programming updates. The context manager also allows to select a semantics of the context to switch between available semantics, e.g., a regular semantics and a paraconsistent one.

We have then investigated the effect of context managers that always guarantee the consistency of a context, i.e., the context manager of each context ensures that under every set of heads of bridge rules the context admits an acceptable belief set. We showed that for such mMCS consistency is ensured if the topology of the information exchange is acyclic (i.e., bridge rules form no cycle). We furthermore showed that for cyclic such mMCS it holds that the reasons of inconsistency always include a cycle and possibly some additional bridge rules that carry information towards the cycle.

Finally, we showed that mMCS and MCS have the same expressivity since each mMCS can be transformed into an equivalent MCS and vice versa. Using this, we could also show that the computational complexity of deciding whether an mMCS is consistent under certain necessary assumptions is the same as the complexity of deciding whether the corresponding MCS is consistent. Hence mMCS allow for an explicit application of (legacy) inconsistency management techniques tailored to specific formalisms while being not more complex than ordinary MCS.

7.1 Open Issues and Future Work

Several issues surrounding inconsistency in MCS have to remain open for this thesis.

Equilibrium Notions. In this thesis we considered an MCS to be inconsistent if it admits no equilibrium. In [29] the notion of grounded equilibrium is introduced as an alternate semantics of MCS and in [124] the concept is generalized to cover a range of eqilibrium notions, where grounded equilibria and equilibria as used within this thesis are just two specific cases. We did not address such notions of equilibria, but we suspect that many results of this thesis carry over to them. Our belief is based on the fact that our basic notions of diagnosis and explanations are built on whether there exists an equilibrium or not, so these notions can be readily extended to grounded equilibria or other forms of equilibria by just using the appropriate notion of equilibrium. On the other hand, it seems unlikely that such a change is without further interference, hence we consider a thorough investigation on the choice of equilibrium semantics and its influence on properties of diagnoses and explanations necessary.

Modifying Knowledge Bases. The notions of diagnosis and explanation consider only modifications of bridge rules while knowledge inside contexts is not modified. Locally modifying knowledge to ensure local consistency can be addressed by context managers as proposed in Chapter 5. These choices satisfy information hiding concerns, since no context is required to exhibit its private knowledge base to identify diagnoses or explanations. In recent work [124] the notions of support and justifications enable the tracing of reasons of inconsistency also through knowledge bases. Hence, it enables global diagnoses that also consider modifications to knowledge bases. Consider for example an inconsistent mMCS where cyclic information flow is the reason of some inconsistency: our notion of diagnosis either removes a bridge rule or makes a bridge rule condition-free to break the cycle, where a diagnosis based on support may also consider the addition of a new fact to one of the involved knowledge bases to stabilise the cycle.

Basic Notions. Regarding our basic notions, there remain several issues for future work. On the computational side, scalability to scenarios with larger data volume and number of bridge rules is desirable, where the intrinsic complexity of our diagnoses and explanations is prohibitive in general. It remains to single out settings where scalability is still possible, and to get a clearer picture of the scalability frontier. This is linked to the complexity of consistency checking for an MCS; restrictions on the interlinking, in numbers and structure (for the latter, see [6]) will be helpful, as well as properties of the context logics (e.g., monotonicity and unique accepted belief sets). Related to this is developing pragmatic variants of our notions, like focusing by protecting bridge rules (which does not increase worst case complexity), giving up properties (e.g., minimality), or by tolerating inconsistency in parts of the system.

Preferences. The clone encoding $M^{\preceq}$ for preferences is able to realise all preference orders, but it comes at the cost of cloning each context of the original MCS. As the plain encoding shows, this is not always necessary. It would be a big improvement to identify preferences whose realisation requires no cloning and also avoid the exponential increase in bridge rules incurred by the plain encoding. Whether such an encoding exists and if it can realise useful classes of preferences is an open question.

We investigated preferences among diagnoses in general, but the aspect of deducible information under a given diagnosis has not been investigated. For example, one could think of a preference like "the best diagnosis is one that leads to a maximum amount of knowledge in the resulting belief sets" (without turning all bridge rules condition-free, i.e., D_2 still is minimal). Such preference requires to consider the resulting equilibria, which our approach does not consider. Luckily, the meta-reasoning encoding can be extended to allow such preferences by simply adding further protected bridge rules that import all beliefs of the contexts to the observation context.

To realise preferences in general, we needed to introduce the notion of an mpm-diagnosis, whose computational complexity is higher than that of regular diagnoses (or diagnoses with protected bridge rules). Currently, there exists no implementation to compute mpm-diagnoses. The same is true for minimal diagnoses with protected bridge rules, but these should be easy to implement on top of the existing implementation to compute minimal diagnoses.

The presented meta-reasoning approaches are centralised and we showed for some filters, how the central observation context may be decomposed into multiple smaller ones depending on the actual filter that is realised. For the clone encoding, this decomposition is not readily possible since it requires some information flow between the decomposed contexts. Using protected bridge rules such information flow in theory is possible, but it is unclear if such a decomposition with respect to diagnoses with protected bridge rules is possible in general.

Managed MCS. An interesting issue for future work are refined semantics for mMCS to discriminate among equilibria, such as an extension of minimal or grounded equilibria [29] to mMCS. Moreover, as management functions may yield alternative knowledge bases, preference of equilibria may be based on preference of alternatives. In particular for consistency restoring, minimality of change seems natural.

Practical Applications. Computing diagnoses, explanations, or equilibria of an MCS is hard, i.e., it is $\mathbf{NP}$-hard and depending on context complexities it may be even harder. Similar $\mathbf{NP}$-hard problems like SAT or ASP resulted in efficient solvers that are able to handle (relatively) large instances while no such efficient solutions are known for MCS (although there has been impressive work for MCS in [5, 6, 49]). Intuitively, a major reason for the relative efficiency of SAT- and ASP-solvers is the fact that they analyse and learn the internal structure of a given instance, which allows them to cut away large portions of the search space that do not contribute to a possible solution. Since we assumed an information hiding regime for contexts, i.e., they do not exhibit their internal structure to the outside, similar approaches for equilibrium computation in MCS are not possible. It seems that giving up on information hiding enables much more efficient and practical algorithms for finding the equilibria of a given MCS and in consequence also for identifying diagnoses or explanations. The notion of support as in [124] might be useful to describe the internal structure of contexts in general terms to more efficiently find equilibria.

Bibliography

[1] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[2] José Júlio Alferes, Luís Moniz Pereira, Halina Przymusinska, and Teodor C. Przymusinski. LUPS—a language for updating logic programs. *Artif. Intell.*, 138(1-2):87–116, 2002.

[3] Marcelo Arenas, Leopoldo E. Bertossi, and Jan Chomicki. Answer sets for consistent query answering in inconsistent databases. *TPLP*, 3(4-5):393–424, 2003.

[4] Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, Cambridge, UK, 2003.

[5] Seif El-Din Bairakdar, Minh Dao-Tran, Thomas Eiter, Michael Fink, and Thomas Krennwallner. Decomposition of distributed nonmonotonic multi-context systems. In Janhunen and Niemelä [85], pages 24–37.

[6] Seif El-Din Bairakdar, Minh Dao-Tran, Thomas Eiter, Michael Fink, and Thomas Krennwallner. The DMCS solver for distributed nonmonotonic multi-context systems. In *European Conference on Logics in Artificial Intelligence (JELIA 2010)*, volume 6341, pages 352–355, 2010.

[7] Marcello Balduccini and Michael Gelfond. Logic programs with consistency-restoring rules. In Patrick Doherty, John McCarthy, and Mary-Anne Williams, editors, *Working Papers of the 2003 AAAI Spring Symposium on Logical Formalization of Commonsense Reasoning*, pages 9–18. AAAI Press, Menlo Park, California, 2003.

[8] Rosamaria Barilaro, Michael Fink, Francesco Ricca, and Giorgio Terracina. Towards query answering in relational multi-context systems. In Cabalar and Son [37], pages 168–173.

[9] N. D. Belnap. A useful four-valued logic. In G. Epstein and J. M. Dunn, editors, *Modern Uses of Multiple-Valued Logic*, pages 7–37. Reidel Publishing Company, Boston, 1977.

[10] C. Berge. *Hypergraphs*. Elsevier Science Publishers B.V. (North-Holland), Amsterdam, 1989.

[11] Philip A. Bernstein, Fausto Giunchiglia, Anastasios Kementsietsidis, John Mylopoulos, Luciano Serafini, and Ilya Zaihrayeu. Data Management for Peer-to-Peer Computing: A Vision. In *Proceedings of the Fifth International Workshop on the Web and Databases, WebDB 2002*, pages 89–94, 2002.

[12] Leopoldo Bertossi. Consistent query answering in databases. *SIGMOD Record*, 35(2):68–76, June 2006.

[13] Leopoldo Bertossi. Database repairing and consistent query answering. *Synthesis Lectures on Data Management*, 3(5):1–121, 2011.

[14] Leopoldo E. Bertossi, Anthony Hunter, and Torsten Schaub, editors. *Inconsistency Tolerance [result from a Dagstuhl seminar]*, volume 3300 of *Lecture Notes in Computer Science*. Springer, 2005.

[15] Philippe Besnard and Torsten Schaub. Signed systems for paraconsistent reasoning. *J. Autom. Reasoning*, 20(1):191–213, 1998.

[16] Antonis Bikakis and Grigoris Antoniou. Distributed defeasible contextual reasoning in ambient computing. In Emile H. L. Aarts, James L. Crowley, Boris E. R. de Ruyter, Heinz Gerhäuser, Alexander Pflaum, Janina Schmidt, and Reiner Wichert, editors, *Ambient Intelligence, European Conference, AmI 2008, Nuremberg, Germany, November 19-22, 2008. Proceedings*, volume 5355 of *Lecture Notes in Computer Science*, pages 308–325. Springer, 2008.

[17] Antonis Bikakis and Grigoris Antoniou. Defeasible contextual reasoning with arguments in ambient intelligence. *IEEE Trans. Knowl. Data Eng.*, 22(11):1492–1506, 2010.

[18] Antonis Bikakis, Grigoris Antoniou, and Panayiotis Hassapis. Alternative strategies for conflict resolution in multi-context systems. In Lazaros S. Iliadis, Ilias Maglogiannis, Grigorios Tsoumakas, Ioannis P. Vlahavas, and Max Bramer, editors, *Artificial Intelligence Applications and Innovations III, Proceedings of the 5TH IFIP Conference on Artificial Intelligence Applications and Innovations (AIAI'2009), April 23-25, 2009, Thessaloniki, Greece*, volume 296 of *IFIP Advances in Information and Communication Technology*, pages 31–40. Springer, 2009.

[19] Antonis Bikakis, Grigoris Antoniou, and Panayiotis Hassapis. Strategies for contextual reasoning with conflicts in ambient intelligence. *Knowl. Inf. Syst.*, 27(1):45–84, 2011.

[20] Arnold Binas and Sheila A. McIlraith. Peer-to-peer query answering with inconsistent knowledge. In *Proceedings of the 11th International Conference on Principles of Knowledge Representation and Reasoning*, pages 329–339, Sydney, Australia, September 16–19 2008.

[21] Howard A. Blair and V. S. Subrahmanian. Paraconsistent logic programming. *Theor. Comput. Sci.*, 68(2):135–154, 1989.

[22] Markus Bögl, Thomas Eiter, Michael Fink, and Peter Schüller. The MCS-IE system for explaining inconsistency in multi-context systems. In Tomi Janhunen and Ilkka Niemelä, editors, *12th European Conference on Logics in Artificial Intelligence (JELIA 2010)*, Lecture Notes in Artificial Intelligence, pages 356–359. Springer, September 2010.

[23] Alexander Borgida and Luciano Serafini. Distributed description logics: Assimilating information from peer sources. *J. Data Semantics*, 1:153–184, 2003.

[24] Paolo Bouquet, Fausto Giunchiglia, Frank van Harmelen, Luciano Serafini, and Heiner Stuckenschmidt. C-OWL: contextualizing ontologies. In Dieter Fensel, Katia P. Sycara, and John Mylopoulos, editors, *The Semantic Web - ISWC 2003, Second International Semantic Web Conference, Sanibel Island, FL, USA, October 20-23, 2003, Proceedings*, volume 2870 of *Lecture Notes in Computer Science*, pages 164–179. Springer, 2003.

[25] Craig Boutilier, Ronen I. Brafman, Carmel Domshlak, Holger H. Hoos, and David Poole. CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *J. Artif. Intell. Res. (JAIR)*, 21:135–191, 2004.

[26] Martin Brain, Martin Gebser, Jörg Pührer, Torsten Schaub, Hans Tompits, and Stefan Woltran. Debugging ASP programs by means of ASP. In *Logic Programming and Nonmonotonic Reasoning (LPNMR)*, pages 31–43, 2007.

[27] Gerhard Brewka. Preferred subtheories: An extended logical framework for default reasoning. In N. S. Sridharan, editor, *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI). Detroit, MI, USA, August 1989*, pages 1043–1048. Morgan Kaufmann, 1989.

[28] Gerhard Brewka. Towards reactive multi-context systems. In Cabalar and Son [37], pages 1–10.

[29] Gerhard Brewka and Thomas Eiter. Equilibria in heterogeneous nonmonotonic multi-context systems. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 385–390, 2007.

[30] Gerhard Brewka and Thomas Eiter. Argumentation context systems: A framework for abstract group argumentation. In Esra Erdem, Fangzhen Lin, and Torsten Schaub, editors, *Logic Programming and Nonmonotonic Reasoning, 10th International Conference, LPNMR 2009, Potsdam, Germany, September 14-18, 2009. Proceedings*, volume 5753 of *Lecture Notes in Computer Science*, pages 44–57. Springer, 2009.

[31] Gerhard Brewka, Thomas Eiter, and Michael Fink. Nonmonotonic multi-context systems: A flexible approach for integrating heterogeneous knowledge sources. In Marcello Balduccini and Tran Cao Son, editors, *Logic Programming, Knowledge Representation, and Nonmonotonic Reasoning*, volume 6565 of *Lecture Notes in Computer Science*, pages 233–258. Springer, 2011.

[32] Gerhard Brewka, Thomas Eiter, Michael Fink, and Antonius Weinzierl. Managed multi-context systems. In Toby Walsh, editor, *IJCAI*, pages 786–791. IJCAI/AAAI, 2011.

[33] Gerhard Brewka, Stefan Ellmauthaler, and Jörg Pührer. Multi-context systems for reactive reasoning in dynamic environments. In *21st European Conference on Artificial Intelligence (ECAI 2014)*, 2014.

[34] Gerhard Brewka, Stefan Ellmauthaler, and Jörg Pührer. Multi-context systems for reactive reasoning in dynamic environments. In *Proceedings of the International Workshop on Reactive Concepts in Knowledge Representation (ReactKnow 2014)*, pages 23–29, 2014.

[35] Gerhard Brewka, Floris Roelofsen, and Luciano Serafini. Contextual default reasoning. In Manuela M. Veloso, editor, *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pages 268–273, 2007.

[36] Tom Bylander, Dean Allemang, Michael C. Tanner, and John R. Josephson. The computational complexity of abduction. *Artificial Intelligence*, 49(1-3):25–60, 1991.

[37] Pedro Cabalar and Tran Cao Son, editors. *Logic Programming and Nonmonotonic Reasoning, 12th International Conference, LPNMR 2013, Corunna, Spain, September 15-19, 2013. Proceedings*, volume 8148 of *Lecture Notes in Computer Science*. Springer, 2013.

[38] Andrea Calì, Domenico Lembo, and Riccardo Rosati. Query rewriting and answering under constraints in data integration systems. In Georg Gottlob and Toby Walsh, editors, *IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15, 2003*, pages 16–21. Morgan Kaufmann, 2003.

[39] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Inconsistency tolerance in P2P data integration: An epistemic logic approach. *Information Systems*, 33(4-5):360–384, 2008.

[40] Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. Logical foundations of peer-to-peer data integration. In Catriel Beeri and Alin Deutsch, editors, *PODS*, pages 241–251. ACM, 2004.

[41] Namyoun Choi, Il-Yeol Song, and Hyoil Han. A survey on ontology mapping. *SIGMOD Rec.*, 35:34–41, September 2006.

[42] Evgeny Dantsin, Thomas Eiter, Georg Gottlob, and Andrei Voronkov. Complexity and expressive power of logic programming. *ACM Computing Surveys*, 33(3):374–425, 2001.

[43] Minh Dao-Tran, Thomas Eiter, Michael Fink, and Thomas Krennwallner. Distributed nonmonotonic multi-context systems. In Lin et al. [97], pages 60–70.

[44] James P. Delgrande and Wolfgang Faber, editors. *Logic Programming and Nonmonotonic Reasoning - 11th International Conference, LPNMR 2011, Vancouver, Canada, May 16-19, 2011. Proceedings*, volume 6645 of *Lecture Notes in Computer Science*. Springer, 2011.

[45] AnHai Doan and Alon Y. Halevy. Semantic integration research in the database community: A brief survey. *AI Magazine*, 26(1):83–94, 2005.

[46] Carmel Domshlak, Ronen I. Brafman, and Solomon Eyal Shimony. Preference-based configuration of web page content. In Bernhard Nebel, editor, *IJCAI*, pages 1451–1456. Morgan Kaufmann, 2001.

[47] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2):321–358, 1995.

[48] Thomas Eiter, Gerhard Brewka, Minh Dao-Tran, Michael Fink, Giovambattista Ianni, and Thomas Krennwallner. Combining nonmonotonic knowledge bases with external sources. In Silvio Ghilardi and Roberto Sebastiani, editors, *Frontiers of Combining Systems, 7th International Symposium, FroCoS 2009, Trento, Italy, September 16-18, 2009. Proceedings*, volume 5749 of *Lecture Notes in Computer Science*, pages 18–42. Springer, 2009.

[49] Thomas Eiter, Michael Fink, Giovambattista Ianni, Thomas Krennwallner, and Peter Schüller. Pushing efficient evaluation of HEX programs by modular decomposition. In James Delgrande and Wolfgang Faber, editors, *Logic Programming and Nonmonotonic Reasoning, 11th International Conference (LPNMR 2011)*, pages 93–106, May 2011. (selected for "Best Papers Track" of IJCAI 2011).

[50] Thomas Eiter, Michael Fink, and João Moura. Paracoherent Answer Set Programming. In Lin et al. [97], pages 486–496.

[51] Thomas Eiter, Michael Fink, and Peter Schüller. Approximations for explanations of inconsistency in partially known multi-context systems. In Delgrande and Faber [44], pages 107–119.

[52] Thomas Eiter, Michael Fink, Peter Schüller, and Antonius Weinzierl. Finding explanations of inconsistency in multi-context systems. In Lin et al. [97], pages 329–339.

[53] Thomas Eiter, Michael Fink, Peter Schüller, and Antonius Weinzierl. Finding explanations of inconsistency in nonmonotonic multi-context systems. Technical Report INFSYS RR-1843-12-09, Institut für Informationssysteme, TU Wien, 2012.

[54] Thomas Eiter, Michael Fink, Peter Schüller, and Antonius Weinzierl. Finding explanations of inconsistency in multi-context systems. *Artificial Intelligence*, 216(0):233–274, 2014.

[55] Thomas Eiter, Michael Fink, and Antonius Weinzierl. Preference-based inconsistency assessment in multi-context systems. In Janhunen and Niemelä [85], pages 143–155.

[56] Thomas Eiter and Georg Gottlob. On the computational cost of disjunctive logic programming: Propositional case. *Ann. Math. Artif. Intell.*, 15(3-4):289–323, 1995.

[57] Thomas Eiter, Giovambattista Ianni, and Thomas Krennwallner. Answer set programming: A primer. In Sergio Tessaris, Enrico Franconi, Thomas Eiter, Claudio Gutierrez, Siegfried Handschuh, Marie-Christine Rousset, and Renate A. Schmidt, editors, *Reasoning Web*, volume 5689 of *Lecture Notes in Computer Science*, pages 40–110. Springer, 2009.

[58] Thomas Eiter, Giovambattista Ianni, Roman Schindlauer, and Hans Tompits. A uniform integration of higher-order reasoning and external evaluations in answer-set programming. In Kaelbling and Saffiotti [86], pages 90–96.

[59] Thomas Eiter, Giovambattista Ianni, Roman Schindlauer, and Hans Tompits. Effective integration of declarative rules with external evaluations for semantic-web reasoning. In *European Semantic Web Conference (ESWC)*, pages 273–287, 2006.

[60] Thomas Eiter and Axel Polleres. Transforming co-np checks to answer set computation by meta-interpretation. In Francesco Buccafurri, editor, *APPIA-GULP-PRODE*, pages 410–421, 2003.

[61] Stefan Ellmauthaler. Generalizing multi-context systems for reactive stream reasoning applications. In Andrew V. Jones and Nicholas Ng, editors, *Proceedings of the 2013 Imperial College Computing Student Workshop (ICCSW 2013)*, OpenAccess Series in Informatics (OASIcs), pages 17–24. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, September 2013.

[62] Stefan Ellmauthaler and Jörg Pührer. Asynchronous multi-context systems. In Stefan Ellmauthaler and Jörg Pührer, editors, *Proceedings of the International Workshop on Reactive Concepts in Knowledge Representation (ReactKnow 2014)*, pages 31–37, 2014.

[63] Faezeh Ensan and Weichang Du. Aspects of inconsistency resolution in modular ontologies. In *Advances in Artificial Intelligence, 21st Conference of the Canadian Society for Computational Studies of Intelligence, Canadian AI 2008, Windsor, Canada, May 28-30, 2008, Proceedings*, volume 5032 of *LNCS*, pages 84–95. Springer, 2008.

[64] Wolfgang Faber, Nicola Leone, and Gerald Pfeifer. Recursive aggregates in disjunctive logic programs: Semantics and complexity. In *European Conference on Logics in Artificial Intelligence (JELIA)*, pages 200–212, 2004.

[65] Michael Fink, Lucantonio Ghionna, and Antonius Weinzierl. Relational information exchange and aggregation in multi-context systems. In Delgrande and Faber [44], pages 120–133.

[66] ACW Finkelstein, D Gabbay, A Hunter, J Kramer, and B Nuseibeh. Inconsistency handling in multiperspective specifications. *IEEE Transactions on Software Engineering*, 20:569–578, 1994.

[67] Dov Gabbay and Anthony Hunter. Making inconsistency respectable 1: A logical framework for inconsistency in reasoning. In *Foundations of Artificial Intelligence Research*, volume 535 of *LNCS*, pages 19–32, 1991.

[68] Dov M. Gabbay. Labelled deductive systems: A position paper. In J. Oikkonen and J. Väänänen, editors, *Proc. of Logic Colloquium '90, Helsinki, Finland, 15–22 July 1990*, volume 2 of *Lecture Notes in Logic*, pages 66–88. Springer-Verlag, Berlin, 1993.

[69] Dov M. Gabbay and Anthony Hunter. Making inconsistency respectable: Part 2 - meta-level handling of inconsistency. In *ECSQARU*, volume 747 of *LNCS*, pages 129–136. Springer, 1993.

[70] Martin Gebser, Jörg Pührer, Torsten Schaub, and Hans Tompits. A meta-programming technique for debugging answer-set programs. In Dieter Fox and Carla P. Gomes, editors, *AAAI*, pages 448–453. AAAI Press, 2008.

[71] Michael Gelfond and Vladimir Lifschitz. The stable model semantics for logic programming. In Kowalski and Bowen [89], pages 1070–1080.

[72] Michael Gelfond and Vladimir Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9(3/4):365–386, 1991.

[73] Chiara Ghidini and Fausto Giunchiglia. Local models semantics, or contextual reasoning=locality+compatibility. *Artif. Intell.*, 127(2):221–259, 2001.

[74] Fausto Giunchiglia and Luciano Serafini. Multilanguage hierarchical logics or: How we can do without modal logics. *Artif. Intell.*, 65(1):29–70, 1994.

[75] Judy Goldsmith, Jérôme Lang, Miroslaw Truszczynski, and Nic Wilson. The computational complexity of dominance and consistency in cp-nets. *J. Artif. Intell. Res. (JAIR)*, 33:403–432, 2008.

[76] Ricardo Gonçalves, Matthias Knorr, and João Leite. Evolving multi-context systems. In Torsten Schaub, Gerhard Friedrich, and Barry O'Sullivan, editors, *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014)*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 375–380. IOS Press, 2014.

[77] Ricardo Gonçalves, Matthias Knorr, and Jo ao Leite. Evolving bridge rules in evolving multi-context systems. In Nils Bulling, Leendert W. N. van der Torre, Serena Villata, Wojtek Jamroga, and Wamberto Vasconcelos, editors, *Computational Logic in Multi-Agent Systems - 15th International Workshop, CLIMA XV, Prague, Czech Republic, August 18-19, 2014. Proceedings*, volume 8624 of *Lecture Notes in Computer Science*, pages 52–69. Springer, 2014.

[78] Georg Gottlob. Complexity results for nonmonotonic logics. *Journal of Logic and Computation*, 2:397–425, 1992.

[79] Paul W. P. J. Grefen and Jennifer Widom. Integrity constraint checking in federated databases. In *Proceedings of the First IFCIS International Conference on Cooperative Information Systems (CoopIS'96), Brussels, Belgium, June 19-21, 1996.*, pages 38–47, 1996.

[80] Dennis Heimbigner and Dennis McLeod. A federated architecture for information management. *ACM Trans. Inf. Syst.*, 3(3):253–278, 1985.

[81] Pascal Hitzler, Markus Krötzsch, and Sebastian Rudolph. *Foundations of Semantic Web Technologies*. Chapman & Hall/CRC, 1st edition, 2009.

[82] A Hunter and R Summerton. Fusion rules for context-dependent aggregation of structured news reports. *Journal of Applied Non-Classical Logics*, 2004.

[83] Anthony Hunter. *Handbook of Defeasible Reasoning and Uncertain Information*, chapter Paraconsistent Logics, pages 11–36. Kluwer, 1996.

[84] Katsumi Inoue and Chiaki Sakama. Abductive framework for nonmonotonic theory change. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, IJCAI 95, Montréal Québec, Canada, August 20-25 1995, 2 Volumes*, pages 204–210, 1995.

[85] Tomi Janhunen and Ilkka Niemelä, editors. *Logics in Artificial Intelligence - 12th European Conference, JELIA 2010, Helsinki, Finland, September 13-15, 2010. Proceedings*, volume 6341 of *Lecture Notes in Computer Science*. Springer, 2010.

[86] Leslie Pack Kaelbling and Alessandro Saffiotti, editors. *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30-August 5, 2005*. Professional Book Center, 2005.

[87] Antonis C. Kakas, Robert A. Kowalski, and Francesca Toni. Abductive logic programming. *J. Log. Comput.*, 2(6):719–770, 1992.

[88] Sébastien Konieczny and Ramón Pino Pérez. Logic based merging. *J. Philosophical Logic*, 40(2):239–270, 2011.

[89] Robert A. Kowalski and Kenneth A. Bowen, editors. *Logic Programming, Proceedings of the Fifth International Conference and Symposium, Seattle, Washington, August 15-19, 1988 (2 Volumes)*. MIT Press, 1988.

[90] Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. Inconsistency-tolerant semantics for description logics. In Pascal Hitzler and Thomas Lukasiewicz, editors, *RR*, volume 6333 of *Lecture Notes in Computer Science*, pages 103–117. Springer, 2010.

[91] Domenico Lembo and Marco Ruzzi. Consistent query answering over description logic ontologies. In *Proc. Conference on Web Reasoning and Rule Systems*, volume 4524 of *LNCS*, pages 194–208. Springer, 2007.

[92] Maurizio Lenzerini. Data integration: A theoretical perspective. In Lucian Popa, Serge Abiteboul, and Phokion G. Kolaitis, editors, *PODS*, pages 233–246. ACM, 2002.

[93] Nicola Leone, Gianluigi Greco, Giovambattista Ianni, Vincenzino Lio, Giorgio Terracina, Thomas Eiter, Wolfgang Faber, Michael Fink, Georg Gottlob, Riccardo Rosati, Domenico Lembo, Maurizio Lenzerini, Marco Ruzzi, Edyta Kalka, Bartosz Nowicki, and Witold Staniszkis. The INFOMIX system for advanced integration of incomplete and inconsistent data. In *SIGMOD*, pages 915–917, 2005.

[94] Nicola Leone, Gerald Pfeifer, Wolfgang Faber, Thomas Eiter, Georg Gottlob, Simona Perri, and Francesco Scarcello. The DLV system for knowledge representation and reasoning. *ACM Trans. Comput. Log.*, 7(3):499–562, 2006.

[95] Nicola Leone, Riccardo Rosati, and Francesco Scarcello. Enhancing answer set planning. Technical Report DBAI-TR-2000-37, Institut für Informationssysteme – Abteilung Databanken und Artificial Intelligence, Technische Universität Wien, 2000.

[96] Vladimir Lifschitz and Hudson Turner. Splitting a logic program. In *International Conference on Logic Programming (ICLP)*, pages 23–37, 1994.

[97] Fangzhen Lin, Ulrike Sattler, and Miroslaw Truszczynski, editors. *Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference, KR 2010, Toronto, Ontario, Canada, May 9-13, 2010*. AAAI Press, 2010.

[98] J.W. Lloyd. Declarative error diagnosis. *New Generation Computing*, 5(2):133–154, 1987.

[99] Jorge Lobo and Carlos Uzcátegui. Abductive change operators. *Fundam. Inform.*, 27(4):385–411, 1996.

[100] Alexander Maedche, Boris Motik, Ljiljana Stojanovic, Rudi Studer, and Raphael Volz. Ontologies for enterprise knowledge management. *IEEE Intelligent Systems*, 18(2):26–33, 2003.

[101] Michael J. Maher. Propositional defeasible logic has linear complexity. *TPLP*, 1(6):691–711, 2001.

[102] Johannes Oetsch, Jörg Pührer, and Hans Tompits. Catching the ouroboros: On debugging non-ground answer-set programs. *TPLP*, 10(4-6):513–529, 2010.

[103] Johannes Oetsch, Jörg Pührer, and Hans Tompits. Stepping through an answer-set program. In Delgrande and Faber [44], pages 134–147.

[104] Christos H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[105] Pavlos Peppas. *Handbook of Knowledge Representation*, volume 3 of *Foundations of Artificial Intelligence*, chapter Belief Revision, pages 317 – 359. Elsevier, 2008.

[106] Luís Moniz Pereira and Miguel Calejo. A framework for prolog debugging. In Kowalski and Bowen [89], pages 481–495.

[107] Luís Moniz Pereira, Carlos Viegas Damásio, and José Júlio Alferes. Debugging by diagnosing assumptions. In Peter Fritszon, editor, *AADEBUG*, volume 749 of *Lecture Notes in Computer Science*, pages 58–74. Springer, 1993.

[108] Luís Moniz Pereira, Carlos Viegas Damásio, and José Júlio Alferes. Diagnosis and debugging as contradiction removal. In Luís Moniz Pereira and Anil Nerode, editors, *LPNMR*, pages 316–330. The MIT Press, 1993.

[109] Luís Moniz Pereira. Rational logic programming, 1986.

[110] Graham Priest. Reasoning about truth. *Artificial Intelligence*, 39(2):231 – 244, 1989.

[111] Teodor C. Przymusinski. Stable semantics for disjunctive programs. *New Generation Comput.*, 9(3/4):401–424, 1991.

[112] R Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81–132, 1980.

[113] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32:57–95, 1987.

[114] Floris Roelofsen and Luciano Serafini. Minimal and absent information in contexts. In Kaelbling and Saffiotti [86], pages 558–563.

[115] Chiaki Sakama and Katsumi Inoue. Paraconsistent stable semantics for extended disjunctive programs. *J. Log. Comput.*, 5(3):265–285, 1995.

[116] Simon Schenk. On the semantics of trust and caching in the semantic web. In Amit P.Sheth et al., editor, *International Semantic Web Conference*, volume 5318 of *Lecture Notes in Computer Science*, pages 533–549. Springer, 2008.

[117] Peter Schüller. *Inconsistency in Multi-Context Systems: Analysis and Efficient Evaluation*. PhD thesis, Vienna University of Technology, 2012.

[118] Peter Schüller and Antonius Weinzierl. Semantic reasoning with SPARQL in heterogeneous multi-context systems. In Camille Salinesi and Oscar Pastor, editors, *Advanced Information Systems Engineering Workshops - CAiSE 2011 International Workshops, London, UK, June 20-24, 2011. Proceedings*, volume 83 of *Lecture Notes in Business Information Processing*, pages 575–585. Springer, 2011.

[119] Luciano Serafini and Martin Homola. Contextualized knowledge repositories for the semantic web. *J. Web Sem.*, 12:64–87, 2012.

[120] Ehud Y. Shapiro. *Algorithmic Program Debugging*. MIT Press, 1983.

[121] Amit P. Sheth and James A. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Comput. Surv.*, 22(3):183–236, 1990.

[122] Ljiljana Stojanovic, Alexander Maedche, Boris Motik, and Nenad Stojanovic. User-driven ontology evolution management. In Asunción Gómez-Pérez and V. Richard Benjamins, editors, *Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, 13th International Conference, EKAW 2002, Siguenza, Spain, October 1-4, 2002, Proceedings*, volume 2473 of *Lecture Notes in Computer Science*, pages 285–300. Springer, 2002.

[123] Tommi Syrjänen. Debugging inconsistent answer set programs. In *International Workshop on Nonmonotonic Reasoning (NMR)*, pages 77–83, 2006.

[124] Shahab Tasharrofi and Eugenia Ternovska. Generalized multi-context systems. In Chitta Baral, Giuseppe De Giacomo, and Thomas Eiter, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference, KR 2014, Vienna, Austria, July 20-24, 2014*. AAAI Press, 2014.

[125] Allen Van Gelder, Kenneth A. Ross, and John S. Schlipf. The well-founded semantics for general logic programs. *Journal of the ACM*, 38(3):619–649, July 1991.

[126] Antonius Weinzierl. Comparing inconsistency resolutions in multi-context systems. In Daniel Lassiter and Marija Slavkovik, editors, *New Directions in Logic, Language and Computation - ESSLLI 2010 and ESSLLI 2011 Student Sessions. Selected Papers*, volume 7415 of *Lecture Notes in Computer Science*, pages 158–174. Springer, 2011.

[127] Yan Zhang and Yulin Ding. CTL model update for system modifications. *J. Artif. Intell. Res. (JAIR)*, 31:113–155, 2008.

[128] Yuting Zhao, Kewen Wang, Rodney Topor, Jeff Z. Pan, and Fausto Giunchiglia. Building heterogeneous multi-context systems by semantic bindings. Technical report, Ingegneria e Scienza dell'Informazione, University of Trento, 2009.

Antonius Weinzierl

Lebenslauf

Person

Name	Antonius Weinzierl
Nationalität	Deutsch

Studium

seit 2009	**Doktoratsstudium**, *Technische Universität Wien*.
2009	**Diplom in Informatik**, *Ludwig-Maximilians-Universität München*.

Forschung

seit 2012	**Universitätsassistent**, *Technische Universität Wien*.
2009–2012	**Projektassistent**, *Technische Universität Wien*.