TU WIEN Informatics

# On A priori Belief Updates in the Epistemic Analysis of Distributed Systems

DISSERTATION

zur Erlangung des akademischen Grades

**Doktor der Technischen Wissenschaften**

eingereicht von

**Giorgio Cignarale**
Matrikelnummer 12029357

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Prof. Ulrich Schmid
Zweitbetreuung: Dr. Roman Kuznets

Diese Dissertation haben begutachtet:

| | |
|---|---|
| Tuomas Tahko | Thomas Bolander |

Wien, 30. Juni 2025

Giorgio Cignarale

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

TU WIEN Informatics

# On A priori Belief Updates in the Epistemic Analysis of Distributed Systems

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

**Doktor der Technischen Wissenschaften**

by

**Giorgio Cignarale**
Registration Number 12029357

to the Faculty of Informatics

at the TU Wien

Advisor: Prof. Ulrich Schmid
Second advisor: Dr. Roman Kuznets

The dissertation has been reviewed by:

Tuomas Tahko

Thomas Bolander

Vienna, June 30, 2025

Giorgio Cignarale

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

# Erklärung zur Verfassung der Arbeit

Giorgio Cignarale

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.

Wien, 30. Juni 2025

Giorgio Cignarale

# Acknowledgements

The results presented in this thesis are the product of a diverse and interdisciplinary collaboration. Each member contributed from their respective areas of expertise: my supervisor, Prof. Ulrich Schmid, provided essential insights in distributed systems and their epistemic analysis; my co-supervisor, Roman Kuznets, contributed with his deep knowledge of epistemic logic; and Prof. Tuomas Tahko, whose philosophical expertise proved invaluable, graciously agreed to collaborate with us on this novel topic. In addition, two fellow PhD students (and dear friends) under Prof. Schmid's supervision, Thomas Schlögl and Hugo Rincon Galeana, contributed to specific results included in this thesis. These contributions were further enriched by numerous discussions with colleagues, mentors, and friends.

I would like to express my deepest gratitude to my supervisors, Ulrich Schmid and Roman Kuznets. Their unwavering support, patience, and guidance have been invaluable throughout this journey and this work would not have been possible without them.

I am also sincerely grateful to Hans van Ditmarsch for his mentorship and the many insightful and stimulating discussions we've had along the way.

I extend my thanks to the reviewers of this thesis, Prof. Tuomas Tahko and Prof. Thomas Bolander, for their thoughtful and constructive comments.

I would like to thank the LogiCS Doctoral College for supporting my research, and in particular, I am deeply grateful for the invaluable help and guidance of the late Anna Prianichnikova, her presence and support made a lasting impact on my journey.

I am also thankful to Prof. Agata Ciabattoni and Prof. Christian Fermüller for their support and encouragement during the final, and often most challenging, months of my PhD.

A special thanks goes to Traude Sommer, whose readiness to help in times of need meant more than words can express.

To all my colleagues (including the lord of lentils and the chicken killer guys), thank you for the joyful, memorable moments we shared. You made our office a place I genuinely looked forward to coming to each day.

A heartfelt thank you to Hugo, for his unwavering support and for treating me like a brother throughout this entire journey.

Ringrazio la mia famiglia per avermi dato le possibilità per intraprendere questo percorso e per il supporto costante. Ringrazio in particolare mio fratello Lorenzo per esserci sempre stato, con pazienza, affetto e supporto incondizionato.

I am deeply grateful to all the people I have met over the years, at TU Wien, conferences, and summer schools, for their time, insights, and the many lessons they have shared with me.

Over the course of these almost five years, I've been through three breakups, five changes of residence, some breakdowns here and there, a pandemic, a wrong surgery, lost twenty kilos then gained ten, started ten new hobbies and abandoned just as many. I've traveled, met new people, made friends, lost some — and, oh right, somewhere in the middle of all that, I did a PhD. I feel ten years older than when I started five years ago, and I really hope that's not a trend. I would never have made it through without all the people that accompanied me in this journey. I am deeply grateful to each and every one of you—for showing up, for supporting me, for being there in moments big and small. This achievement is not mine alone, and it's only right that I dedicate it to all of you. Thank you.

# Kurzfassung

Diese Dissertation befindet sich an der Schnittstelle von drei sehr unterschiedlichen Bereichen: Philosophie, epistemic logic und distributed systems. Logik und Philosophie teilen eine sehr lange Tradition, mindestens seit Aristoteles' Syllogismen. Während die Verbindung zwischen Logik und distributed systems jüngeren Datums ist, hat sich die epistemische Modellierung distributed systems als äußerst fruchtbar erwiesen – sowohl bei der Entwicklung verteilter Algorithmen als auch bei der Formalisierung verteilter Aufgaben. Philosophie und verteilte Systeme wurden bislang nur selten gemeinsam betrachtet, obwohl es – wenig überraschend – viele philosophische Fragestellungen gibt, die sich aus dem Bereich distributed systems ergeben, insbesondere in Bezug auf die Rollen der Handlungsträger (Agents) und deren (zugeschriebenen) epistemischen Zuständen.

Der Kern dieser Arbeit besteht darin, die drei Disziplinen miteinander zu verknüpfen, um erste Antworten auf die Frage zu geben: Welche Rolle spielt apriorisches Wissen in distributed systems? Diese Frage erfordert eine interdisziplinäre Herangehensweise: Verteilte Systeme bilden den Kontext, in dem die Frage gestellt wird, apriorisches Wissen ist ein philosophischer Begriff, und Wissen in distributed systems wird auf natürliche Weise durch epistemische Logik formalisiert. In dieser Arbeit behandeln wir diese und verwandte Fragestellungen aus der kombinierten Perspektive aller drei Fachbereiche.

Die Dissertation zeigt, dass die explizite Modellierung von apriorischem Wissen in distributed systems – analog zur Modellierung anderer epistemischer Begriffe – zu einer Reihe neuer Erkenntnisse führt: Wir bieten eine formale Darstellung, die eine aktuelle Interpretation der Unterscheidung zwischen apriorischem und aposteriorischem Wissen mit dem Entwurfsprozess verteilter Systeme in Verbindung bringt. Auf diesem philosophischen Fundament entwickeln wir eine Familie von Logiken, die einige der üblichen apriorischen Annahmen in distributed systems lockern. Schließlich stellen wir Werkzeuge zur Verfügung, mit denen sich apriorische Aktualisierungen mithilfe dynamischer epistemischer Logik durchführen lassen – wodurch sich das apriorische Wissen von Agenten dynamisch verändern lässt. Insbesondere führen wir zwei unterschiedliche Mechanismen zur apriorischen Aktualisierung ein: einen, der das Eingreifen des Systemdesigners repräsentiert, und einen anderen, der die Fähigkeit von Agenten in selbstadaptiven Systemen darstellt, sich autonom an unerwartetes Verhalten anzupassen.

Die Potenz dieses Ansatzes beschränkt sich jedoch nicht allein auf verteilte Systeme: diese Dissertation wirft auch grundlegende philosophische Fragen auf – etwa zu Ethik und Kreativität –, die eng mit Themen der erklärbaren Künstlichen Intelligenz verknüpft sind. Gleichzeitig fordert

sie die Entwicklung neuer logischer Werkzeuge heraus, die unser formales Verständnis von (Meta-)Schlussfolgerung schärfen können.

# Abstract

This thesis is at the intersection of three very different fields: philosophy, epistemic logic and distributed systems. Logic and philosophy share a very long tradition, at least since Aristotle's *syllogisms*. While the ties between logic and distributed systems are more recent, the epistemic modelling of distributed systems has proved to be extremely fruitful both in the design of distributed algorithms and in the formalization of distributed tasks. Philosophy and distributed systems have rarely been seen together, even though, unsurprisingly, there are many philosophical questions arising from the distributed field, especially related to the role of agency of artificial agents and of their (attributed) epistemic states. The core of this thesis is to relate the three fields in conjunction to give some initial answers to the question: *What is the role of a priori knowledge in distributed systems?* This question really requires the three fields combined to give an answer: distributed systems are the arena in which the question is posed, a priori knowledge is a philosophical notion and knowledge in distributed systems is formalized naturally by epistemic logic. In this thesis, we will address this and related questions from the combined perspective of the three fields. This thesis proves that explicitly modelling a priori knowledge in distributed systems provides a number of novel insights, as it happened for the modelling of other epistemic notions: indeed, we provide a formal account that relates a recent interpretation of the a priori vs. a posteriori distinction to the process of designing distributed systems. On this philosophical groundwork we build up a family of logics that relax some of the common a priori assumptions in distributed systems. Finally, we provide the tools to perform a priori updates using Dynamic Epistemic Logic, allowing to change the a priori knowledge of agents dynamically. In particular, we introduce two distinct a priori update mechanisms, one representing the intervention of the system designer, and the other representing the ability of agents in self-adapting systems to adapt autonomously to unexpected behavior. The fruitfulness of the approach however is not limited to distributed systems only: the inquiry raises some crucial philosophical questions related to e.g. ethics and creativity, deeply linked to explainable artificial intelligence topics; at the same time it urges the development of new logic tools that can sharpen our formal understanding of (meta-level) reasoning.

# Contents

CHAPTER 1

# Introduction and formal preliminaries

My PhD thesis has an interdisciplinary scope, involving three very different domains: distributed systems, logic, and philosophy. In a nutshell, its goal is to explore and formally specify the intricate interplay of the a posteriori knowledge obtained by the participating processes during the execution of a fault-tolerant distributed system and the a priori knowledge built into its implementation by the system designer, with the view of eventually using it as a basis for an envisioned epistemic reasoning framework for *self-adaptive and self-organizing* (SASO) systems[THvM+14, BG09], which are becoming increasingly popular nowadays.

A distributed system [Lyn96] is made up of a set of spatially distributed processes that can perform local computations and communicate by sending messages with each other. Solving problems in such systems, like reaching agreement on some decision to be taken, is difficult due to the inherent uncertainty of the processes regarding computation speeds, message delays and, in the case of fault-tolerant distributed systems, faults. For example, deterministically reaching agreement is impossible in asynchronous systems when just a single process *might* crashes [FLP85]. Recently, epistemic analysis has even been extended to multi-agent systems where agents might be byzantine faulty [LSP82]. Since byzantine faulty agents may behave arbitrarily and even collude to fail the system, creating unique challenges, formal reasoning becomes even more crucial here.

Epistemic logic [Hin62] is known as a powerful tool for modeling and analysis of such distributed systems, which view them as multi-agent systems. It allows to focus on the knowledge agents gather during an execution of the system, and eventually use that for taking actions (like sending messages), which in turn can change the epistemic status of other agents [FHMV95]. Dynamic Epistemic Logic (DEL)[vDvdHK07] even allows to formally express the epistemic change caused by such actions within the logic, namely, by means of *updating* the underlying (Kripke) model. Consequently, viewed from an agent-perspective, computing the effects of updates and *synthesizing* updates achieving some desired epistemic change are crucial for solving problems in a distributed fashion. Still, it is important to mention that all the existing work on epistemic reasoning focuses on knowledge that, viewed in philosophical terms, is a posteriori in that it is obtained by the agents during system execution.

What is essentially neglected in the epistemic reasoning, or rather, only considered indirectly, is the large amount of a priori knowledge that is built into the agents during the system design phase. In fact, the agents are executing protocols which have been implemented by the system designer based on a large number of implicit and explicit system assumptions like bounds on message delays and numbers and severeness of faults. Apart from being assumed a priori, these assumptions are also supposed to hold throughout the lifetime of the system, and effectively define the range of possible behaviors of the system. Unfortunately, however, modern distributed systems are characterized by a vast spectrum of possible behaviors, which oftentimes cannot even be exhaustively envisioned beforehand. Consequently, some system assumptions may turn out to be wrong, in the sense that the system designer might have overlooked certain possibilities or might have imposed inadequate restrictions. Obviously, the resulting system cannot be expected to reach its goal in this case, even if all the protocols are implemented correctly. Moreover, nothing is usually foreseen for an agent to react to such design mistakes, except to consider itself to be faulty, jeopardizing the system's correct behavior.

These problems and the trend for more and more complex and also larger applications have fueled the need for developing adaptive distributed systems. Besides straightforward solutions that allow processes to monitor their behaviors and adapt certain parameters (within some range) "on the fly", self-adaptive and self-organizing systems are becoming increasingly popular. SASO systems have some built-in self-reflection and self-adaption capabilities, which allows them to adapt (some of) the underlying system assumptions, initially provided by the system designer, autonomously during a run. Unfortunately, however, this is still done in a very ad hoc fashion today: we are not aware of any epistemic logic or other formal framework that would allow to reason rigorously about SASO systems.

The aim of this thesis is to provide a first formal foundation for reasoning about a priori knowledge within the context of distributed systems and, in particular, SASO systems. Given the conceptual novelty of the proposed approach, it was essential to identify a suitable philosophical foundation to support our intuitions regarding the role of a priori knowledge in distributed systems. Almost by chance, we encountered the recent work of Tahko [Tah08, Tah11] on the distinction between a priori and a posteriori knowledge. To our surprise, his philosophical insights closely aligned with the kind of reasoning that arises in distributed contexts. Building on this connection, we employed tools from epistemic logic and dynamic epistemic logic to formalize the notion of a priori knowledge.

It is evident that a connection exists between system assumptions, typically crafted by the system designer *prior to* the operational phase, and a priori knowledge. On the other hand, in the execution of a system, processes can exclusively acquire knowledge about facts occurring within the system, aligning with the philosophical concept of knowledge derived from experience. In a nutshell, the core separation between a priori and a posteriori in distributed systems is given by the fact that while the system assumptions forming the a priori knowledge determine the set of allowed runs in a system, the actual system execution determines the a posteriori knowledge gained by agents.

If the a posteriori experience falsifies some a priori system assumption, a bootstrapping cycle is triggered, wherein a priori assumptions can be modified on the basis of the information deduced from the mismatch between the two. Due to the fallibility of this a priori knowledge, we coined

the term *a priori belief* to acknowledge that there is no guarantee that such "knowledge" faithfully describes the actual world, i.e., it is factive in the usual sense of epistemic logic terminology. On the newly formed basis, the system execution continues until a new mismatch arises, triggering another iteration of the bootstrapping loop. This interaction can be formalized in epistemic logic by separating the set of system states considered possible by the agents, i.e., the underlying model, constituting the core of that agent's a priori beliefs, and the knowledge gained by the agents during system execution as their a posteriori knowledge. This separation is a mandatory prerequisite for the ultimate goal of my thesis, namely, providing the basic mechanisms needed for implementing Tahko's bootstrapping relation in the setting of SASO systems.

Needless to say, the overall aim is not to implement the full bootstrapping loop, which would also involve elaborate a priori reasoning, i.e. a reasoning process that leads to the formulation of new suitable system assumptions[1]. Rather, my targets are fundamental basic concepts and mechanisms that must be available for this purpose, and demonstrating their utility by implementing a relatively simple bootstrapping loop. The first major foundational issue is the question of how to "translate" the wealth of conceivable system assumptions into a corresponding model. This is particularly true for byzantine agents, but also for messages with finite delays, for example, which immediately lead to huge models. The problem becomes even worse when the usual strict dichotomy between *correct* and *faulty* agents is changed to multiple sets of rational agents that pursue different goals: Since an agent receiving a message does not know a priori to which sets the sender agent belongs to, it might not even be able to correctly interpret, i.e., decode, the message. Given the relevance of rational agents in blockchain systems and game-theory-related settings in general, I decided to incorporate a novel epistemic approach into my thesis that represents the (in)ability of an agent to interpret a message by means of interpretation functions.

The main part of my thesis is devoted to model update mechanisms that can be used for changing the model in which agents' reasoning takes place in the case a problem with the underlying a priori assumptions has been detected by an agent or by the system designer. I identified several different aspects of model update approaches, which need to be treated very differently. First, model updates can be either (i) *global*, i.e., viewed from the perspective of the system designer, who first invokes a global update synthesis method and then reboots the whole system with the updated model, or (ii) *local*, which represents the situation wherein an agent locally discovers a discrepancy between its a posteriori knowledge and a priori knowledge, and invokes a model update aiming at correcting the faulty system assumptions. A completely orthogonal aspect of model update mechanisms is their scope. In practice, one will usually prefer model updates that do not dramatically change the entire model just because of a relatively minor change in some system assumption. When using goal-formula-based update synthesis, for example, a crucial question is how to find a "minimal" goal formula for correcting some particular system assumption. In my thesis, I developed an update synthesis method with "minimal" side effects, in the sense that it preserves as much of the underlying model as possible, avoiding drastic and unwanted model changes.

[1]To give a glimpse of the problem, a priori reasoning is involved in the formulation of e.g. scientific hypotheses in the scientific inquiry. Discovering the exact laws of a priori reasoning amounts to discover how Einstein came up with the relativity theory. This task goes beyond the current thesis .

Another aspect of the model update mechanisms introduced in this thesis is their visibility to other agents. In this thesis, both in case of global and local updates, we model fully private updates where agents that are not directly involved in the updates are not even aware of the update taking place. As a result of these updates, the common knowledge of the Kripke model, if present before the update, is lost, as different agents may consider different underlying models. One of the key results of this thesis is a model update mechanism for fully private updates, simulating agents' self-correction upon detecting a mismatch between their a posteriori knowledge and a priori knowledge.

In general, my thesis extensively broadens the research conducted by the FWF ByzDEL project in the context of fault-tolerant distributed systems, providing the basic formalism and tools for modeling and analysis of autonomous SASO systems. Hence, (fault-tolerant) distributed systems are the main motivation behind this thesis, but its ultimate goal is to provide a logic formalization of a priori beliefs and of a priori belief updates. It will be left to future research to build on those in an attempt to actually close the bootstrapping loop by also adding diagnosis approaches and, in particular, a priori reasoning to the framework.

The remainder of the introduction provides the foundational background for this thesis, primarily synthesizing existing work in the areas of distributed systems, epistemic logic, and philosophy. It is organized as follows: Section 1.1 is an overview of the field of (fault-tolerant) distributed systems, with particular emphasis on heterogeneous distributed systems (Section 1.1.1), wherein agents follow different protocols, and adaptive and self-adaptive systems (Section 1.1.2), in which agents can react dynamically to changes of the system not necessarily foreseen in their protocols. In Section 1.2 we introduce the basic tools of epistemic logic (EL) (Section 1.2.1) and its application to the epistemic analysis of distributed systems. Section 1.2.4 introduces dynamic epistemic logic (DEL), which extends EL with update operations that modify dynamically the underlying model. We introduce variants of DEL crucial for the development of the thesis in Section 1.2.5. We discuss the philosophical account of the a priori vs. a posteriori distinction discussed by Tahko in Section 1.3. Finally, an overview of the thesis is offered in Section 1.4.

## 1.1 Distributed systems: an overview

Distributed systems [Lyn96, CDKB11] consist of multiple spatially separated computing nodes that execute their programs independently of each other. Nodes can exchange information by sending messages over a communication network or via shared memory. In the distributed computing literature, the active entities are called *processes* and are assumed to execute some protocols in the course of a run that constitutes a possible time-evolution of the system. Designing distributed systems is difficult due to the inherent uncertainty of any process about the actual computation speeds and message delays etc. The design of such a system inevitably involves the need to come up with a set of system assumptions (like message delay bounds and failure rates) that are assumed to hold throughout the lifetime of the system.

A crucial complication is that distributed systems are characterized by a vast spectrum of possible behaviors, which often cannot even be exhaustively envisioned beforehand. Especially challenging is the design of fault-tolerant distributed systems, which must also be able to cope with (some)

erroneous behavior of processes and communication. Special kinds of failures, such as byzantine failures [LSP82] where faulty agents may deviate arbitrarily from their protocols, are particularly problematic, as malicious agents may start to execute actions that jeopardize the original goals of the system. In particular, byzantine agents can communicate in erratic, arbitrary, or deceptive manner. Malicious faulty agents may have a "hidden agenda", in which case, instead of following the original commonly known protocol, a faulty agent (or a group of faulty agents) can execute actions (possibly in consort with each other) that jeopardize the original goals of the system.

Correct agents are typically not aware of the goals or the capabilities of such faulty agents. To guarantee a certain degree of efficiency or even solvability, such systems require even stronger system assumptions than fault-free systems. Without such restrictions, provably correct solutions for a distributed task do not exist. These assumptions must usually be commonly known by all agents, such as the basic communication mechanism, the protocol of all correct agents, the data encoding used in their messages, etc.[2,3]

### 1.1.1 Heterogeneous distributed systems

An additional source of uncertainty, which increases the spectrum of possible behavior of distributed systems, is the variety of protocols that can be executed by agents. Intuitively, a protocol can be generally defined as a function from local states to non-empty set of actions. In general, the epistemic analysis of distributed systems occurs in system executions generated by the (common) protocols of agents in a particular setting [BZM14]. Usually, simplified scenarios assume common knowledge of the protocols[4] among the agents, meaning that the set of possible states of the system is fully determined by all the possible combinations of all possible protocols executed by other agents. In byzantine fault-tolerant systems, this is overly complicated by the completely unrestricted behavior of faulty processes. Consequently, this assumption is not always viable, especially for larger distributed systems, wherein agents are assigned many different tasks, some of which might not be known by other agents. We refer to these systems as *heterogeneous distributed systems*. In such systems we usually assume agents to be partitioned into types, depending on the protocol they are executing. In heterogeneous distributed systems we can restrict the common knowledge of the protocols assumption to agents of the same type. All agents of the same type, which typically work towards the same goal, use the same protocol that is commonly known to all agents of this type. What is not generally known to an agent is the distribution of agents into types and the actual protocol of a type different from its own. In other words, agent $a$ generally does neither know the type nor the protocol of agent $b$.

All the communication in the system is governed by the protocols. Whereas all protocols must use the same basic communication mechanism and a common layering structure [MR02], i.e., (possibly non-synchronous) communication rounds, agents of different types generally communicate according to different protocol rules, data formats, encodings, etc. Communication actions

[2] Later on we will refer to the whole corpus of these common assumptions to as *a priori knowledge*.

[3] For the possible world semantics, this translates into the assumption of common knowledge of the model [Art20] (Section 1.2.2) , which enables the agents to compute epistemic states of other agents, a task necessary for a typical coordination problem like consensus [CnGM22].

[4] We will explore the epistemic implications of the common knowledge of the protocols throughout the thesis.

are triggered by preconditions that depend on the protocol of the agent's type. Consequently, the interpretation of each message depends on:

- the knowledge of the receiver about the type(s) the sender may belong to;
- the knowledge of the receiver about the communication protocol of this (these) type(s).

More formally, we consider a finite set of processes $\mathcal{A} = \{p_1, \ldots, p_n\}$ that communicate with each other by using a joint communication mechanism, such as, e.g., shared memory objects or point-to-point messages. Each process executes some protocol with a name (= type) taken from a commonly known set of names $\Pi$. However, no assumption is made about the types and actual protocols of the distinct agents $i$ and $j$ being identical or mutually known. All protocols are organized in a common, possibly non-synchronous communication round structure. We also require that the system has a common notion of time, represented by a directed set $T = \mathbb{N}$.

**Definition 1** (Heterogeneous distributed system)**.** *We say that a tuple* $\langle \Pi, \mathcal{A}, \mathcal{P}, C, T \rangle$ *is a* heterogeneous distributed system *iff*

- $\mathcal{A} = \{p_1, \ldots, p_n\}$ *is a finite set of processes;*
- $\Pi = \{A_1, \ldots, A_k\}$ *is a partition of* $\mathcal{A}$ *into agent types;*
- $\mathcal{P} = \{\mathcal{P}_1, \ldots, \mathcal{P}_k\}$ *is a collection of protocols that correspond to* $\Pi$*, one protocol per agent type;*
- $C$ *is a communication medium; and*
- $T$ *is a directed set representing global times.*

*The joint protocol of* $\langle \mathcal{A}, \Pi, \mathcal{P}, C, T \rangle$ *is the protocol formed by the protocols of all the agents.*

In this setting, given multiple possibly non-cooperating teams of agents, we need to re-define the notion of tasks and solvability. In general, the purpose of a protocol is to allow processes to solve distributed computing problems. An important subclass of such problems are *tasks* like consensus [LSP82], where the processes start with some local input constituting the system's initial configuration, and need to eventually compute some output i.e., a valid output configuration. In heterogeneous distributed systems, it is generally not possible to impose restrictions on the output of processes in other partitions.

**Definition 2** (Partial task)**.** *We say that a tuple* $\langle S, \mathcal{I}, \mathcal{O}, \Delta \rangle$ *is a* partial task relative to $S \subseteq \Pi$ *iff* $\mathcal{I}$ *is a set of input configurations for* $\mathcal{A}$*;* $\mathcal{O}$ *is a set of output configurations for* $S$*; and* $\Delta$ *is the validity correspondence function that maps valid input configurations of the system to a subset of valid output configurations for* $S$*.*

**Definition 3** (Solvability)**.** *Let* $\langle \mathcal{A}, \Pi, \mathcal{P}, C, T \rangle$ *be a heterogeneous distributed system. We say that set of agent types* $A_i \in \Pi$ can solve a partial task $\mathcal{T} = \langle S, \mathcal{I}, \mathcal{O}, \Delta \rangle$ *iff for any input configuration* $\sigma \in \mathcal{I}$*, the execution of the joint protocol of* $\langle \mathcal{A}, \Pi, \mathcal{P}, C, T \rangle$ *leads to a valid output configuration* $\rho$*, relatively to* $S$*, written* $\rho|_S \in \Delta(\sigma)$*.*

Note that traditional distributed systems with *benign failures*, where processes can only crash or lose messages, fall into the particular case where $\Pi = \{\mathcal{A}\}$ and there is one unique protocol executed by all processes. Similarly, distributed systems with send-restricted byzantine faults (no false perceptions of received messages, but arbitrary message sending) could be modeled as an instance with two types $\Pi^B = \{\mathsf{Correct}, \mathsf{Faulty}\}$, where all agents of type Correct follow the intended protocol, whereas agents of type Faulty can arbitrarily deviate from it.

### 1.1.2 Adaptive and self-adaptive systems

We have already mentioned that the complexity of modern distributed systems makes them inflexible w.r.t. to changes during operation, resulting in exploding maintenance costs. System designers are often unable to foresee every possible situation their system will be confronted with during its lifetime at design time. To overcome this issue, adaptive systems of various sorts have become increasingly popular. In this thesis, we mainly consider three kinds of system:

- *Traditional Distributed Systems* (including heterogeneous ones) where the system can be adapted and modified only by the direct external intervention of the system designer;
- *Traditional adaptive systems* (or simply adaptive systems) where agents cannot adapt their behavior during execution in a fully autonomous way, as they are limited to some possible corrections envisioned by the system designer beforehand; and;
- *SASO systems* that are able to self-adapt and self-organize with respect to unforeseen situations at run-time, without intervention of the system designer.

Indeed, traditional adaptive systems are in between the inflexibility of traditional distributed systems and the full-adaptability of SASO systems, as the role of the agents is restrained to simply choosing a different set of system assumptions should the need arise. SASO systems, on the other hand, despite being a heterogeneous class, share several self-* abilities [BG09], such as self-monitoring, self-assessment, self-awareness, self-reflection, context-awareness, planning, and self-improvement abilities, enabled by mechanisms dynamically reacting to changes in the environment or the system itself [BG09, THvM+14]. These capabilities allow SASO systems to react to changes both at the system level (e.g., some parameters need to be adjusted) and at the assumption level (e.g., even the specification of the system can be adapted to the situation at hand in an autonomous way, in order to maintain a certain (degraded) level of performance). The key element here is the ability of agents to reason at a meta-level of reasoning, that is, they are able to reason about the tools that they use for reasoning, being able to know the limits of their knowledge and to adapt this knowledge in case of unexpected behavior, marking a difference from traditional adaptive systems. This importance on the meta-level of knowledge is present

in the definitions of these self-* abilities. Computational self-reflection, in particular, has been defined [THvM+14, p.152] as:

> *The ability of the system to continuously monitor and improve its own behavior in an uncertain, dynamic, and time-variant environment for situations that may not have been anticipated at design-time of the system.*

This translates into the ability of agents to (i) monitor the behavior of the system and (ii) reason about it, guided by a meta-representation of the system itself. The traditional role of agents, namely, to operate according to some well-designed protocol, is, hence, augmented by the capability to observe (part of) the overall system behavior and to change (part of) the meta-representation accordingly, in an attempt to adapt to the changing environment and to possibly optimize the overall performance. These mechanisms underlie the ability of such systems to react to changes that were not envisioned by the system designer, which allows SASO systems to automatically adapt to such changes after deployment.

Whereas this allows a certain evolution of the system specification without the direct intervention of the system designer, it must be noted that this also opens up new sources and types of complex misbehavior of the system. Indeed, given the fairly weak reasoning capabilities of agents, compared to the system designer, it stands to reason that the "re-design decisions" made by a SASO system may be far from being correct, let alone optimal. Moreover, the process of designing a correct SASO system is considerably more complicated than that of a traditional system. All design steps are less compartmentalized, and there are many more features to be considered in each step (such as the mechanisms for self-modification and self-representation). Since a solid conceptual and formal basis, let alone tool support, is still lacking, existing solutions are largely based on *ad hoc* approaches.

A particular challenge in the design of fault-tolerant distributed systems is the impossibility of delegating the authority to change the self-representation of the whole SASO system from the system designer to a single "designer agent", as this would constitute a critical single point of failure. It is therefore reasonable to consider a system consisting of agents augmented with "designer agents" functionality, which can modify their own view of the system but not that of others.

We will discuss the ties between a priori knowledge and adaptive systems in Chapter 2 and in Chapter 5 we introduce a first a priori belief update mechanism that allows to simulate some of the self-* properties of SASO agents.

## 1.2 Epistemic logic and distributed systems

Formal epistemic modeling and analysis proved to be extremely useful for characterizing, once the system is designed, its evolution over time [HM90], for determining what processes can compute based on their local state in a given model [BZM14, CGM14, GM18, KPSF19b, KPSF19a, FKS21], and for deriving impossibility results [MT88]. In this context, processes are agents

commissioned to accomplish some task by following a specific protocol. A protocol can be viewed as a list of possible actions that an agent may perform in a given situation if certain preconditions are satisfied. Epistemic logic is especially well-fitted for knowledge-based protocols, i.e., protocols whose preconditions on actions make explicit reference to the local knowledge of the acting agents. Importantly, every protocol can be viewed as a knowledge-based protocol, as suggested by the Knowledge of Preconditions principle [Mos15][5], which states that a protocol instruction to act based on a precondition $\varphi$ can only be followed if the agent knows $\varphi$ to hold.

### 1.2.1 Standard epistemic logic

Epistemic logic is used to formally reason about knowledge in a plethora of frameworks, including possible world semantics[Hin62] and it has a long successful tradition both in formal epistemology and computer science. The key idea is to represent the various possibilities using "possible worlds" and to represent the knowledge (and the uncertainty) of agents via accessibility relations among these possibilities. For example, in the standard epistemic interpretation, an agent is said to know a formula $\varphi$ if such formula holds in all worlds that are accessible to this agent. In this interpretation, two accessible worlds are said to be indistinguishable for an agent, meaning that that agent does not have enough information to distinguish the two possible scenarios.

In this section, we introduce the basic tools and formal definitions that will be used throughout the thesis, especially in Chapters 3 to 5. The following definitions and theorems follow closely the ones in [vDHvdHK15]. We assume a fixed finite non-empty set of agents $\mathcal{A}$ and a non-empty set of propositions *Prop*. To reason about agents' epistemic states we introduce the so-called Kripke semantics:

**Definition 4** (Kripke Frames and Kripke Models)**.** *A Kripke frame* $\mathsf{F}$ *for the set of agents* $\mathcal{A}$ *is a tuple* $\langle W, R\rangle$ *where* $W \neq \varnothing$ *is a non-empty set of* possible worlds*, called* domain *and a function* $R\colon \mathcal{A} \to 2^{W\times W}$ *that assigns to each agent* $i \in \mathcal{A}$ *a binary* accessibility relation $R_i = R(i) \subseteq W \times W$*. We use the notation* $R_i(u) := \{v \in W \mid uR_iv\}$ *for the set of all worlds that agent* $i$ *considers possible in world* $u$*.*

*A Kripke model* $\mathcal{M} = \langle \mathsf{F}, V\rangle$ *is obtained by adding to a Kripke frame a* valuation function $V\colon Prop \to 2^W$ *that assigns to each atomic proposition* $p \in Prop$ *a set of worlds* $V(p) \subseteq W$ *where* $p$ *holds. A* pointed Kripke model *is a pair* $(\mathcal{M}, v)$ *where* $\mathcal{M}$ *is a Kripke model and* $v \in W$ *represents the* real *(or* actual*)* world*.*

**Definition 5** (Language)**.** *The basic epistemic language* $\mathcal{L}$ *is defined by the following BNF:*

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid K_i\varphi$$

*where* $p \in Prop$ *is an atomic proposition and* $i \in \mathcal{A}$*. We use standard propositional abbreviations:* $\varphi \vee \psi := \neg(\neg\varphi \wedge \neg\psi); \varphi \to \psi := \neg\varphi \vee \psi; \bot := p \wedge \neg p$ *and* $\top := \neg\bot$*.*

**Definition 6** (Semantics of $\mathcal{L}$)**.** *Given a Kripke model* $\mathcal{M} = \langle W, R, V\rangle$ *with* $w \in W$*,* $i \in \mathcal{A}$*:*

[5] We make this notion precise in Theorem 12.

$\mathcal{M}, w \vDash p$ *iff* $w \in V(p)$

$\mathcal{M}, w \vDash \neg\varphi$ *iff* $\mathcal{M}, w \nvDash \varphi$

$\mathcal{M}, w \vDash (\varphi \wedge \psi)$ *iff* $\mathcal{M}, w \vDash \varphi$ *and* $\mathcal{M}, w \vDash \psi$

$\mathcal{M}, w \vDash K_i\varphi$ *iff* $\mathcal{M}, w \vDash \varphi$ *for all* $w' \in W$ *s.t.* $w' \in R_i(w)$

*A formula $\varphi$ is valid in $\mathcal{M}$ if, for all $w \in W$, $\mathcal{M}, w \vDash \varphi$. If $\varphi$ is valid in $\mathcal{M}$, we use the notation $\mathcal{M} \vDash \varphi$. If $\varphi$ is not valid in $\mathcal{M}$, we use $\mathcal{M} \nvDash \varphi$. Given a class of models $\mathbf{C}$, we say that $\varphi$ is valid in the class of models $\mathbf{C}$ iff $\mathcal{M} \vDash \varphi$ for all $\mathcal{M} \in \mathbf{C}$.*

An analogous notion of validity can be defined for Kripke frames:

**Definition 7.** *A formula $\varphi$ is valid in $\mathsf{F} = \langle W, R\rangle$ iff for all $\mathcal{M} = \langle \mathsf{F}, V\rangle$, $\mathcal{M} \vDash \varphi$. Given a class of frames $F$, we say that $\varphi$ is valid in $F$ iff $\mathsf{F} \vDash \varphi$ for all $\mathsf{F} \in F$.*

We now define frame properties and corresponding valid formulas following the standard account in [vDHvdHK15]:

**Definition 8** (Frame properties)**.** *A binary relation $R \subseteq W \times W$ is called:*

- reflexive *iff $wRw$ for all $w \in W$. The class of reflexive models is denoted $\mathcal{KT}$;*
- transitive *iff for all $w, v, u \in W$ we have $wRu$ whenever $wRv$ and $vRu$. The class of transitive models is denoted $\mathcal{K}4$;*
- euclidean *iff for all $w, v, u \in W$ we have $vRu$ whenever $wRv$ and $wRu$. The class of euclidean models is denoted $\mathcal{K}5$;*

*In addition, a binary relation is* symmetric *iff for all $w, v \in W$ we have $vRw$ whenever $wRv$. Relation $R$ is called an* equivalence relation *iff it is reflexive, transitive, and euclidean. The class of models where $R$ is an equivalence relation is denoted **S5**;*

In the epistemic analysis of distributed systems, one of the most common logics is **S5** for knowledge $K_a\varphi$, which we usually assume unless specified otherwise.

**Definition 9** (Valid formulas)**.** *(i)* $\mathcal{KT} \vDash K_i\varphi \to \varphi$;

*(ii)* $\mathcal{K}4 \vDash K_i\varphi \to K_iK_i\varphi$;

*(iii)* $\mathcal{K}5 \vDash \neg K_i\varphi \to K_i\neg K_i\varphi$

The intuitive meaning these properties are the following: (i) is called *factivity*, and it states that what agents know is true, (ii) is called *positive introspection*, and it means that agents know what they know; finally (iii) is called *negative introspection* captures the idea that agents know what they don't know.

For many applications, however, **S5** is an assumption too strong. It is obvious that in uncertain contexts, such as fault-tolerant distributed systems, factivity might fail: as agents are usually uncertain whether they are correct or faulty, their (presumed) "knowledge" cannot be trusted. For this reason, in this thesis we often drop factivity. To avoid confusion, we use the symbol $K$ to refer to **S5** knowledge, and we use the symbol $B$ to refer to $\mathcal{K}45$ *belief* and rely on the multimodal doxastic language of belief $\mathcal{L}^B$:

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid B_i\varphi \tag{1.1}$$

where $i \in \mathcal{A}$ and $p \in Prop$. Formula $B_i\varphi$ means *agent i believes φ (to be true)*. The semantics for $\mathcal{L}^B$ is the same as in Definition 6, but with the important distinction that we generally do not assume the factivity of beliefs, while we retain positive and negative introspection of beliefs. The other Boolean connectives are defined as usual and $\widehat{B}_i\varphi := \neg B_i\neg\varphi$ representing the dual modality *considers possible*.

Since a class of models can be defined by a set of validities, we get the following result:

**Lemma 10** (Fewer models means more validities)**.** *Let $C_1$ and $C_2$ be two collections of models such that $C_1 \subseteq C_2$. Then $C_2 \vDash \varphi$ implies $C_1 \vDash \varphi$ for any formula $\varphi \in \mathcal{L}$.*

### Group modalities and common knowledge

Knowledge is an important tool also because it is possible to formulate group modalities that consider the epistemic state of a group of agents. For a non-empty group of agents $G \subseteq \mathcal{A}$, the most commonly used are:

- Mutual knowledge $E_G\varphi$: Everybody in a group $G$ knows $\varphi$, formally defined as the conjunction of the knowledge of agents in $G$: $E_G\varphi := \bigwedge_{i\in G} K_i\varphi$. Alternatively, $E_G\varphi$ is definable via a dedicated accessibility relation $R_G = \cup_{i\in G} R_i$, and evaluating $E_G\varphi$ as follows:

$$\mathcal{M}, s \vDash E_G\varphi \text{ iff } \mathcal{M}, s' \vDash \varphi \text{ for all } s' \in W \text{ s.t. } sR_Gs' \tag{1.2}$$

- Distributed knowledge $D_G\varphi$: This is the knowledge that agents would gain if they could unify their local views. To evaluate $D_G\varphi$, we define the relation as the intersection of the relations of individual agents: $R_G^D = \cap_{i\in G} R_i$ , and evaluate $D_G\varphi$ as follows:

$$\mathcal{M}, s \vDash D_G\varphi \text{ iff } \mathcal{M}, s' \vDash \varphi \text{ for all } s' \in W \text{ s.t. } sR_G^Ds' \tag{1.3}$$

  The idea behind the above definition is that if some agent has eliminated a world as an epistemic alternative, then so will the group.

- Common knowledge $C_G\varphi$: Everybody in $G$ knows that everybody in $G$ knows that everybody in $G$ knows, and so on. To avoid an infinite conjunction, we again evaluate $C_G\varphi$ via the accessibility relation: $R_G^C := (\cup_{i\in G} R_i)^*$, where $(Q)^*$ is the reflexive and transitive closure of $Q$. In this way, we can evaluate $C_G\varphi$ as follows:

$$\mathcal{M}, s \vDash C_G\varphi \text{ iff } \mathcal{M}, s' \vDash \varphi \text{ for all } s' \in W \text{ s.t. } sR_G^C s' \tag{1.4}$$

  Intuitively, $C_G\varphi$ requires that $\varphi$ holds at any state along any path (of arbitrary depth) made by accessibility relations from agents in $G$.

Naturally the basic language $\mathcal{L}$ can be extended with the group modalities. $E_G\varphi$ can be added to the standard language $\mathcal{L}$ as the abbreviation of $\bigwedge_{i\in G} K_i\varphi$. Since distributed and common knowledge use different accessibility relations, we use $\mathcal{L}^D$ for the language defined by the following BNF:

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid K_i\varphi \mid D_G\varphi \tag{1.5}$$

And we use $\mathcal{L}^C$ for the language defined by the following BNF:

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid K_i\varphi \mid C_G\varphi \tag{1.6}$$

Where the semantics of the new operators is defined above.

We formally define expressivity in order to compare logical languages:

**Definition 11** (Expressivity)**.** *Given two logical languages $\mathcal{L}_1$, $\mathcal{L}_2$ and a class of Kripke structures $C$, we say that $\mathcal{L}_2$ is at least as expressive as $\mathcal{L}_1$, denoted $\mathcal{L}_1 \preceq_C \mathcal{L}_2$, if, for every formula $\varphi \in \mathcal{L}_1$, there is a $\psi \in \mathcal{L}_2$ such that the two formulas are equivalent on $C$. If the converse holds, the two languages are equally expressive on $C$, denoted $\mathcal{L}_1 \equiv_C \mathcal{L}_2$. If $\mathcal{L}_2$ is strictly more expressive than $\mathcal{L}_1$ we write $\mathcal{L}_1 \prec_C \mathcal{L}_2$. If $\mathcal{L}_1 \npreceq_C \mathcal{L}_2$ and $\mathcal{L}_2 \npreceq_C \mathcal{L}_1$ then the two languages are said to be incomparable.*

The languages enriched with common knowledge $\mathcal{L}^C$ and the languages with distributed knowledge $\mathcal{L}^D$ are more expressive (Definition 11) than the basic epistemic language $\mathcal{L}$ [vDvdHK07].

Common knowledge is deeply related to coordination and to agreement problems [FHMV95], and agents are usually required to have some initial common knowledge in order to solve tasks. In later sections of the thesis, we will see how common knowledge plays also a crucial role in defining system assumptions and agents' a priori knowledge.

### 1.2.2 Common knowledge of the model

In Section 1.2.1, we have introduced the notion of common knowledge, which is crucial for solving coordination tasks like consensus. Epistemic logic formalizes common knowledge as a modality. Independently of the frame conditions, for example, axioms and other validities are common

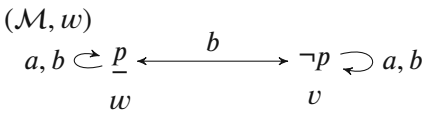



**Figure 1.1:** Pointed Kripke model $(\mathcal{M}, w)$ where $a$ knows that $p$, while $b$ does not.

knowledge of all agents. This is not harmful in any way, but we can see that common knowledge brings a lot more than expected. For example, one basic question that may arise concerns the ability of agents to use other agents' accessibility relations: how does an agent $a$ know that there is common knowledge of a formula? From an epistemic perspective, $a$ needs to travel from the supposed actual state to any state accessible via an arrow labeled by any agent. But how does the agent know that between two states there is a $b$-arrow when there is no $a$-arrow leading to that state? The only possible answer is that the model itself is actually commonly known by agents. As argued by Artemov [Art20], in multi-agent settings, *common knowledge of the model* (CKM) is required by agents in order to compute higher-order knowledge of other agents. While Artemov's argument focuses on uncertainties about facts, it can also be extended to agents' uncertainty about attitudes of other agents.

In this thesis, we identify the underlying assumption, in multi-agent Kripke models, in the implicit ontological distinction between two kinds of possible worlds: (a) worlds that are actually possible (AP) and (b) worlds that are only virtually possible (VP)[6]. While the former worlds constitute, for a given agent, the arena in which the actual world might lie, the latter kind of worlds are considered only to the extent of computing other agents' knowledge. As an example, consider the pointed Kripke model $(\mathcal{M}, w)$ of Figure 1.1, where agent $b$ is uncertain whether $p$ or $\neg p$ is the case, while agent $a$ knows that $p$. Thus, agent $a$ does not consider the world $v$ where $\neg p$ holds as actually possible, because $a$ already ruled out that possibility. However, the world $v$ and the $b$-accessible arrows from $w$ to $v$ (and back) are used by $a$ to compute $b$'s higher order knowledge. In this sense, in $\mathcal{M}, w$ for agent $a$ world $w$ is AP while world $v$ is only VP[7]

Artemov argues that without the CKM (comprising both APs and VPs), agents would not be able to compute such higher-order knowledge. While in general some form of common knowledge assumptions are unavoidable, in the following chapters we will also see examples where the CKM is not desirable, leading to a multitude of (partially disjoint) sets each representing the completely private local view of a given agent. In particular, a priori knowledge updates, which are meant to modify the very system assumptions, may break the common knowledge of the model, especially if conducted on an agent-by-agent basis as in Chapter 5.

[6] In standard epistemic logic, the notion of actually possible world correspond to the closure of the actual world under the accessibility relation of that agent. In [BA11b, Def. 3], it is referred to as the associated local state. However, in the context of a priori updates, the situation becomes more complicated, as the actual world might not be considered actually possible anymore, and some virtually possible worlds can become actually possible worlds for the updating agent. We will return to this problem in Chapter 4 and Chapter 5.

[7] In other words, the APs for an agent is made by the closure of the actual world under the accessibility relation of that agent, i.e., the "associated local state" in the sense of [Ba11a].

### 1.2.3 Runs and systems framework

In this section we introduce the standard framework for reasoning about distributed systems, namely the *runs and systems* framework [FHMV95, HM90], which employs a slightly modified Kripke structure that naturally includes time. The runs and systems framework is among the most popular and successful in modeling knowledge-based protocols in distributed systems. Most of the work on formalization of (byzantine) distributed systems is done in this framework. In this thesis, however, we will rely mostly on standard standard Kripke modeling, as the latter has a wider literature on dynamic updates and non-common situations, in which we are interested to model a priori updates. Since we draw many concepts from the runs and system literature, we report here its main elements.

A system is modeled as a set of runs, each representing a possible time-evolution of the global system state. A run is a sequence of global states, ordered by time. Formally, an interpreted system $\mathcal{I} = (\Sigma, \pi)$ is made of a set of all possible runs $\Sigma$ and it determines the set of pairs $(r, t)$ of a run $r \in \Sigma$ and time $t \in \mathbb{N}$ (called *points*) representing global states; and the evaluation function $\pi$ determines which atoms are true at a point $(r, t)$. While time is modeled discretely for simplicity, it might not be generally available to agents, e.g. in asynchronous systems. Each agent has access to its history, i.e., the collection of actions performed and events witnessed by the agent, constituting their local view, a notion of utmost importance in distributed systems: Agents do not have the global view of the system or of its evolution; they can keep track of only the limited portion of the system they can directly observe, and it is only this portion that is recorded in their history. Only the system designer is assumed to have the global view of the system, even if not necessarily correct and complete. In line with the standard account, we use $r(t)$ for the global state, and notation $r_i(t)$ for the local state of agent $i$ at run $r$ and time $t$. A protocol $P_i$ for agent $i$ is a function from the set $L_i$ of $i$'s local states to non-empty set of actions $ACT_i$. A run $r$ is consistent with a protocol $P$ if it could be generated by running protocol $P$ [BZM14].

Knowledge is a powerful conceptual abstraction for distributed systems, especially since reasoning about agents' "states of knowledge" at various points in the execution of a distributed protocol [HM90, Hal03] has proven to be extremely fruitful both for the designer and the analyzer of a distributed system. Two pairs $(r, t)$ and $(r', t')$ are indistinguishable for agent $i$ iff $i$ has the same local state in both global states represented by those points, formally, if $r_i(t) = r'_i(t')$. The modal knowledge operator $K_i$ is used to capture that agent $i$ knows some fact $\varphi$ in a run $r \in \Sigma$ and and time $t \in \mathcal{N}$, according to the standard **S5** interpretation of knowledge, written $(\mathcal{I}, r, t) \vDash K_i\varphi$. This interpretation of knowledge captures the intuition that, being in the global state $(r, t)$ of the interpreted system $\mathcal{I}$, agent $i$ knows $\varphi$ if and only if $\varphi$ holds in every global state $(r', t')$ that is indistinguishable from $(r, t)$ for $i$. Related notions such as group knowledge and common knowledge can be defined in the standard way. Indeed, knowledge is at the core of the agents' ability to act according to the protocol, as stated by the Knowledge of Preconditions principle [Mos15]:

**Theorem 12** (Knowledge of precondition principle)**.** *Let $\alpha$ be a conscious action for agent $i$ in $\mathcal{I}$. If $\varphi$ is a necessary condition for $does_i(\alpha)$ in $\mathcal{I}$, then $K_i\varphi$ is also a necessary condition for $does_i(\alpha)$ in $\mathcal{I}$.*

where an action $\alpha$ is a conscious action for $i$ in $\mathcal{I}$ if $i$'s local state completely determines whether $i$ performs $\alpha$ (written $does_i(\alpha)$).

While trivial for preconditions based on the local state of the acting agent itself, this observation comes to the fore for global preconditions, also involving other agents, as is common for coordination problems such as consensus.

The other important component introduced by the runs and systems framework is the time component. We extend the basic epistemic language $\mathcal{L}$ with temporal modalities, which results in $\mathcal{L}^{\Diamond}$ defined by the following BNF:

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid K_a\varphi \mid \Diamond\varphi \tag{1.7}$$

where, for a run $r \in \Sigma$, time $t \in \mathbb{N}$, the operator $\Diamond\varphi$ is defined as: $(\mathcal{I}, r, t) \vDash \Diamond\varphi$ iff $(\mathcal{I}, r, t') \vDash \varphi$ for some $t' \geq t$.

Adding the time components to the logic allows to express naturally crucial properties of some distributed tasks, like stabilizing agreement [CFG24], which is a termination-free version of the standard agreement problem. It also allows to represent message passing without introducing any dynamic operator in the language. This epistemic framework has also been successfully applied to byzantine fault-tolerant distributed systems [FKS21, KPSF19b, KPSF19a, KPS+19], by using the hope modality $H_i\varphi := correct_i \rightarrow B_i\varphi$, where $B_i\varphi$ represents belief of agent $i$ and is understood in the spirit of belief as defeasible knowledge [MS93], where $B_i\varphi := K_i(correct_i \rightarrow \varphi)$[8]. This hope modality $H_i\varphi$ is equivalent to a disjunction $\neg correct_i \vee (correct_i \wedge B_i\varphi)$, suggesting that a message $\varphi$ from $i$ is interpreted as the uncertainty between $i$ being faulty or the epistemic state of $i$ confirming $\varphi$ in case $i$ is a correct agent. In the former case, the message carries no meaning whatsoever, as faulty agents have inconsistent epistemic attitudes, which means they can send any possible message (and not only inconsistent ones). Consequently, the axiomatization of hope in [vDFK22] takes $H_i\bot$ to be the definition of faulty agents because only a faulty agent can send contradictory messages.

As we will see in the next section, dynamic model-transforming operations can be expressed in the language and each resulting Kripke model can be seen as a snapshot of the system, mimicking the evolution a run in a runs and systems framework[9].

### 1.2.4 Dynamic epistemic logic

Standard epistemic logic can be enriched with modalities that express informational change due to actions, encoding indirectly the passage of time.

[8] Unlike the belief operator introduced of Equation (1.1), the one introduced in [MS93] is based on a knowledge modality and it is conditional to the correctness of the agent in question, while preserving positive and negative introspection.

[9] One key difference between the two is that the runs and systems framework uses a linear-time temporal logic, while most standard dynamic epistemic logic account employ a branching-time temporal logic. For details about the distinction between linear and branching time logics, see [GR21]

We begin by introducing the simpler version of dynamic epistemic logics, namely public announcement logic *PAL* [Pla89]. Public announcements inform simultaneously all agents about some information $\varphi$. The following definitions and theorems follow the ones in [vDvdHK07]. We first extend the basic multimodal epistemic language $\mathcal{L}$ with modalities expressing public announcements:

**Definition 13** (Language of PAL)**.** *The epistemic language with public announcements $\mathcal{L}^{PAL}$ for agents from $\mathcal{A}$ is defined by*

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid K_i\varphi \mid [\varphi]\varphi$$

*where $p \in Prop$ is an atomic proposition and $i \in \mathcal{A}$.*

**Definition 14** (Truth)**.** Truth of a formula $\varphi \in \mathcal{L}^{PAL}$ at a world $w$ of a Kripke model $\mathcal{M} = \langle W, R, V\rangle$ *is defined by recursion on the structure of $\varphi$: $\mathcal{M}, w \vDash p$ iff $w \in V(p)$; boolean connectives behave classically; $\mathcal{M}, w \vDash K_i\varphi$ iff $\mathcal{M}, u \vDash \varphi$ for all $u \in W$ such that $wR_iu$; finally, $\mathcal{M}, w \vDash [\varphi]\psi$ iff either $\mathcal{M}, w \nvDash \varphi$ or ($\mathcal{M}, w \vDash \varphi$ and $\mathcal{M}|\varphi, w \vDash \psi$), where $\mathcal{M}|\varphi := \langle W', R', V'\rangle$ is defined as:*

- $W' := \{u \in W \mid \mathcal{M}, u \vDash \varphi\}$ *(note that the actual world $w \in W'$ whenever $W'$ is constructed);*
- $R'_i := R_i \cap (W' \times W')$ *for each $i \in \mathcal{A}$;*
- $V'(p) := V(p) \cap W'$ *for each $p \in Prop$.*

*Formula $\varphi$ is* false at world *$w$, denoted $\mathcal{M}, w \nvDash \varphi$, iff it is not true at $w$.*

We briefly introduce an axiomatization of *PA*L in order to show some expressivity results.[10]

**Definition 15** (PAL Theory)**.** *The axiomatic theory PAL is defined in Table 1.1. We write $PAL \vdash \varphi$ (or $\vdash_{PAL} \varphi$) to denote that the formula $\varphi$ is derivable in the axiomatic theory PAL. Its negation is written $PAL \nvdash \varphi$.*

Axioms P1–P5 are called reduction axioms. Note that, from left to right, the complexity of the formulas to which the update model applies decreases. Intuitively, P4 says that agent $a$ knows $\varphi$ after the true public announcement of $\psi$ if and only if $[\psi]\varphi$ is true in all of the worlds that can be reached by an $a$-arrow before the announcement.

The following expressivity results can be found in [Pla89]:

**Theorem 16** (PAL Expressivity Theorem)**.** *Given $\mathcal{L}^{PAL}$ and $\mathcal{L}$, for each formula $\varphi \in \mathcal{L}^{PAL}$ there is a formula $\psi \in \mathcal{L}$ such that $\vDash \varphi \leftrightarrow \psi$, i.e. the two languages are equally expressive over the class of all pointed Kripke model.*

[10] However, we will not introduce an axiomatization for all the dynamic logics introduced here.

| | |
|---|---|
| CL. | Axioms of classical propositional logic |
| K. | $K_a(\varphi \to \psi) \to (K_a\varphi \to K_a\psi)$ |
| T. | $K_a\varphi \to \varphi$ |
| 4. | $K_a\varphi \to K_aK_a\varphi$ |
| 5. | $\neg K_a\varphi \to K_a\neg K_a\varphi$ |
| P1. | $[\psi]p \longleftrightarrow (\psi \to p)$ for $p \in Prop$ |
| P2. | $[\psi]\neg\varphi \longleftrightarrow (\psi \to \neg[\psi]\varphi)$ |
| P3. | $[\psi](\varphi \wedge \theta) \longleftrightarrow ([\psi]\varphi \wedge [\psi]\theta)$ |
| P4. | $[\psi]K_a\varphi \longleftrightarrow (\psi \to K_a[\psi]\varphi)$ |
| P5. | $[\psi][\theta]\varphi \longleftrightarrow ([\psi \wedge [\psi]\theta]\varphi)$ |

$$\frac{\varphi \to \psi \qquad \varphi}{\psi}\ \text{MP} \qquad \frac{\varphi}{K_a\varphi}\ \text{KN}$$

**Table 1.1:** Theory of PAL [vDvdHK07].

The theorem easily follows from the reduction axioms exposed in Table 1.1: every formula containing a public announcement operator can be reduced to a formula of a simpler complexity, meaning that for every formula in $\mathcal{L}^{PAL}$ it is possible to construct an equivalent one in $\mathcal{L}$.

In addition, the order of public announcements is important in some cases:

**Lemma 17** (Public announcements are not commutative)**.** *Given arbitrary formulas $\varphi, \psi, \theta \in \mathcal{L}^K$, it is not the case that $\vDash_{S5} [\varphi][\psi]\theta \longleftrightarrow [\psi][\varphi]\theta$, i.e., the order of public announcements is (sometimes) relevant.*

*Proof.* As a counterexample, take $\varphi = p$ and $\psi = \neg K_a p$. Suppose that in some world $s$ of a model $\mathcal{M}$, $\mathcal{M}, s \vDash_{S5} \neg K_a p$. Announcing $\varphi$ and then $\psi$ in $s$ results in $K_a\bot$, while announcing them in the opposite order results in a restriction of the model where $p$ holds (as the first announcement is simply uninformative). □

Alternatively to the model-transforming operation originally proposed by Plaza [Pla89], which is a world-deleting operation, Gerbrandy and Groeneveld [GG97] independently proposed an update mechanism that deletes arrows instead of worlds. For example, an update of $\varphi$ does not delete the worlds where $\neg\varphi$ is true, but rather deletes only the arrows pointing to worlds where $\neg\varphi$ holds. Conveniently, this semantics does not require the announcement to be truthful. Moreover, the two approaches are expressively equivalent, as they are both as expressive as the standard language of epistemic logic $\mathcal{L}$. In the remainder of the thesis, we will focus on world-deleting announcements.

The explanatory power of dynamic epistemic logic can be better appreciated by considering complex epistemic puzzles where agents take epistemic actions (such as public announcements) to solve a certain task. In the following, we briefly introduce two puzzles (and their solutions) that will be used throughout the thesis.

### Muddy children puzzle

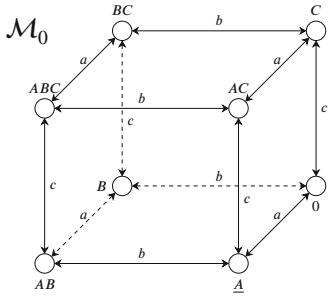
In the *muddy children puzzle (MCP)* [FHMV95], $n$ children are playing in the mud, and some of them, say $k$, get mud on their forehead. Each can see the mud on others but not on his own forehead. The father comes and announces "at least one of you has mud on her forehead". The father then starts repeating the question: "if you know whether you are muddy or not, step forward". Since the children are perfect reasoners, the muddy children will not step up for the first $k-1$ times the father asks the question, but it can be shown ([FHMV95]) that the muddy children will all step up at the $kth$ iteration. For instance, with three children playing and one muddy child, the muddy child should immediately step up. This variable is not known to the agent itself, formally $\neg K_a m_a \wedge \neg K_a \neg m_a$, but every agent knows the other's variable, that is, for all $b \neq a$, $K_a m_b \vee K_a \neg m_b$. Father's role is of an external announcer, making his epistemic state irrelevant to the solution to the puzzle.[11] Importantly, without the father's initial announcement 'at least one of you has mud on her forehead" the puzzle cannot be solved. We present a solution for $n = 3$ and $k = 1$. The initial Kripke model $\mathcal{M}_0$ is represented in Figure 1.2a. In particular, the set of agents $\mathcal{A} = \{a, b, c\}$, and father has the role of an external announcer and not of an agent trying to solve the puzzle. The puzzle is solved after two iterations of the announcement "if you know whether you are muddy or not step forward". Let us see why. Intuitively, after father's announcement "at least one of you has mud on her forehead" the three agents $a, b, c$ know that at least one agent is muddy (Figure 1.2b). Say $a$ is the only muddy child. From her perspective, $b$ and $c$ are clean, which means that she is the only muddy one. She will consequently step forward after the first iteration of the announcement: "if you know whether you are muddy or not step forward". On the other hand, $b$ and $c$ each see one muddy agent and one clean. This means that at the first iteration they don't know whether they are themselves muddy, so they don't step forward . However, in the second iteration (Figure 1.2c) they can step forward: since $a$ stepped forward at the first iteration, they now know that she saw two clean agents. Thus, $b$ and $c$ step forward after the second iteration.

### Consecutive numbers

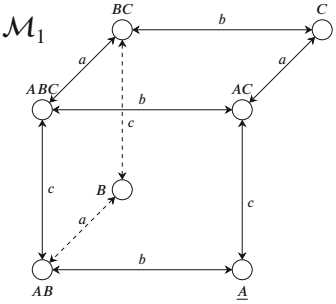
Two agents $a$ and $b$ are privately told two natural numbers. In addition, they are publicly told that the two numbers are consecutive (making it common knowledge). Suppose that $a$ is told number 1 and $b$ is told number 2. They are allowed to state whether they know the other's number or not, but not allowed to communicate their own number. In the standard formulation, the puzzle is solved by the following dialog:

- $a$: I don't know your number.
- $b$: I don't know your number either.
- $a$: Now I know your number.

[11] From a distributed perspective, father's role is akin to one of the system designer: the first announcement "at least one of you has mud on her forehead", which delimits the case of alternatives considered possible by agents, can be seen as an a priori knowledge update, setting the arena of possibilities for the agents.




**(a)** Kripke model $\mathcal{M}_0$ for the Muddy children puzzle with three agents of Section 1.2.4, before father's initial announcement. $D$ in the name of the state corresponds to $m_d$ being true at the state, i.e., agent $d$ being muddy. $\underline{A}$ indicates that $A$ is the actual world. Bidirectional arrows (both dashed and solid) represent indistinguishability for agents, e.g., agent $b$ cannot distinguish between states $ABC$ and $AC$. Reflexive loops are omitted.




**(b)** Kripke model $\mathcal{M}_1$ resulting from the children stepping up after father's initial announcement. Reflexive loops are omitted.

$\mathcal{M}_2$

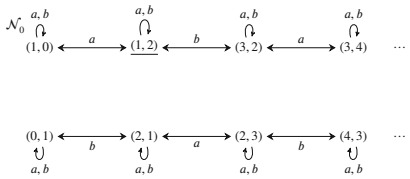

**(c)** Kripke model $\mathcal{M}_2$ resulting from the children stepping up after father's second iteration of the announcement. Reflexive loops are omitted.

**Figure 1.2**

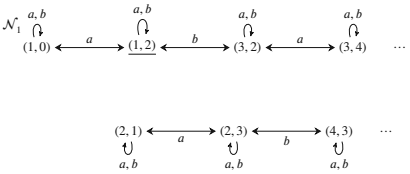
- $b$: Now I know yours too.

Here, the first statement by $a$ is uninformative to $b$. [12]. By the second statement, $b$ makes it clear that $b$'s number is not 0, which enables $a$ to conclude that $b$'s number must be 2 (since $a$ holds 1

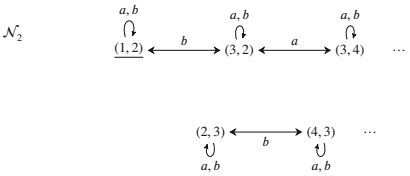
[12]It is uninformative because $b$ already knows that $a$ does not know his number. If it was not the case, then $a$ was communicated 0, which clearly $b$ excluded.



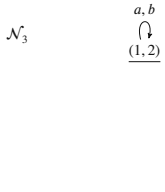

**(a)** Kripke model $\mathcal{N}_0$ representing the initial state of both agents in the consecutive numbers puzzle. Each pair represents the pair of numbers held by respectively $a$ and $b$. The actual world is underlined. The Kripke model has two disconnected components representing respectively the cases where $a$ and $b$ are told the same numbers but swapped.




**(b)** Kripke model $\mathcal{N}_1$ resulting from $a$'s initial announcement.




**(c)** Kripke model $\mathcal{N}_2$ resulting from $b$'s announcement. Now $a$ knows $b$ has two, and can announce that she knows her number.




**(d)** Kripke model $\mathcal{N}_3$ resulting from $a$'s announcement. Now everybody knows the pair of numbers.

**Figure 1.3**

and the numbers are consecutive). So $a$ can announce that she knows $b$'s value, which reveals to $b$ that $a$ has number 1.

The standard epistemic modeling of this example involves a Kripke model that each agent is supposed to build based on the rules of the puzzle in a way that makes this model commonly known to both agents.

**Remark 18.** *In order to solve many epistemic puzzles, such as those introduced above, a certain amount of common knowledge is required. In particular, the agents use the same Kripke model to reason not only about their epistemic state, but also about other agents' epistemic states. Clearly,*

*this a priori assumption plays a crucial role and the consequences of dropping this assumption are explored in detail when dealing with a priori updates in Chapters 4 and 5.*

**Remark 19.** *When formalizing epistemic puzzles, it might not be the case that there is always a unique Kripke model representing the target epistemic situation and its evolution [Art16]. This means that once we drop some common knowledge of agents regarding the model, then it becomes highly unlikely that the other agents are actually using the same model to reason about the same epistemic puzzle. Even in case of complete information, there is no guarantee. At best, agents might use distinct bisimilar models, that reflect epistemic change in the same way.*

### 1.2.5 Action model logics

Epistemic actions modeling communication, especially in distributed systems, are not limited to public announcements only: private messages and/or semi-public messages can also be formally represented. Relational structures such as *action models* in *Action Model Logic* (AML) [BMS98, vDvdHK07] and *arrow update models* in the *Arrow Update Logic* (AUL) [KR11a] are used to represent the evolution of the agents' knowledge under information change in complex communication scenarios. Because AUL and AML have been shown to be equally *update expressive* [DDvdHKK20], we use the term "update models" to refer to either action models of AML or arrow update models of AUL. [13] In the following, we will briefly introduce some basic definitions of action model logic following [vDvdHK07]:

**Definition 20** (Pointed action model)**.** *For a set of agents $\mathcal{A}$, an* action model *is a triple $U = \langle E, Q, \mathsf{pre}\rangle$ where $\langle E, Q\rangle$ is a finite Kripke frame with domain $E$ consisting of* action points *or* events*, and the* precondition function $\mathsf{pre} : E \to \mathcal{L}$ *assigns the precondition* $\mathsf{pre}(\beta) \in \mathcal{L}$ *that is necessary for an event $\beta \in E$ to happen. A* pointed action model *is a pair $(U, \alpha)$ where $\alpha \in E$ represents the* actual event/action point*.*

An action model simulates the results of an epistemic action in a Kripke model via the *product update* operation:

**Definition 21** (Product update)**.** *The* (restricted modal) product update *of a Kripke model $\mathcal{M} = \langle S, R, V\rangle$ with an action model $U = \langle E, Q, \mathsf{pre}\rangle$ is a Kripke model $\mathcal{M} \otimes U := \langle S', R', V'\rangle$ where*

- $S' := \{(v, \beta) \in S \times E \mid \mathcal{M}, v \vDash \mathsf{pre}(\beta)\}$,
- $R'_i := \big\{\big((v, \beta), (u, \gamma)\big) \in S' \times S' \mid (v, u) \in R_i \text{ and } (\beta, \gamma) \in Q_i\big\}$,
- $V'(p) := \{(v, \beta) \in S' \mid v \in V(p)\}$.

*If $S' = \emptyset$, the product update is undefined.*

[13] While there are some attempts to integrate announcement logics and the runs and systems framework [PS10], the vast majority of the results rely on standard Kripke semantics.

A product update provides the semantics for a communication scenario represented by a pointed action model $(U, \alpha)$: formula $\varphi$ is true after the communication $U$, i.e., $\mathcal{M}, w \vDash [U, \alpha]\varphi$, iff $\mathcal{M} \otimes U, (w, \alpha) \vDash \varphi$.

**Definition 22** (Language of AML)**.** *The epistemic language with action models $\mathcal{L}^{AML}$ is defined by the following grammar:*

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid K_i\varphi \mid [U, \alpha]\varphi$$

*where $p \in Prop$, $\alpha \in E$ is an action point and $i \in \mathcal{A}$.*

**Definition 23** (Semantics of AML)**.** *All the standard cases are defined as usual, the new one is defined as follows: $\mathcal{M}, w \vDash [U, \alpha]\varphi$ iff either $\mathcal{M}, w \nvDash pre(\alpha)$ or $\mathcal{M} \otimes U, (w, \alpha) \vDash \varphi$, where $\mathcal{M} \otimes U := \langle W', R', V' \rangle$ is defined as in Definition 21.*

We can now introduce the notion of bisimilarity both for Kripke frames and action models:

**Definition 24** (Bisimulation)**.** *A* bisimulation between Kripke frames $\langle W, R \rangle$ *and* $\langle W', R' \rangle$ *is a nonempty binary relation $\mathcal{B} \subseteq W \times W'$, such that for every $v \mathcal{B} v'$ and $i \in \mathcal{A}$:*

- ***Forth**: if $vR_iu$, then there is $u' \in S'$ such that $v' R_i' u'$ and $u \mathcal{B} u'$;*
- ***Back**: if $v' R_i' u'$, then there is $u \in S$ such that $vR_iu$ and $u \mathcal{B} u'$.*

*A* bisimulation between Kripke models $\mathcal{M} = \langle S, R, V \rangle$ *and* $\mathcal{M}' = \langle S', R', V' \rangle$ *is a bisimulation $\mathcal{B} \subseteq S \times S'$ between their frames $\langle S, R \rangle$ and $\langle S', R' \rangle$ that additionally satisfies for every $v \mathcal{B} v'$ and $p \in Prop$:*

- ***Atoms:** $v \in V(p)$ iff $v' \in V'(p)$.*

*A* bisimulation between action models $U = \langle E, Q, \mathsf{pre} \rangle$ *and* $U' = \langle E', Q', \mathsf{pre}' \rangle$ *is a bisimulation $\mathcal{B} \subseteq E \times E'$ between their frames $\langle E, Q \rangle$ and $\langle E', Q' \rangle$ that additionally satisfies for every $\alpha \mathcal{B} \alpha'$:*

- ***Pre:** $\mathsf{pre}(\alpha) \equiv \mathsf{pre}'(\alpha')$.*

*Pointed Kripke models $(\mathcal{M}, v)$ and $(\mathcal{M}', v')$ are* bisimilar*, denoted $\mathcal{M}, v \leftrightarroweq \mathcal{M}', v'$, iff there is a bisimulation $\mathcal{B}$ between $\mathcal{M}$ and $\mathcal{M}'$ such that $v \mathcal{B} v'$.*

*Pointed Kripke models $(\mathcal{M}, v)$ and $(\mathcal{M}', v')$ are $G$-*bisimilar *for a group $G \subseteq \mathcal{A}$ of agents, denoted $\mathcal{M}, v \leftrightarroweq_G \mathcal{M}', v'$, iff for every $a \in G$*

- ***G-Forth**: if $vR_au$, then there is $u' \in S'$ such that $v' R_a' u'$ and $\mathcal{M}, u \leftrightarroweq \mathcal{M}, u'$;*
- ***G-Back**: if $v' R_a' u'$, then there is $u \in S$ such that $vR_au$ and $\mathcal{M}, u \leftrightarroweq \mathcal{M}, u'$.*

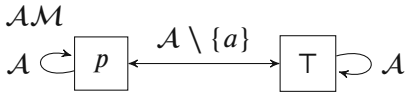



**Figure 1.4:** Action model $\mathcal{AM}$ corresponding to a semi-private announcement, where $a$ learns that $p$, while Everybody else learns that either $a$ learned $p$ or that $a$ learned nothing.

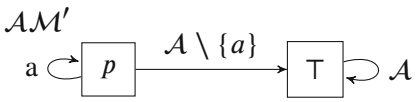



**Figure 1.5:** Action model $\mathcal{AM}'$ corresponding to a private announcement, where $a$ learns that $p$, while everyone else thinks that nothing happened. Note that $\mathcal{AM}'$ differs from $\mathcal{AM}$ of Figure 1.4 by the fact that relations are not required to be symmetric.

*The definitions of bisimilarity and G-bisimilarity for pointed action models are analogous.*

**Definition 25** (Equivalences)**.** *Pointed Kripke models* $(\mathcal{M}, v)$ *and* $(\mathcal{M}', v')$ *are called* modally equivalent*, denoted* $\mathcal{M}, v \equiv \mathcal{M}', v'$*, when it holds that* $\mathcal{M}, v \vDash \varphi$ *iff* $\mathcal{M}', v' \vDash \varphi$ *for all formulas* $\varphi$*. They are called* $G$-equivalent for group $G \subseteq \mathcal{A}$*, denoted* $\mathcal{M}, v \equiv_G \mathcal{M}', v'$*, whenever* $\mathcal{M}, v \vDash B_a\varphi$ *iff* $\mathcal{M}', v' \vDash B_a\varphi$ *for all formulas* $\varphi$ *and agents* $a \in G$*.*

**Theorem 26** (Bisimilarity implies modal equivalence)**.**

1. *If* $\mathcal{M}, v \leftrightarroweq \mathcal{M}', v'$*, then* $\mathcal{M}, v \equiv \mathcal{M}', v'$*.*
2. *If* $\mathcal{M}, v \leftrightarroweq_G \mathcal{M}', v'$*, then* $\mathcal{M}, v \equiv_G \mathcal{M}', v'$*.*

*Proof.* The first statement is standard (see [BdRV01]). The second statement follows immediately from the first: since $\mathcal{M}, v \equiv \mathcal{M}', v'$ then for all agents $a \in \mathcal{A}$, $\mathcal{M}, v \vDash B_a\varphi$ iff $\mathcal{M}', v' \vDash B_a\varphi$. Thus, if $\mathcal{M}, v \leftrightarroweq_G \mathcal{M}', v'$ they are also modally equivalent w.r.t. any set $G \subseteq \mathcal{A}$. □

**Definition 27** (Idempotence)**.** *We call an action model* $(U, \alpha)$ idempotent *iff the Kripke model resulting from the pointed update* $\big(\mathcal{M}\odot U,\ (w, \alpha)\big)$ *is defined, then also* $\Big((\mathcal{M}\odot U)\odot U,\ \big((w, \alpha), \alpha\big)\Big)$ *is defined, and the two are isomorphic for any pointed Kripke model* $(\mathcal{M}, w)$*.*

With action models, it is possible to represent complex communication scenarios, such as semi-private announcements and fully private announcements [BR16], shown respectively in Figure 1.4 and Figure 1.5. As for PAL, AML also has similar reduction axioms, making the proof of soundness and completeness almost identical to PAL [vDvdHK07]. In particular, both languages $\mathcal{L}^{PAL}$ and $\mathcal{L}^{AML}$ are exponentially more succinct than $\mathcal{L}$ [FvdHIK13].

Similarly to Kripke models and the modeling of epistemic puzzles (Remark 19), specific epistemic effect can be obtained via different action models.

We now introduce some extensions of AML that will be used for comparison with related work in future Chapters. For each of the logics introduced we briefly state some basic definitions

and results. Steaming from the idea that updates can be conducted via modifying accessibility relations, *Arrow Update Logic* (AUL) [KR11b] uses triples of the form $(\theta, a, \psi)$ to encode the information about the preconditions ($\theta$), the postconditions ($\psi$) and the agent(s) involved ($a$) in an action into the mechanism for deleting/generating arrows, generalizing AML.

**Definition 28** (Language of AUL [KR11b])**.** *For $p \in Prop$, $i \in \mathcal{A}$ and $\theta, \psi \in \mathcal{L}$, the language $\mathcal{L}^{AUL}$ is defined by the following BNF:*

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid K_i\varphi \mid [U]\varphi$$

$$U ::= (\theta, a, \psi) \mid (\theta, a, \psi), U$$

In particular, each triple $(\theta, a, \psi)$ is called an $a$-arrow specification with source condition $\theta$ and target condition $\psi$. A finite, comma-separated sequence $U$ of arrow specifications for one or more agents is called an *arrow update*. An arrow update $U$ describes the following Kripke model-transforming operation:

**Definition 29** (Semantics of Arrow Updates [KR11b])**.** *Given a Kripke model $\mathcal{M} = \langle W, R, V\rangle$, and an arrow update $U$:*

$$\mathcal{M}, w \vDash [U]\varphi \text{ iff } (\mathcal{M} * U), w \vDash \varphi \tag{1.8}$$

*where the model $\mathcal{M} * U = \langle W^*, R^*, V^*\rangle$ is defined as:*

- $W^* = W$;
- $R^* = \{sR_a^*s' \mid sR_as' \text{ and } \exists(\theta, a, \psi) \in U \text{ s.t. } \mathcal{M}, s \vDash \theta \text{ and } \mathcal{M}, s' \vDash \psi\}$; *and*
- $v \in V^*(p)$ *iff* $v \in V(p)$

Informally, the update procedure works as follows: every $a$-arrow from world $s$ and $s'$ are preserved if $U$ contains the triple $(\theta, a, \psi)$ such that $\theta$ is true at the source world $s$ and $\psi$ is true at the target world $s'$; all arrows that do not satisfy this condition are deleted. Interestingly, the public announcement of $\varphi$ can be represented in AUL using the following arrow update for all agents $a \in \mathcal{A}$: $U = (\top, a, \varphi)$. An important limitation of arrow updates is that the information communicated is assumed to be public, and the update is performed by all agents. This means that arrow updates generalize public announcements, but do not generalize action models. To cope with this non-common announcements, AUL has been extended to Generalized Arrow Update Logic (*GAUL*) in [KR11a]:

**Definition 30** (Generalized Arrow Update [KR11a])**.** *An arrow update $U$ is a pair $(O^U, \boldsymbol{a}^U)$ consisting of:*

- *a finite non-empty set $O^U$ of outcomes.*

- *an arrow function* $\boldsymbol{a}^U : \mathcal{A} \times O^U \to L \times O^U \times L$ *having a finite graph, with* $\boldsymbol{a}^U_a(o) =^{def} \boldsymbol{a}^U(a, o)$.

*The tuple* $(\varphi, o', \varphi') \in \boldsymbol{a}^U_a(o)$ *is an a-arrow with source o, source condition* $\varphi$*, target* $o'$*, and target condition* $\varphi'$*. An arrow update* $U$ *and an outcome* $o \in O^U$ *form a pointed arrow update* $(U, o)$ *with point o.*

An example of an arrow update is presented in Figure 1.6, representing the same communication scenario previously introduced for action models in Figure 1.5. Comparing the two figures, the main difference is that in AML the information about the precondition is encoded in the action point, while in AUL is encoded in the arrow, together with information about the precondition, allowing the modeling of more complex scenarios.

**Definition 31** (Language of GAUL [KR11a])**.** *The language* $\mathcal{L}^{GAUL}$ *is the smallest language defined inductively as follows: Let* $\mathcal{L}^0$ *be the language of multi-modal logic built up from atoms in Prop using negation* $\neg$*, conjunction* $\wedge$*, an epistemic modal operator* $K_a$ *for each* $a \in \mathcal{A}$*. Other Boolean and modal connectives are defined as usual. The language* $\mathcal{L}^{i+1}$ *is the set of formulas* $\varphi$ *formed by the grammar:*

$$\varphi ::= \psi \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid K_a\varphi \mid [U, o]\varphi, \tag{1.9}$$

*where* $\psi \in \mathcal{L}^i$*,* $a \in \mathcal{A}$*, and* $(U, 0) \in \mathcal{U}_*(\mathcal{L}^i)$*, where* $\mathcal{U}_*(\mathcal{L})$ *is the set of pointed arrow updates for* $\mathcal{L}$*. Finally, define the set* $\mathcal{L} =^{def} \bigcup_{i\in\mathbb{N}} \mathcal{L}^i$ *of formulas of GAUL, the set* $\mathcal{U} =^{def} \mathcal{U}(\mathcal{L})$ *of arrow updates, and the set* $\mathcal{U}_* =^{def} \mathcal{U}_*(\mathcal{L})$ *of pointed arrow updates.*

**Definition 32** (Semantics of GAUL [KR11a])**.** *Given a Kripke model* $\mathcal{M} = \langle W, R, V\rangle$*, the binary truth relation* $\vDash$ *between pointed Kripke models and formulas is defined by structural induction on the formula construction. The classical cases are defined as usual; the new case is defined as follows:*

$\mathcal{M}, s \vDash [U, o]\varphi$ *iff* $\mathcal{M} * U, (s, o) \vDash \varphi$*,*

*where* $\mathcal{M} * U$ *is defined as follows:*

- $W^{\mathcal{M}*U} := \{(s, o) \in W \times O^U\}$
- $R_a^{\mathcal{M}*U}((s, o)) := \{(s', o') \in R_a(s) \times O^U \mid \exists(\varphi, o', \varphi') \in \boldsymbol{a}^U_a(o) : (\mathcal{M}, s \vDash \varphi \text{ and } \mathcal{M}, s' \vDash \varphi')\}$
- $V^{\mathcal{M}*U}(p) := V(p) \times O^U$

Similarly to the logics introduced before, there are reduction axioms for AUL and GAUL. [KR11a] shows that both the language $\mathcal{L}^{AML}$ and the language $\mathcal{L}^{GAUL}$ are equally expressive and equally update expressive.

Many of the dynamic logics introduced so far have been extended by using quantification over announcements. We introduce the basic definitions of a few of these logics, starting from *Arbitrary Public Announcement Logic* (APAL) [BBvD+08].

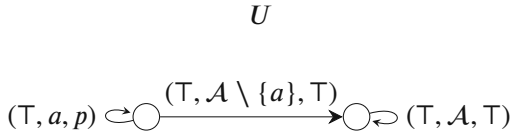



**Figure 1.6:** Arrow update model $U$ where $a$ secretly learns that $p$, while everybody else thinks that nothing happened.

**Definition 33** (Language of APAL [BBvD$^+$08])**.** *The epistemic language with arbitrary public announcements $\mathcal{L}^{APAL}$ is defined by:*

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid K_i\varphi \mid [\varphi]\varphi \mid [!]\varphi$$

*where $p \in Prop$ is an atomic proposition and $i \in \mathcal{A}$.*

**Definition 34** (Semantics of APAL)**.** *We describe only the new case, as the remaining is akin to PAL. Given a model $\mathcal{M} = \langle W, R, V\rangle$, $s \in W$, $\varphi \in \mathcal{L}^{APAL}$ and $\psi \in \mathcal{L}^{PAL}$:*

$$\mathcal{M}, s \vDash [!]\varphi \text{ iff } \forall\psi \in \mathcal{L}^{PAL} : \mathcal{M}, s \vDash [\psi]\varphi \tag{1.10}$$

*and the dual:*

$$\mathcal{M}, s \vDash \langle!\rangle\varphi \text{ iff } \exists\psi \in \mathcal{L}^{PAL} : \mathcal{M}, s \vDash \langle\psi\rangle\varphi \tag{1.11}$$

The operator $[!]\varphi$ is very powerful as it quantifies over formulas with all propositional variables (even those not explicitly present in $\varphi$) and over formulas of arbitrary finite modal depth.

The following theorem states that APAL is more expressive than PAL (and so more expressive than standard epistemic logic):

**Theorem 35** ([BBvD$^+$08])**.** *Over the class of all pointed S5 Kripke models with agent set $\mathcal{A}$:*

- *if $\mid \mathcal{A} \mid = 1$ then $\mathcal{L}^{APAL}, \mathcal{L}^{PAL}$ and $\mathcal{L}$ are equally expressive;*
- *if $\mid \mathcal{A} \mid > 1$ then $\mathcal{L}^{APAL}$ is strictly more expressive than both $\mathcal{L}^{PAL}$ and $\mathcal{L}$.*

A similar generalization can be provided for AML, resulting in the logic AAML [Hal13] that we briefly introduce now. AAML makes use of *action signatures*:

**Definition 36** (Action signatures [BM04])**.** *An action signature, $\Sigma = (S, R, (s_1, \dots s_n))$ consists of an underlying Kripke frame $\mathcal{F} = (S, R)$ along with a non-repeating list of non-trivial actions $s_1, \dots s_n \in S$. Given a logical language $\mathcal{L}$ such that $\top \in \mathcal{L}$ and a list of formulas $\varphi_1, \dots, \varphi_n \in \mathcal{L}$, it is possible to obtain an action model $\Sigma(\varphi_1, \dots, \varphi_n) = (S, R, pre)$, with preconditions defined on $\mathcal{L}$, where $pre(s_i) = \varphi_i$ for $i = 1 \dots n$ and $pre(s) = \top$ otherwise.*

*We denote the set of all signatures with $\mathcal{S}$, so that $\Sigma = (S, R, (s_1, \dots s_n) \in \mathcal{S}$. Given a set $T$ of signatures $T = (s_1, \dots s_n) \subseteq \mathcal{S}$, we use abbreviations $[M_T]\varphi = [\Sigma T, \varphi_1 \dots \varphi_n]\varphi$ where $M = \Sigma(\varphi_1, \dots, \varphi_n)$; and if $T = \{s\}$ then we write $[M_s]\varphi$.*

*The semantics of action signatures is defined as:*

$$\mathcal{M}, s \vDash [M_s]\varphi \text{ iff } \mathcal{M}, s \vDash pre(s) implies (\mathcal{M}, s) \otimes M_s \vDash \varphi; \tag{1.12}$$

$$\mathcal{M}, s \vDash [M_T]\varphi \text{ iff for every } s \in T : \mathcal{M}, s \vDash [M_s]\varphi \tag{1.13}$$

**Definition 37** (Language of AAML [Hal13])**.** *The language $\mathcal{L}^{AAML}$ is inductively defined as:*

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid K_i\varphi \mid [\Sigma T, \varphi_1 \dots \varphi_n]\varphi \mid \forall_B\varphi$$

*where $p \in Prop$ $i \in \mathcal{A}$ and $B \subseteq \mathcal{A}$*

The formula $\forall_B\varphi$ may be read as every action model (that is a $B$-refinement) results in becoming true, where refinements are defined as follows:

**Definition 38** (Refinements [Hal13])**.** *Let $B \subseteq \mathcal{A}$, $\mathcal{M} = \langle W, R, V\rangle$ and $\mathcal{M}' = \langle W', R', V'\rangle$. A non empty relation $\mathcal{R} \subseteq W \times W'$ is a B-refinement from $\mathcal{M}$ to $\mathcal{M}'$ iff for every $p \in Prop$, $a \in \mathcal{A}$, $c \in \mathcal{A} \setminus B$ and $(s, s') \in \mathcal{R}$ the conditions atoms-p, forth-c and back-a holds:*

- *atom-p $s \in V(p)$ iff $s' \in V'(p)$;*
- *forth-c for every $sR_ct$ there is a $s'R'_ct'$ such that $(t, t') \in \mathcal{R}$;*
- *back-a for every $s'R'_at'$ there is a $sR_at$ such that $(t, t') \in \mathcal{R}$.*

*If there exists a B-refinement from $\mathcal{M}$ to $\mathcal{M}'$ we denote it as $\mathcal{M} \geq_B \mathcal{M}'$. The same definition holds for pointed models, and for the cases of $B = \mathcal{A}$ and $B = \{a\}$.*

**Definition 39** (Semantics of AAML [Hal13])**.** *We describe only the new case, as the remaining is akin to AML. Given a model $\mathcal{M} = \langle W, R, V\rangle$, $s \in W$,*

*$\mathcal{M}, s \vDash \forall_B\varphi$ iff for every $M_s \in \mathcal{S}$ s.t. $pre \subseteq \mathcal{L}$ if $\mathcal{M}, s \vDash pre(s)$ and $(\mathcal{M}, s) \geq_B (\mathcal{M}, s) \otimes M_s$ then $(\mathcal{M}, s) \otimes M_s \vDash \varphi$*

[Hal13] proves that $\mathcal{L}^{AAML}$ and $\mathcal{L}^{AML}$ are equivalent expressive for formulas in $\mathcal{L}^{AML}$, and that The logic $AAML_{S5}$ is expressively equivalent to the logic S5.

Last, but not least, AUL can also be extended with quantification over updates, resulting in the Arbitrary Arrow Update logic AAUL:

**Definition 40** (Language of AAUL [vvKK17])**.** *For $p \in Prop$ and $i \in \mathcal{A}$, the language for AAUL $\mathcal{L}^{AAUL}$ is defined by the following BNF:*

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid K_i\varphi \mid [a]\varphi \mid [U]\varphi \mid [\updownarrow]\varphi$$

$$U ::= (\theta, a, \psi) \mid (\theta, a, \psi), U$$

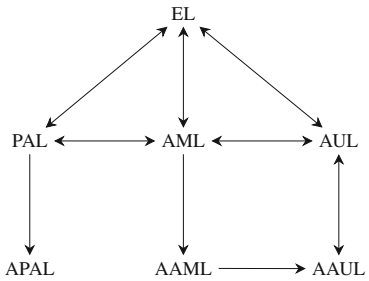



**Figure 1.7:** [vvKK17]. Expressivity comparison among the dynamic logics considered, over the class of all Kripke models. EL corresponds to the standard epistemic logic. An arrow from one logic to another means that the target logic is at least as expressive as the origin.

**Definition 41** (Semantics of AAUL [vvKK17])**.** *Given a Kripke model $\mathcal{M} = \langle W, R, V \rangle$, the binary truth relation $\vDash$ between pointed Kripke models and formulas is defined by structural induction on the formula construction, where the new case is defined as follows:*

$\mathcal{M}, s \vDash [\updownarrow]\varphi$ *iff* $\mathcal{M} * U, (s, o) \vDash [U]\varphi$*, for each* $U \in \mathcal{L}^{AUL}$*,*

*where $\mathcal{M} * U$ is defined as follows:*

- $W^{\mathcal{M}*U}\{(s, o) \in W \times O^U\}$
- $R_a^{\mathcal{M}*U}((s, o)) := \{(s', o') \in R_a(s) \times O^U | \exists(\varphi, o', \varphi') \in \boldsymbol{a}_a^U(o) : (\mathcal{M}, s \vDash \varphi \text{ and } \mathcal{M}, s' \vDash \varphi')\}$
- $V^{\mathcal{M}*U}(p) := V(p) \times O^U$

[vvKK17] proves that APAL and AAUL are incomparable in expressivity for a finite set of agents and a countably infinite set of propositional variables. In S5, for the single agent case, $\mathcal{L}$ is equally expressive to $\mathcal{L}^{AAUL}$, for more than one agent, the latter is more expressive. An overall expressivity comparison of the dynamic epistemic logics exposed in this section can be found in [vvKK17], and is reported here in Figure 1.7.

### 1.2.6 Related work in DEL

In this section we further discuss three other approaches using dynamic epistemic logic. We will use the content of this section mainly as a comparison with the existing literature in the following chapters.

**Recovering from false beliefs** A key limitation of the standard DEL framework is that agents cannot meaningfully assimilate information that contradicts their knowledge or beliefs. For instance, if an agent knows $p$, and is presented with a public announcement of $\neg p$ (assumed to be truthful), the agent's epistemic state becomes inconsistent. In such cases, the agent is effectively said to "know everything," including contradictions. This inconsistency arises because, under the standard semantics, the announcement eliminates all worlds in which $p$ holds, leaving the agent

with no accessible worlds that satisfy the precondition $\neg p$. As a result, the agent's knowledge collapses, and the model fails to represent a coherent epistemic state.

To address such issues, a substantial body of work within DEL focuses on recovering from false or inconsistent epistemic states. Inspired by the tradition of belief revision [AGM85], one well-established approach involves the use of plausibility models [BS06, BS16]. The central idea behind plausibility models is to enrich the Kripke structure with agent-specific plausibility orderings over possible worlds, representing which worlds an agent considers more plausible. Accordingly, an agent believes a proposition $p$ if $p$ holds in all of the most plausible worlds. When the agent receives new information (e.g., an announcement of $\neg p$), rather than eliminating all worlds where $p$ holds (as in public announcement logic), the plausibility ordering is revised so that the $\neg p$-worlds become more plausible than the $p$-worlds. This allows for belief revision without inconsistency or trivialization.

**Update model synthesis** Another important question in DEL is whether it is possible, given an input model and an output model, to find a corresponding action model that produces some desired output. This is precisely the scope of the update *synthesis* task, where the aim is to (i) find whether there exists an update model that makes a given goal formula $\varphi$ true and (ii) construct that update model from $\varphi$ [DDvdHKK20]. Because of the strength of the desiderata of this goal, there is no general guarantee that synthesis is possible. For example, in PAL and AUL it is not possible in general. However, [Hal13] proved that it is possible for AML.

Existing synthesis methods typically work in a language extended with quantifiers over updates, such as the AAML [Hal13] or AAUL [DDvdHKK20] introduced above.

**Epistemic planning** Epistemic planning is concerned with the question of finding, given the current epistemic state of an agent and a target epistemic state, a sequence of actions (also called a *plan*) that leads from the former to the latter [Bol17]. This problem has been extensively studied using tools from DEL, where each action, along with its associated uncertainty in multi-agent contexts, is represented by an action model. The effects of these actions, both epistemic and ontic (on the physical world), are then evaluated in the resulting updated model. Naturally, the core challenge of identifying, for a given plan $P$ within a class of plans, a solution, that is, a sequence of actions that successfully achieves the plan's goal, is closely connected to the action model synthesis problem.

In Chapter 4 we will introduce an update synthesis method to recover from false/unwanted beliefs in the context of fault-tolerant distributed systems. Instead of relying on plausibility orders [BS16] we rely on standard DEL by re-setting the belief of agents according to a given goal formula. These updates are intended as global a priori updates initiated by the system designer aiming at modifying the (a priori) belief of some agents without rebooting the whole system. In Section 4.4 we will compare our update mechanism with the ones introduced in this section.

## 1.3 Philosophy: a priori knowledge

The notion of "a priori" is familiar to most people from Kant's famous work, but its history goes back further, at least to the Latin translations of Euclid's Elements. During this long history, the

distinction between a priori and a posteriori knowledge has seen many developments and has been the subject of an enormous amount of quite polarizing discussions. Besides knowledge, the debate also concerns related topics such as modality [Kri72, Put78], justification [Bon97, Cas03], revisability [Kit80][Bur93, Bur98], and many others; see [Cas14] for a comprehensive overview of the debate. As claimed by [Cas03], the relevance of the dispute goes beyond epistemology and reaches virtually any area of philosophy and the notions of "a priori" and "a posteriori" have been applied widely. In particular, a novel view of a priori knowledge was proposed by Tahko [Tah08, Tah11] in the context of a broader metaphysics of science (cf. [Tah15]). Along the same lines, we argue that these notions can also be put to fruitful use in the field of distributed computing systems, where a priori knowledge plays a crucial role both in their design process and their epistemic analysis.

### 1.3.1 A modal and fallible approach to a priori knowledge

In the standard view, the concepts of a priori and a posteriori are mutually defined by their opposite relation with respect to the notion of experience: the former is defined as independent of experience, while the latter relies on experience [Kit80, p.4]. However, the distinction between the two notions is not as sharp as it might seem at first, and influential philosophers [Qui51, Wil13] have problematized it. For instance, Williamson observes a problem with the fact that, traditionally, every specific way of knowing has been taken to be either a priori or a posteriori, and not both. This introduces a problem, because there are cases where we seem to be able to come to know the same proposition both by a priori and a posteriori means. Accordingly, we would need to refine the distinction if we don't want the same piece of knowledge to be both a priori and a posteriori. Williamson speculates that 'the best fit to current practice with the term is to stipulate that a truth is a posteriori if and only if it can be known a posteriori but cannot be known a priori' [Wil13, p.293]. But as Williamson goes on to point out, Kripke's [Kri72] famous examples of contingent a priori truths and necessary a posteriori truths may cause further problems here. In particular, if there are contingent a priori truths, such as Kripke's example concerning the standard meter, then it seems that they must also be knowable a posteriori. This severs the link between apriority and necessity. The upshot for Williamson is that the differences between a priori and a posteriori knowledge are superficial because experience plays a role for both kinds of knowledge, and he argues that this role is not just 'purely enabling'—like it would perhaps be in the case of concept acquisition—while also being less than 'strictly evidential', such as in the case of pure perceptual information.

We accept Williamson's challenge as it applies to the traditional distinction between a priori and a posteriori knowledge and acknowledge that this requires a revision of these concepts. However, the discussion has grown enormously and proponents of the distinction have related it to other notions rather than to experience. Among these, Tahko [Tah08, Tah11] claims that the key concept grounding the a priori is "modality"; more precisely, metaphysical modality (i.e., possibility and necessity). In light of Kripke's argument against the link between apriority and necessity, this move may seem ill-advised, as we have just seen with reference to Williamson. Instead, Tahko posits that Kripke's analysis does not rule out the connection between apriority and metaphysical modality in general, but only between apriority and necessity. In his view,

the a priori and metaphysical modality can still relate via the notion of possibility. So, a priori reasoning concerns metaphysical possibilities. In Tahko's view, this enables an analysis of a priori and a posteriori knowledge that retains some aspects of the traditional conception while giving up the sharp distinction in terms of experience that Williamson focuses on.

Viewed more abstractly, a priori reasoning operates in the realm of metaphysical possibility and determines the range of different possible scenarios that are consistent with the data currently at hand. In Tahko's account, a priori reasoning is not supposed to be mere conceivability, nor is it based on intuitions or any special rational capacity. Rather, a priori reasoning is considered a fairly mundane capacity that rational agents have, closely analogous to the capacity to formulate a scientific hypothesis. Accordingly, this is especially fitting for the process of scientific discovery, to which Tahko explicitly refers [Tah11, Tah20], where the aim is to determine what is true in the actual world. In this process, the role of the a priori is to determine the range of possibilities that need to be tested by empirical means. We do not need to take a stand here regarding the nature of a priori reasoning, as it is sufficient for the present application to consider the much more restricted case of processes (agents) of a distributed system, and such agents gain their initial a priori "knowledge" more directly from the system designer. The definition of apriority that emerges from Tahko's account is something like the following (after Tahko, [Tah08, p.60]):

**Definition 42** (A Priori Knowledge)**.** *Any metaphysically possible, logically consistent a priori proposition may constitute 'a priori knowledge', whether or not it is true in the actual world.*

The modal element of Tahko's view should be clear from this, but what about the fallibility? In fact, Tahko suggests that there are two senses of fallibility at play here; one is human error, in which case the proposition wasn't really a priori to begin with (say, because it was not consistent). Whereas this is, of course, a source of potential error, we consider it as something that we can safely ignore in our considerations as (i) it is way less likely to occur compared, e.g., to the wealth of possible programmer's errors and (ii) nothing could be done about it anyway. On the other hand, insofar as the proposition is a priori, it would always correspond with a metaphysical possibility. So, in the latter case, the fallibility just concerns the question of whether the metaphysical possibility captured by the proposition is also actual (or factive). It is this aspect of Tahko's account that turns out to be the most useful for our application, as we will detail in Chapter 2: the agents in a system may be considered to be fallible in a structurally similar sense. What this means in the context of Tahko's account is that we never falsify the a priori metaphysical possibility, but we may falsify its actuality (say, because new empirical evidence rules out something we thought might be actual but was merely possible). In this thesis we will focus on this latter sense of fallibility, as it is concerned with a priori fallible propositions and not with a posteriori propositions mistakenly considered a priori because of human error.

A key upshot of Tahko's framework is that there is a constant interplay between the a priori and the a posteriori, called a bootstrapping relationship[Tah11, p.152]: "A priori knowledge generally advances in very small steps: we introduce an a priori proposition, which we then attempt to verify by a posteriori means; this is the core of the bootstrapping relationship". Tahko's main

example of this relation concerns the phenomenon of gravitational redshift[14] in the history of science [Tah11, p.152]. Gravity's effect on light was a phenomenon envisioned by Newton's theory, but the results were inaccurate due to Newton's assumptions about the corpuscular nature of light. By assuming the wave nature of light and the possibility of change in time flow, Einstein's general theory of relativity, on the other hand, was able to obtain more accurate predictions. In a nutshell, Tahko claims that the dynamics behind this process involve a priori reasoning and a posteriori verification.

The process is represented in Figure 1.8. The first step, at the bottom, is the a posteriori basis representing all the knowledge about the gravitational redshift before Newton, together with the respective assumptions on the nature of light. Newton's theory of gravitational redshift represents an a priori step in the bootstrapping sequence that introduces a new hypothesis to the framework. Effectively, an a priori step is the introduction of a proposition that is believed to be metaphysically possible. This hypothesis is then tested against the a posteriori basis: as a result, it is established that the hypothesis provides a better explanation of physical phenomena (although it fails to coherently explain all of them). Newton's theory hence is incorporated into a new a posteriori basis, becoming part of the adopted a priori beliefs about the actual world. This is the process by which Tahko suggests knowledge to accumulate, i.e., a verified a priori proposition becomes part of the established a posteriori basis. Einstein's work constitutes the next a priori step, starting out from the established post-Newton a posteriori basis, and once empirically verified, it is incorporated into a new (and more sophisticated) a posteriori framework.

This presentation is, of course, a simplification of the real process and does not include all the single instances of the bootstrapping relation (i.e., the many hypotheses and experiments conducted by different people). Nonetheless, it shows the interplay between the a priori propositions intended to explain a certain phenomenon and the a posteriori framework that effectively puts these a priori hypotheses to a test. As pointed out by Tahko [Tah08, p.66], the process of a priori reasoning is not simple guessing because the new a priori propositions are based on the a posteriori framework at hand: their aim (at least in a scientific context) is to explain phenomena that contradict the actual a posteriori basis. Note that this does not make these a priori propositions empirical or a posteriori. Rather, in a scientific context, we generally (although not always) delimit our attention to those a priori propositions that are compatible with the existing a posteriori basis. If the new a priori propositions prove to be (partially) correct by removing (some of) these contradictions, they can enrich the a posteriori framework, providing the basis for further development.

Given that a priori propositions are evaluated in the actual world, they are always susceptible to a falsification at a later stage: at any time, new contradictions can be discovered that invalidate (part of) the current a posteriori basis. Hence, as noted above, and following BonJour's account [Bon97], Tahko claims that a priori reasoning can be fallible: a verified—or, better, not falsified—a priori proposition does not need to be metaphysically necessary: it is always conceivable that we may discover a new phenomenon falsifying it in the actual world. On the other hand, according to Tahko's modal interpretation of a priori knowledge, the falsification of an a priori proposition cannot change its a priori status. Indeed, were a priori propositions necessary, then a falsification

[14] Gravitational redshift refers to the change in the wavelength of light and other electromagnetic radiation when it travels from a stronger gravitational field to a weaker one [Tah11, p.152].

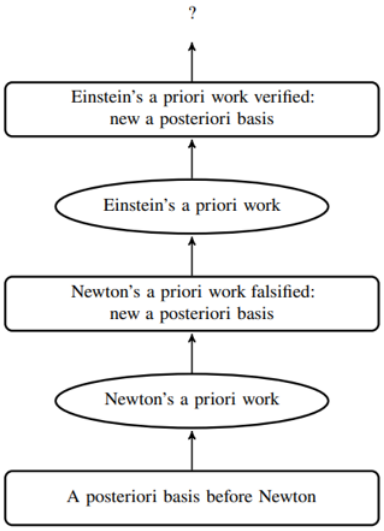



**Figure 1.8:** The bootstrapping relation.

of one would have proven that it was not a priori in the first place. This contradiction disappears if, instead of being considered necessary, a priori propositions are viewed as restricting the set of possible scenarios. A falsified a priori proposition still retains its a priori status but has been shown not to hold in the actual world. At the same time, a proposition that is mistakenly considered a priori can be dismissed by showing that the proposition cannot be possible. Note that the verification process itself can be purely empirical: testing the actuality of a proposition is done solely by a posteriori means, without questioning the status of the proposition, but only assessing its status in the actual world.

We are not arguing here in favor of this particular view of a priori knowledge in the general philosophical context considered by Tahko. Our goals are more modest: we only claim that his notion of a priori knowledge and the bootstrapping process are well-suited for capturing the concept of a priori knowledge that is embedded in the epistemic reasoning in man-made systems. The resulting notion of a priori may also be fruitfully compared to Burge's [Bur93, Bur98, Bur10, Bur20] influential work in developing the notion. Burge famously argued in favor of a moderate form of rationalism, whereby rationalism is not about 'unrevisability, infallibility, indubitability, or innateness' [Bur98, p.2], but rather about the force of epistemic warrant. This produces a notion of apriority that concerns justification or entitlement; an agent's knowledge is a priori if it is supported by an a priori warrant that needs no further warrant for it to be knowledge.

This view has various interesting features that are compatible with the notion of apriority adopted here. In particular, Burge is adamant that a belief can have both empirical and a priori justification or entitlement and argues that a "justification or entitlement is apriori if its justificational force is in no way constituted or enhanced by reference to or reliance on the specifics of some range of sense experiences or perceptual beliefs" [Bur93, p.458]. This allows for considerable flexibility on what counts as a priori, and the approach is broadly compatible with Tahko's account of the a priori. While Burge does not put forward a modal analysis of apriority, his work has helped to

disentangle the notion from its traditional connotations involving unrevisability and infallibility.

These are also key features of the notion of apriority that we employ. Moreover, Burge's notion of apriority has also laid the path for a more natural use of the notion in contexts that involve non-human agents. For instance, Burge [Bur98, p.4] allows that "knowledge that an individual obtains by being told a proposition by another person, where the individual's warrant resides in the interlocution, can be apriori". In the present context, the a priori knowledge of the agents comes directly from the system designer, so we obviously need a notion of apriority that is flexible in a similar fashion as Burge's notion. Fortunately, Tahko's approach provides a similar flexibility, with the addition of the modal element that turns out to be useful for our purposes.

## 1.4 Outline of the thesis

The content of this thesis is articulated in four main chapters:

- Chapter 2 builds upon [CSTK23] and its dedicated to the development of the philosophical framework of a priori beliefs in distributed systems, drawing from the recent philosophical account proposed by Tahko about a priori knowledge [Tah08, Tah11]. In this chapter, we introduce the key notions of a priori belief and a priori belief updates (Section 2.1.2), describing their properties based on the kind of agents initiating it (system designer, processes (or agents) and designer agents). The chapter also establishes an informal connection between epistemic logic and the notions of a priori/ a posteriori beliefs (Section 2.1.4), that will be formalized in later chapters.

- Chapter 3, steaming from [CKRS23], proposes a relaxation of the assumption of common a priori knowledge of the protocols, introducing a logic suitable for the epistemic analysis of heterogeneous distributed systems. In particular, we divide the set of agents into types, each corresponding to a different protocol being executed, and associate to each type pair an interpretation function that allows agents to interpret information from agents of the other type. The two elements are combined in the creed modality, which expresses knowledge of agents conditional to the type of the sender in communication scenarios. We prove soundness and completeness of the logic.

- Chapter 4 is based on [SKC24] (under review), and it introduces a mechanism to model global a priori belief updates via synthesis of a pointed action model from a goal formula representing the target update from the system designer. We introduce the notion of pointed update that reduces the complexity of the resulting model, and we introduce a novel synthesis method producing stratified action models that preserve the beliefs of non-updated agents and change the (higher order) beliefs of only the agents involved in the goal formula. As a result, the update mechanism preserves consistency of agents whenever possible and does not always increase the size of the resulting model exponentially.

- Chapter 5, based on the work in [CK24], presents the main contribution of this thesis: the introduction of local a priori updates, initiated autonomously by agents in self-adaptive self-organizing (SASO) systems in response to unexpected behavior. The purpose of

these updates is to restore the epistemic consistency of the updating agent. We prove that this update mechanism preserves the beliefs of agents not involved in the update, while successfully restoring consistency for the updating agent. Additionally, we provide heuristics and formal guidelines for how this process can be systematically described, once the initial space of possible updates has been defined.

Finally, an recollection of the main accomplishments of the thesis is provided in Chapter 6.

CHAPTER 2

# The role of a priori knowledge in the design and analysis of distributed systems

This chapter[1] is dedicated to the connection between the philosophical notion of a priori knowledge and distributed systems. The overall goal of this chapter is to introduce the philosophical preliminaries for a formalization of a priori knowledge in distributed systems exposed in the subsequent chapters. In turn, these results can be used to improve both the design and analysis of distributed systems. We have already stressed in the introduction that designing distributed systems can be extremely challenging due to the inherent uncertainty of any process about the many system parameters, such as actual computation speeds and message delays, to name a few. The design of such a system inevitably involves the need to come up with a set of system assumptions (like message delay bounds and failure rates) that are assumed to hold throughout the lifetime of the system. This brings us to the first connection between the philosophical literature and the epistemic analysis of distributed systems, since the relevant system assumptions are best understood as holding "a priori", in the sense that they are assumed to hold irrespectively of any execution of the system. This is of course not quite the usual sense of a priori knowledge familiar from philosophical literature, i.e., knowledge that is independent of experience. But as we will see in this chapter, there are striking structural similarities between this applied notion of the a priori and the philosophical interpretation of a priori knowledge presented in Section 1.3.

Unlike in the philosophical framework, where a priori knowledge is ascribed to a human agent, in the context of distributed systems, we have to distinguish between two kinds of agency, namely:

[1]The first part of this chapter is based on joint work with Tuomas Tahko, Ulrich Schmid and Roman Kuznets, [CSTK23].

- the system designer, the idealized (traditionally human) designer of the system, who is responsible for its development and well-functioning, and
- the processes (or agents) of a distributed system that are computing devices with only a limited view of the system (compared to the view the system designer has); agents acts according to their protocol (unless faulty), which is usually designed by the system designer.

Hereafter we will use the term "agent" exclusively to refer to the processes of a distributed system, unless specified otherwise. Obliviously, there is a huge gap between the knowledge of agents, as opposed to the system designer, both in its a priori and a posteriori versions, at least in traditional distributed systems. On the one hand, the (ideal) system designer has a global view of the system, which means that she knows what is the current execution of the system under consideration and what are the parameters and assumptions that determine this execution. Agents, on the other hand, are bounded by their protocols (which are usually commonly known) and act based on their limited local view of the system. Analyzing the system in a possible worlds setting [FHMV95], we could say that agents are bounded to the sets of possible states present in the model. This set of states is determined by the protocols of agents and thus by system assumptions, which are, in turn, a product of the system designer.

Obviously, the system designer is also able to modify the system at hand, by changing system parameters and/or system assumptions; these modifications are reflected in the system by modifying the set of possible states that agents might reach during a certain execution. Importantly, agents only passively receive these changes, and they might not even be aware of them if the system designer chooses so, e.g. in a complete system reboot. Under this light, the set of system assumptions held by the system designer can be seen as the core of the system designer's a priori knowledge. While this usage of the notion of "a priori" is non-standard in contemporary epistemology, we show that the notion is well fitted for describing the design process and the epistemic understanding thereof in distributed systems.[2]

Naturally, the system designer's a priori knowledge also determines the a priori knowledge of the agents, as we will argue in later sections. Furthermore, despite some differences, the two agencies share the same notion of a priori knowledge, which is modal and fallible (Section 1.3). Deviating from existing literature, we hence use the term a priori "belief", combining Tahko's interpretation of fallibility and the usage of the term in epistemic logic, where belief encapsulates falliblism by lacking factivity (Section 1.2), while preserving its modal connotation related to possibility.

On the other hand, the knowledge (resp. belief) that is obtained by the agents during protocol execution will be called "a posteriori knowledge"[3] (resp. a posteriori belief). Thus, we use the term "a posteriori knowledge" interchangeably with "epistemic knowledge", which is used in the field of epistemic logic. The corresponding notion for the system designer will be introduced later in this chapter.

[2]What we propose is a sort of "conceptual engineering", i.e., fixing or adapting an old concept for a new application [Cha20].

[3]Whereas it is typically viewed from the system analyzer's perspective, a posteriori knowledge also forms the basis of the agents' actions, as pointed out by the Knowledge of Preconditions principle [Mos15].

One of the main advantages of explicitly incorporating a priori beliefs in the epistemic analysis of fault-tolerant distributed systems is that it allows to deal formally with SASO systems: in traditional distributed systems (fault-tolerant or not), agents lack any ability to react to violations of the system assumptions, which is typical of design mistakes. Only the system designer can, upon detecting such violations, re-design the system accordingly. On the other hand, agents in SASO systems are able to detect and to dynamically react to design mistakes autonomously, thanks to their abilities to deal with the system assumption via a priori reasoning and a priori belief updates. This autonomous use of a priori beliefs allows agents in SASO systems to self-adapt to unexpected situation by correcting themselves and modifying the system assumptions. These insights culminate in Chapter 5, where we introduce a mechanism for self-recovery from inadequate system assumptions via a priori updates.

## 2.1 A priori beliefs in traditional distributed systems

In Section 1.3, we discussed Tahko's fallibilistic view of a priori knowledge, which is based on the notion of possibility and is in a constant interplay with a posteriori knowledge by means of the bootstrapping relation. In this section, we will provide a similar interpretation of a priori beliefs in the process of designing distributed systems and demonstrate that many aspects of his view fit very well to our specific context. We maintain the modal aspect of Tahko's definition: In the present context, the modal space is determined by the set of alternative scenarios considered possible by the agent in question.

We will start with preliminary definitions[4] of the terms and concepts used that will be further elaborated on later. We sometimes abbreviate "a priori beliefs" as "APrBs". We start by distinguishing the system designer's a priori beliefs from the system designer's a posteriori beliefs/knowledge.

**Definition 43** (System Designer A Priori Beliefs (APrBs))**.** *Any proposition that is meant to describe some key features of the system under consideration may constitute 'a priori belief', whether it actually holds or not in the current system. It comprises system assumptions, such as the basic safety and liveness specifications, and system features, such as agents' characteristics, network topology, failure models, communication delays, encoding of messages etc., but also background beliefs of the system designer. Importantly, it is always possible that these system assumptions and features may be violated in a real system, i.e., they are fallible.*

**Definition 44** (System Designer A Posteriori Belief/Knowledge)**.** *Any proposition relative to the system at hand that can be inferred from an execution (or a simulation) of the system itself.*

The System Designer APrBs need not be clearly and formally formulated: especially during the early phases of the design process, the system designer has only fuzzy and informal concepts in mind, and the desired result is reached by trial and error. Also, her a posteriori knowledge needs not to be formally defined, although it can, e.g. using epistemic logic and/or other specification languages.

[4] Given the philosophical nature of this chapter, the definitions introduced here will not contain any formalism. Nonetheless, we use the definition environment to add clarity to the concepts used.

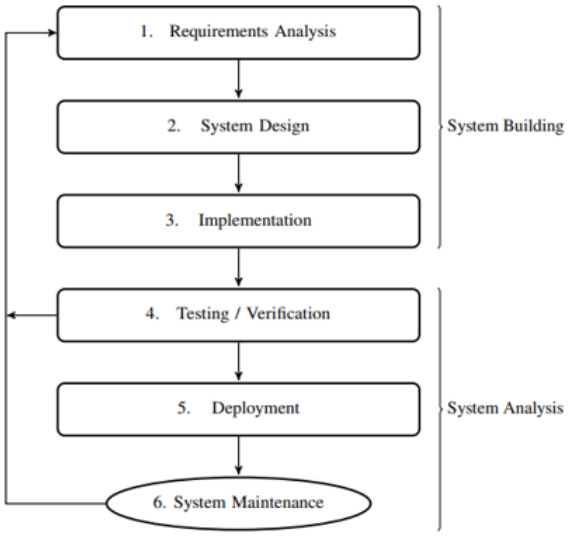



**Figure 2.1:** Design of a traditional distributed system.

To exemplify the distinction, consider an ATM system in the phase of design. A basic a priori belief of the system designer, in the form of a system assumption, is that such a system provides money to the client if some conditions are met. Thus, the proposition "if the agent knows that the conditions are met, provide the requested amount" can be thought of as an a priori proposition that will be implemented in the system. Suppose the system is functioning. The system designer observes that some agents misbehave, i.e., the conditions are met but the money is not provided. This information constitutes a posteriori knowledge for the system designer as it is a fact inferred from the behavior of the system. This information also contradicts the system designer's a priori belief, as it violates the proposition stated above, justifying the usage of the term "belief" instead of "knowledge".

Intuitively, the definition of a posteriori knowledge/belief is the classical one, that is, knowledge/belief gained from experience: the main difference is that experience in distributed settings means experience of the system execution. On the other hand, a priori beliefs cannot be inferred from the system execution: system parameters and assumption determines the execution, but are not learned during it. One might learn that some assumptions are true or false during execution, but the falsification/verification does not alter its a priori status, as for the broader philosophical framework exposed in Section 1.3.

### 2.1.1 System design overview

The fundamental role played by the system designer's a priori beliefs is better captured by considering the process of designing a distributed system from scratch. Figure 2.1 is an abstract representation of the traditional system engineering approach in the design of distributed systems. In the early phase of the process, called System Building Phase, the goal is to provide a set of properties that the system must satisfy, collectively termed system assumptions for simplicity.

Their formulation is the aim of the Requirements Analysis step (1.), which is arguably the most abstract and also the least formalizable part of the design process. The system designer considers which properties (e.g., safety and liveness specifications) the system shall satisfy and the assumptions and constraints under which they are to be guaranteed. For example, an ATM system will have to respond correctly to client requests, providing money whenever certain conditions are met. Ideally, all these properties and conditions are formulated precisely, completely, and consistently. In reality, however, these specifications are but ideas that shape the system in the very process of making. There are various approaches to requirements engineering, which are typically ad hoc and vary greatly both in their methodology and outcome. Once the specifications are sufficiently formulated, the actual building of the system begins, usually starting out from higher abstraction levels (System Design, 2.) to lower abstraction levels (Implementation, 3.). Typically, design approaches are less ad hoc and often supported by tools, in particular, for the automatic translation (compilation) from higher to lower abstraction levels. It is obvious that both the System Design and Implementation steps heavily rely on the Requirements Analysis step: if the requirements turn out to be inadequate, then both the high level design and the implementation will have to change accordingly. Spotting such discrepancies, along with the obvious strive for eliminating design and implementation errors, is in the scope of the Testing/Verification step (4.). Here, we enter the System Analysis Phase that checks whether the system actually satisfies the desired properties. Ideally, automatic verification techniques, such as model checking, are used to formally prove that the implemented system satisfies the required properties. In practice, however, manual testing must usually be resorted to.

After the implementation has been tested and the requirements specifications have been verified, which typically involves several iterations of the loop, the system is finally deployed and put into operation (Deployment, 5.). What were "ideal" requirements in the system designer's mind are now implemented in a concrete system, the behavior of which is usually monitored in some way. In the final System Maintenance step (6.), errors or other deficiencies detected by means of monitoring, as well as new feature requests, trigger a new iteration of the loop: Changes will be applied where needed, which may, in the worst case, be the Requirements Analysis step. In this case, a chain reaction of changes at all lower levels is usually unavoidable.

Of course, this picture is a simplification of what happens in reality, where the process is often much less linear. It should also be obvious that this traditional approach to system engineering potentially facilitates a comprehensive specification of the requirements and the verification of the system correctness, but is very inflexible with respect to changing requirements specifications: accounting for a new feature typically triggers another iteration of the complete system design cycle.

We emphasize that the system assumptions specified by the system designer in the Requirements Analysis step must be incorporated in the system in a way that can be satisfied during system operation, and we claim that this process is actualized in the formulation of an a priori belief set by the system designer. Interestingly, this happens not only in the early stages of the System Building Phase, but continues to play an important role during the whole lifetime of the system:[5] The life-cycle of a distributed system usually consists of a sequence of iterations of the outer

[5]This section focuses exclusively on "traditional" distributed systems that cannot autonomously change their

loop in Figure 2.1. In fact, each such iteration actually consists of two alternating phases, the Development Phase and Operating Phase. The Development Phase refers to the first four steps (Requirements Analysis to Testing/Verification) and is conducted by the system designer; the Operating Phasestarts with the fifth step (Deployment) and is autonomously performed by the agents. In the rest of this section, we will focus on the Development Phase and on the role of the system designer's a priori beliefs.

### 2.1.2 The role of a priori beliefs during development phase

During the Development Phase, the system designer needs to specify the system's assumptions that will hold in all states of the system, delimiting the set of states that are reachable during execution. This limited set of possibilities must also be provided to agents: in their reasoning, agents will consider only those states of the system that are provided by the system designer. Consequently, agents' epistemic attitudes (if any) are intended to range only over those states provided by the system designer. For this reason, we can call them "agents' a priori beliefs", or, to distinguish them from the corresponding notion for the system designer, "Operational APrB set":

**Definition 45** (Operational APrB Set (Agents))**.** *The set of states of the system that are reachable for a given agent. It is provided by the system designer during the development process either via their protocols or by some configuration data.*

Especially in traditional systems, the System Designer APrBs subsume all the a priori beliefs the to-be-developed system could possibly rely upon, in a format that is accessible to the system designer only, while the agents' Operational APrB sets can be viewed as a translation (and reduction) of the System Designer APrB into a format that is accessible to the agents. Consequently, the system designer has access to agents' Operational APrB sets: even if her knowledge of it might not be complete (e.g., due to some errors), she can observe the system behavior during monitoring. Thus, agents' Operational APrB sets constitute part of the a posteriori knowledge of the system designer, as it is used by the system designer to check whether the system functions accordingly. Clearly, though, it is only the Operational APrB sets that will govern the actual runs of the distributed systems after deployment. We will consider the role of the Operational APrB sets in Section 2.1.3, where we consider in more detail the Operating Phase.

The following example demonstrates that the a priori beliefs indeed restrict the states considered possible both by the system designer (System Designer APrBs) and, as a consequence, by the agents (Operational APrB sets).

**Example 46** (Safety property specification)**.** *In a simple traffic light system, a safety property could be informally formulated by the system designer as "the light is never both green and red at the same time." This a priori belief will then be formalized by the system designer into the agents' Operational APrB sets. Thus, such a system, while correct, will never reach a state where the variables for green light and red light are both true, meaning that such a state is not present*

underlying a priori assumptions during runtime. Indeed, in more complex systems such as SASO systems, also agents are able to play partially the role of the system designer.

*in the set of allowed runs. By enforcing this property, the system considered by the agents is restricted, a priori ruling out certain system states. Note that the situation of having the light both green and red at the same time is nevertheless globally possible: for example, such a violation of the specification can happen in the presence of failures. However, this situation will never be considered possible by the agents in any run because a state violating this safety property is not encompassed by the agents' Operational APrB sets. This is the reason why fault-tolerant systems amplify the importance of a priori beliefs.*

We stress here that a safety property like the one in Example 46 is indeed an element of the System Designer APrBs, which is then transposed into the corresponding formal property in the agents' Operational APrB sets: it does not depend on any execution, nor is it something that the agents could learn during execution; rather, it restricts the set of possible states the agents consider. One might argue that these a priori propositions are believed by agents within the system but are known to the external system designer. This, however, does not hold since the system designer can be mistaken in the formulation and/or the implementation of such a property, qualifying it as belief rather than knowledge.

Since the System Designer APrBs determines which variables are relevant and what the agent protocols must do in order to meet the specification, they are compatible with what Tahko refers to as "the a priori principles" [Tah11, p.162], grounding the very first a priori basis since any correct reasoning of the agents in the system must adhere to it.[6]

Figure 2.2 graphically represents the role of a priori beliefs during the Development Phase: namely, the various forms of belief/knowledge (rectangles) and some of the steps (edges with arrows) involved in its generation. In the figure, the arrows on the left indicate the dynamics of the generation of a priori beliefs throughout the Development Phase. From bottom to top, the figure shows the System Designer APrBs, consisting of the system designer's a priori beliefs about the system (constraints, safety and liveness specifications, and other properties the system needs to account for) as provided by the Requirements Analysis step. The System Designer APrBs are present in the designer's mind while designing the system and are thus implemented into agents' protocols, becoming the agents' Operational APrB sets. As such, the Operational APrB sets are somewhat in between the Development Phase and Operating Phase: they are transposed by the system designer onto the agents in the former, and used by the agents when executing their protocols in the latter. However, whereas the agents do produce (and update) their local a posteriori knowledge in the Operating Phase, as this forms the basis of their actions, they can neither autonomously produce nor update their Operational APrB sets.

The arrow going from agents' a posteriori knowledge to the System Designer APrBs stands for the dynamic process, by the system designer, of modifying some a priori belief. This is typically triggered when the system behaves unexpectedly during the Operating Phase, e.g., because of a severe mismatch between the specification and the monitored behavior or if a new feature has to be implemented. Depending on the source of the problem, she will apply changes by re-entering

[6]Tahko seems to have in mind basic logical principles like the law of non-contradiction, which may provide a foundational a priori basis. We do not need to take a stand on this broader issue here, but the structure of the foundational a priori principles is reflected by the System Designer APrBs.

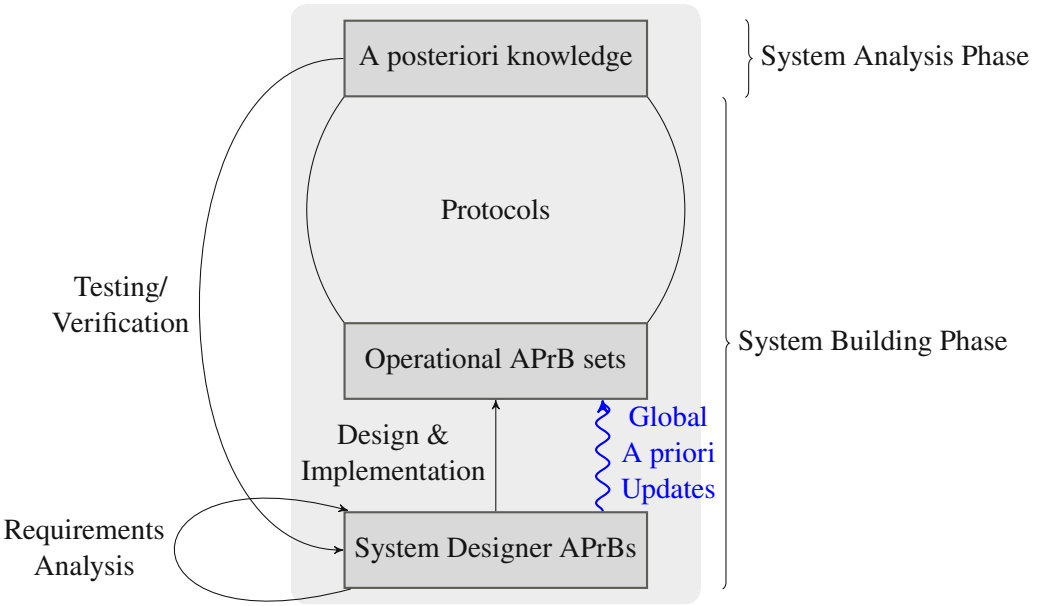



**Figure 2.2:** Illustration of the Development Phase of a traditional distributed system. Rectangles depict the various forms of knowledge/belief; arrows represent the steps in the design process generating them.

one of the steps in the design process. In case of an implementation error, the system designer might solve the problem by keeping the original System Designer APrBs and modifying only some minor system parameter, without re-configuring substantial portions of the system; in some cases, she may also need to adapt the Operational APrB sets. Design errors, on the other hand, require special handling, as fixing specification errors typically additionally requires changes of the System Designer APrBs, necessitating corresponding modifications of the whole system. If the system specification is to be changed, it inevitably creates the need for an *a priori belief update*, which plays a crucial role in this thesis:

**Definition 47** (A priori Belief Update)**.** *An a priori belief update, or simply* a priori update*, is an update conducted by a system designer (respectively an agent) aiming to update some a priori beliefs of some agent(s).*

The definition is extremely general, as the specific features and properties of a priori updates change dramatically depending both on the agent performing them and on the target agent(s). This difference is given by the extreme asymmetry between the role of the system designer w.r.t. the agents in distributed contexts. In Chapter 5, Definition 122 we formally define a priori belief updates using the tools of dynamic epistemic logic. In this section, we focus on the a priori updates conducted by the system designer, which can be of two kinds:

- *Self-updates*: the system designer, upon discovering a mismatch between the system require-

ments and the system executions, decides to update its a priori beliefs about the system (e.g. coming up with different system features). This is an a priori update of the system designer's a priori beliefs, represented in Figure 2.2 by the loop in the Requirements Analysis.

- *Global a priori Updates*: the system designer updates the a priori beliefs of the agents in the system. This operation does *not* update the system designer's a priori beliefs, but only the a priori beliefs of agents. Consequently, it is an a priori update only from the viewpoint of the agents (which usually are unable to detect such updates). It is represented in Figure 2.2 by the blue arrow[7].

Crucially, a self a priori update from the system designer will very likely trigger a global a priori update, as the changes in the assumptions of the system designer will likely be reflected in the system. At the same time, self a priori updates happen constantly, especially in the early phase of the design process. Unlike global a priori updates, they don't need to be formalized in order to have an effect on the resulting system. In particular, neither the update of the System Designer APrBs nor that of the Operational APrB sets can cause inconsistency problems here: for the former, because we assume a single[8] designer; for the latter, because all agents start from scratch in the newly built system.

Evidently, both the Operational APrB sets and System Designer APrBs are purely a priori from the system's viewpoint[9] and constitute the basis for the bootstrapping relation executed within the a posteriori framework formed by the system. At the Development Phase, the term "a posteriori basis"[10] subsumes all the information that the system designer has of the system, including the explicit portion of her System Designer APrBs, the Operational APrB sets of agents, and the system itself. In the presence of recalcitrant experience, the a posteriori basis will be revised and will undergo an a posteriori verification step. Consequently, the set of states considered possible by the system designer is fallible and dynamic as it is constantly reformulated based on the feedback given by the system (or a simulation thereof). In addition, the system designer also has some "background a priori beliefs" that range from propositions relevant to distributed systems to other fields, such as mathematics, philosophy, and so on. For example, some basic logical rules, such as the principle of non-contradiction, can be considered part of this tacit background, as suggested by Tahko [Tah08, Tah11].

It is clear that the exposed a priori belief dynamics involve a bootstrapping relation, in the spirit of Tahko: the system designer, based on newly acquired evidence (that can be gained both during the Development Phase or Operating Phase), discovers that her a priori beliefs cannot guarantee the desired properties of the system; to achieve her goal, she needs to come up with new a priori

[7] Global a priori updates are the topic of Chapter 4, where they will be treated more formally.

[8] The situation is different if one considers design teams since it often happens that not all designers share a common System Designer APrB set due to inconsistent updates. Modeling this situation is currently outside of our scope.

[9] By contrast, from the System Designer's point of view, the System Designer APrBs typically represent previously acquired (a posteriori) beliefs rather than a priori beliefs.

[10] Note that at the initial stage there is typically no implemented prototype system for experimental verification/testing. Nevertheless, some a posteriori verification is already possible by simulations or even model checking, using a coarse initial system model in order to detect specification violations.

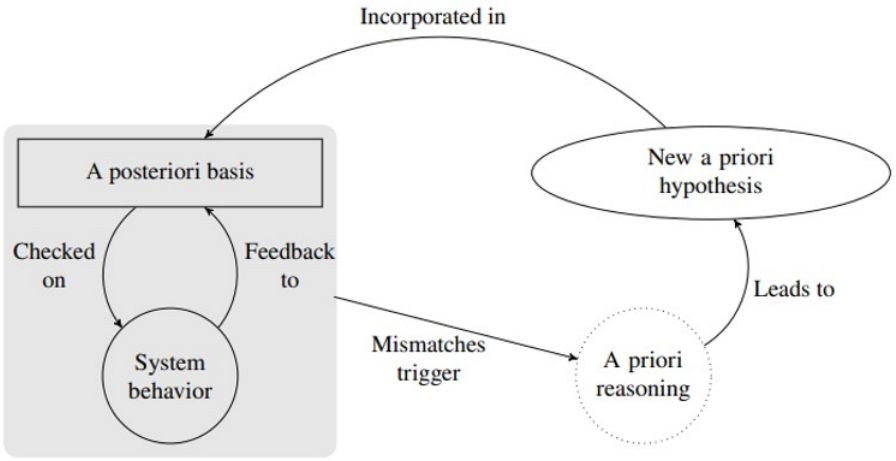



**Figure 2.3:** The bootstrapping relation in one cycle of the design procedure of distributed systems.

beliefs and a corresponding a priori update. However, in distributed systems, the situation is more complex as shown in Figure 2.3.

Here the a posteriori basis, which includes the a posteriori knowledge and some a priori assumptions, is checked on the actual system behavior.[11] The Testing/Verification (respectively System Maintenance) step acts as a checker of the a posteriori base, providing feedback to the system designer regarding whether the requirements and assumptions incorporated in the System Designer APrBs and Operational APrB sets are satisfied. If there is a mismatch between the intended behavior of the system and the effective behavior, then the system designer needs to revise its a priori beliefs, in a process that we label *a priori reasoning*, which consists in the process of formulating and/or modifying a priori beliefs[12]. These new a priori hypotheses are then integrated into the a posteriori basis, and the cycle can begin anew in case of further mismatch.

We conclude this section by comparing the role of a priori beliefs during the Development Phase in the present analysis with the use of a priori knowledge in the philosophical account proposed by Tahko [Tah08, Tah11]. Firstly, the objects and objectives in our context are very different from the classic setting addressed by Tahko, where the primary goal is to check which a priori propositions hold in the only "system" at hand, i.e., in the actual world. In particular, the knowledge of the scientific community about the world is the a posteriori basis against which the actuality of a priori propositions is tested. In man-made distributed systems, however, the situation is rather the opposite: the system designer wants some safety and liveness specifications to hold, under assumptions matching environmental constraints, and thus has to come up with the right system implementation that guarantees those. Hence, the system designer implements (or simulates the implementation of) agents' protocols according to environmental constraints and has some expectations about her system's overall behavior. A priori reasoning by the system designer

[11] Recall that in the original setting proposed by Tahko, the a posteriori basis is checked against experimental evidence. Once again, in distributed systems is the system behavior that constitutes the required experimental evidence.

[12] We will return to this topic in Section 2.3.

is triggered when her expectations are not matched during (simulated or real) testing. Such mismatches can happen for two main reasons that require separate handling:

- Since the protocols largely depend on the Operational APrB sets, it might be the case that the mismatch is due to an incompatibility between the desired properties and the current Operational APrB sets. To fix this, a revision of the Operational APrB sets is needed: the agents must be equipped with different a priori beliefs in order to achieve the target specification, necessarily influencing the underlying protocols.

- It may be the case that the target specifications cannot be achieved by the current System Designer APrBs: she must come up with different System Designer APrBs, which usually translates into different Operational APrB sets for the agents in the system, which, in turn, will not leave the agents' protocols and the system parameters untouched. This case is similar to the case of scientific inquiry, with one main difference: While the actual world cannot be changed in scientific inquiry process, in the design of distributed systems not only is it possible to change the agents' protocols and the system features, but any change to the a priori belief assumptions (being it to the Operational APrB sets or to the System Designer APrBs) will lead to a new set of possible worlds.[13]

To exemplify the difference between Tahko [Tah08, Tah11] and our view, consider the (very general) a priori proposition "there is no state of the system where $Y$ holds." Taking this proposition in Tahko's context of scientific inquiry, it could be translated to "in the actual world $Y$ does not hold." The goal of scientific inquiry is to test this proposition, that is, to (tentatively) verify the actuality (i.e., truth) of this proposition by means of critical experiments that aim at determining whether $Y$ holds. If it turns out that $Y$ holds, we cannot just change the world to make $Y$ false. The only alternative is to find another a priori proposition that would describe the targeted feature of the actual world more accurately. On the other hand, in the design of a distributed system, in the same situation, the system designer has the option to modify the agents' protocols (and other system parameters), before changing the a priori beliefs about the system; those will be modified only upon discovery that no protocol, together with the current system parameters and a priori beliefs, can satisfy the target requirements. Crucially, though, the modification of the a posteriori basis at any level has consequences at the underlying levels, resulting in a different system. Consequently, in the present analysis, executing bootstrapping steps does not result in a tree-shaped time evolution of the system as in Figure 1.8, in particular, if we ignore the possibility of maintaining multiple versions. After all, we allow the environment and even the desired safety and liveness specifications to just change, which implies that the System Designer APrBs and Operational APrB sets meant to accurately capture those will not just "pile up" in order to match reality better and better, but just change arbitrarily, and so will the implemented system.

[13] It is clear that the a priori reasoning step in the bootstrapping relation involves a revision process of the system designer's beliefs about the system. While the connection with existing belief revision approaches, like AGM [AGM85], is immediate, we do not investigate such implementation issues here.

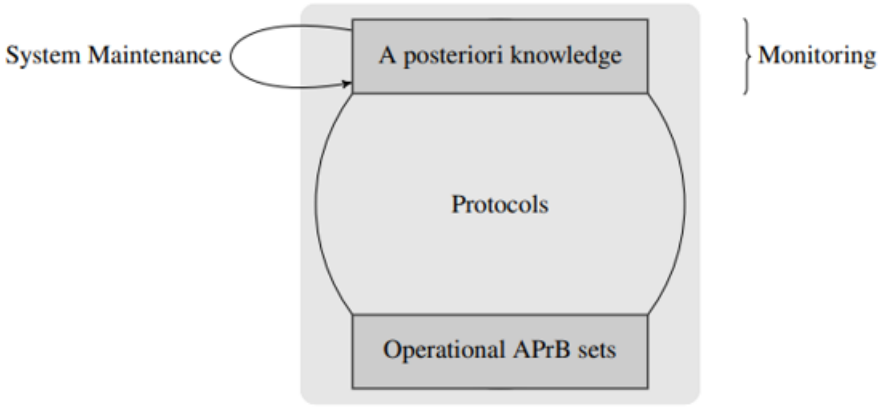



**Figure 2.4:** Illustration of the Operating Phaseof a traditional distributed system. Rectangles depict the various forms of knowledge/belief; arrows represent the steps in the design process modifying them.

### 2.1.3 The role of a priori beliefs in the operating phase

In the previous section, we have described the role of the system designer's a priori beliefs and we have explored its connection to the a priori beliefs of agents. In this section, we describe the role of a priori beliefs in traditional systems during the Operating Phase, where agents execute autonomously while being monitored by the system designer. Thus, the focus is on the Operational APrB sets agents are provided with. We explain how agents use these sets and how this notion relates to the epistemic notion of possible worlds typical of the epistemic analysis of distributed systems.

Figure 2.4 illustrates the Operating Phase of a traditional distributed system. Compared to the same figure for the Development Phase Figure 2.2, this one is only a portion of it. Recall that the agents' Operational APrB sets are provided by the system designer and are made accessible to the agents (via the protocol implementation or via some configuration data) during the Development Phase, which means that the Operational APrB sets remain unchanged throughout the Operating Phase, unless a new cycle of the design process is conducted by the system designer.

The a priori reasoning and the update of the Operational APrB sets can be performed solely by the system designer during the (intermediate) Design Phase, since agents in traditional distributed systems (unlike SASO systems) lack any introspection on the a priori assumptions governing the executions: when faced with unexpected behavior, they will reach wrong conclusions without hypothesizing about the set of system states that they are provided with.

The above considerations invite further comparisons w.r.t. the view developed by Tahko [Tah08, Tah11], wherein the a posteriori framework provides a basis for testing the actuality of a priori propositions and can be performed by the agent that can also perform a priori reasoning. This view quite nicely fits the situation in the Development Phase, where the system designer indeed uses the a posteriori basis (in the Testing/Verification step) to test the appropriateness of the assumptions in the System Designer APrBs. Moreover, it is the system designer who, in case

testing uncovers problems, corrects the design or implementation errors or, if needed, performs a priori reasoning in order to figure out new a priori beliefs and applies the corresponding a priori belief update to change inappropriate assumptions in the System Designer APrBs.

In the Operating Phase, however, the situation is very different. The a posteriori knowledge of the agents consisting of their local view of the system is inadequate for testing their respective Operational APrB sets. The best one can hope for here is that an agent detects that there must be something wrong with its Operational APrB set; in a fault-tolerant system, however, it would just (potentially mistakenly) blame itself as having failed instead.[14] Indeed, agents cannot update their a priori beliefs in the Operating Phase: an update of the Operational APrB set can only be enforced by external means, i.e., by the system designer, in a new cycle of the design loop as described in the previous section. This is indeed a substantial difference with respect to the account exposed by Tahko [Tah08, Tah11]: in the process of scientific inquiry, a priori reasoning may be initiated by the same agent who performs a posteriori reasoning. In traditional man-made distributed systems, the agents do not have such power: they lack the self-reflection and self-awareness tools, as well as a priori reasoning capabilities, to perform Operational APrB updates autonomously.

### 2.1.4 Epistemic logic and a priori beliefs

In this section, we establish a connection between the concepts exposed in this chapter and epistemic logic, by showing the role of the a priori beliefs in determining the set of possible states typical of the epistemic analysis of distributed systems. First, we establish the connection between protocols and agents' knowledge: a (knowledge-based) protocol can be translated in a Kripke model using interpreted systems [FHMV95], wherein the set of reachable states of a knowledge-based protocol can be encoded as a set of states of a Kripke model. Equivalently, in standard Kripke modeling, each Kripke model can be seen as a snapshot of a sequence representing the evolution of the system. Since the set of runs compatible with a system are generated according to a protocol (Section 1.2.3), the resulting Kripke model is also generated by the protocols. If the protocols are commonly known, then also the resulting set of possible states (together with the accessibility relations) is, making the resulting Kripke model also commonly known.[15]

In the previous sections, we endorsed the point of view of the system designer that needs to come up with system assumptions prior to the testing of the system. These assumptions are embedded into the system, into agent's protocols and into configuration data, determining the set of possible system behaviors. In particular, one of the distinguishing properties of the agents' a priori beliefs is that they do not depend on any particular experience: they are not recorded in the agents' history, they cannot be inferred from their local states, and they are often not even explicit in their protocols. This means that agents during testing/Operating Phase in traditional systems are blind to the a priori beliefs governing the system execution. Still, agents' a priori beliefs are taken to hold in the runs considered possible by an agent. However, such a priori beliefs may eventually

[14] In fault-tolerant distributed systems, agents can also believe other agents to be faulty. However, in case of a mismatch in its Operational APrB set, an agent has no other option than to consider itself faulty because, in order to consider another agent faulty, there must be a possible state where that agent is faulty. Since there is a mismatch in the Operational APrB set, such a possible state might not exist.

[15] We provide a logic framework where we relax the common knowledge of the protocols in Chapter 3.

turn out to be wrong, in the sense that there may be runs of the real system that the agent does not consider possible and that violate the intended assumptions, e.g., runs that violate the assumed maximum message delay. Whether the agent is able to detect such a violation or not, its protocols are likely to be inadequate for reaching their goals in such a run.

In order to properly account for a priori beliefs in the epistemic modeling of distributed systems, we advocate that three sets of possible states need to be considered by each agent,[16] as shown in Figure 2.5:

(i) The *Operational APrB set*, which is provided by the system designer and defines the set of runs and, hence, the set of possible states considered possible by an agent. Since the set of states of a system can be represented by an (epistemic) Kripke model, the (initial) Operational APrB set of an agent corresponds to the initial model that it is considering.

(ii) The *operationally possible states*, which we define as a subset of the agents' Operational APrB set consisting of the states that are candidates for being the actual state in the current execution. Note that this set is not static: while initially the two sets (i) and (ii) coincide, protocol execution will result in a shrinking of the latter, since the ability of a distributed system to solve some task, like distributed agreement [LSP82], crucially relies on the agents' abilities to rule out certain states as not being the actual state. Fewer states means less uncertainty and thus a gain in knowledge: this process is usually referred to as "epistemic reasoning" in the field of epistemic logic. But since such inferences are made based on local (a posteriori) observations, we will use the term "a posteriori reasoning". Note that such excluded states, despite not considered valid candidates for being actual, are still a priori possible, since they were defined by the agents' Operational APrB set. Obviously, the a posteriori reasoning process can only happen within the "space" of possibilities defined by the Operational APrB set; this set of operationally possible states can hence never exceed the set defined by agents' Operational APrB set.

(iii) The (singleton) set of the actual state, which is (in idealized scenarios, i.e., if the Operational APrB set of that agent is correct) a proper subset of the previous two sets.

The following examples illustrate this interplay between a posteriori reasoning and the agent's Operational APrB set:

**Example 48** (Epistemic Modeling of a Simple ATM System)**.** *In the simple ATM example introduced earlier, the set of alternative scenarios could be represented as a Kripke model with eight possible states*[17] *that define the set of possibilities considered by the agents: (a) there is a withdrawal request (1) or not (0); (b) all the conditions for accepting the request are met (1) or not (0); and (c) the requested amount is available (1) or not (0). In their a posteriori*

[16] The picture in multi-agents settings is more complicated, as it requires a distinction between actually possible worlds and virtually possible worlds (Section 1.2.2), which might not match in case of incomparable Operational APrB sets.

[17] For simplicity, we will consider the initial snapshot at time $t$ of a single execution.

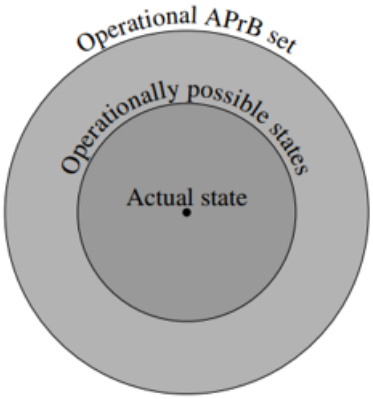



**Figure 2.5:** Simple representation of the subset relation between the Operational APrB set and the set of possible worlds involved in a posteriori reasoning. For simplicity, we focus on a single agent only.

*reasoning, agents can build their a posteriori knowledge solely based on this Kripke model and would not consider other scenarios (like providing a different amount of money than the one requested) possible at all. More specifically, the local a posteriori knowledge accumulated by some agent in the course of the protocol execution represents this agent's local view of the actual state, which is usually only partial. This a posteriori knowledge determines the protocol actions to be performed next. It depends on the current set of operationally possible states and possibly causes updates to it. Nonetheless, for a state to be considered operationally possible, it must naturally be considered possible first: hence, this subset of states is inherently constrained by the agent's Operational APrB set. For example, in a conventional ATM system, no agent would consider a state operationally possible where a customer can withdraw a negative amount of money because this is not a possibility foreseen by their respective Operational APrB sets.*

**Example 49** (Epistemic Modeling of Muddy Children Protocols)**.** *Recall the muddy children puzzle formulation from Section 1.2.4. In line with the standard interpretation, we consider the children to be the agents solving the puzzle, and the father has the role of an external announcer, but is not considered to be an agent trying to solve the puzzle. In distributed settings, father's announcements can be modeled as events that are commonly witnessed by all agents. The set of initial states encodes all the combinations of muddy/clean agents, forming the Operational APrB sets of the agents, which determines a corresponding set of runs. However, there are no protocols that would allow agents with these Operational APrB sets to solve the problem. By contrast, it can be solved if the possible state where nobody is muddy is dropped from the set of operationally possible states, which is usually achieved in the puzzle by an announcement originating from the father. From any of the remaining initial states, communication can then be used to reliably identify the initial state of the actual run, by gradually discarding (im)possible states from the agents' sets of operationally possible states. To complicate matters, suppose now that agent a is equipped with a wrong Operational APrB assumption and believes that another agent (say, b) is not one of the agents participating in solving the puzzle while it is. This means that agent a's*

*protocol will ignore agent b relative to the Muddy Children task. Hence, albeit a may receive messages relevant to the task that are actually sent by b, it might reach the wrong conclusion about its own status: this depends on the initial distribution, and thus the correct solution cannot be reached in each possible run*[18]*. What could happen, however, is that a detects that there must be something wrong regarding its Operational APrB set in some run. In any case, what would be needed here for a correct solution would be a way to update the Operational APrB set of agent a during a run, a feature that is available only by the intervention of the system designer*[19].

It is also interesting to view the situation discussed in Example 49 from the perspective of agent $b$ who does know about $a$'s incorrect Operational APrB set: its initial Kripke model in its Operational APrB set is richer than $a$'s model since it includes an additional agent. Besides the fact that the agents no longer have common knowledge of some Operational APrB set, it is important to note the key role of Operational APrB sets in communication: agent $a$ is aware of the presence of agent $b$ in general, but since it is not considered to be a player in the Muddy Children task, its messages are disregarded or misinterpreted.

More subtle instances of the task involve agents having a "hidden agenda": they could follow a protocol whose purpose it is to trick other agents into not being able to solve the Muddy Children Puzzle in some runs, by exploiting the (simple) epistemic reasoning that underlies the protocols of standard agents. Such considerations led us to the conviction that explicitly incorporating a priori beliefs in an epistemic reasoning framework is particularly fruitful for the analysis of byzantine fault-tolerant distributed systems and, crucially, for their ability to interpret and adequately react to such deviations.

Finally, recall that the system designer might not have complete knowledge of agents' Operational APrB sets: this is what usually happens in prohibitively large systems or in the early phases of design, where the system designer has not (yet) envisioned all possible states of the system.

## 2.2 A priori beliefs in SASO systems

In this section, we consider the role of a priori beliefs in SASO systems, and we bring forth the idea that the ability of SASO agents to adapt to emerging unexpected behaviour is given by an autonomous use of their a priori beliefs. More specifically, given their access to (at least some of) their own a priori beliefs, SASO agents are able to perform a priori beliefs updates in an autonomous way. This feature allows agents to react to system behaviors not considered possible during Development phase. To our knowledge, there is no connection between the literature about SASO systems and the epistemic analysis of distributed systems. One of the goal of this thesis is to establish this connection by interpreting some of the self-* abilities of SASO agents in terms of a priori reasoning capabilities. While the connection is only formulated informally in this section, a formalization of some of these abilities is offered in Chapter 5.

[18] Consider the case where $\mathcal{A} = \{a, b, c\}$ and $b$ is the only muddy agent. Since $a$ knows that there is at least one muddy agent after father's announcement and since $c$ is muddy and $a$ believes that $b$ is not participating, the only reasonable conclusion for $a$ is that $a$ is the muddy agent, which is incorrect.

[19] We will return to this example in a more formal way when introducing local a priori belief updates in Chapter 5.

First of all, it is important to recall that traditional systems can of course also designed to be adaptive. However, this requires that the "range" of adaption is already defined in the System Designer APrB during the Development Phase: epistemic reasoning can then be employed to perform the actual adaption during the runs in the Operating Phase, using a static and limited set of a priori assumptions. By contrast, here, we will consider systems that are capable of *changing* the Operational APrB of the agents during the Operating Phase, without an intermediate Development Phase.

Since SASO systems are designed to adapt to situations *not* initially envisioned by the system designer, they are characterized by the presence of agents that have some of the design capabilities that in traditional systems belong to the system designer only. Thus, to distinguish agents equipped with self-* abilities from the system designer and the agents of a traditional distributed system, we refer to them as Designer-agents:

- *Designer-agents* (DAs), are processes that possess self-* abilities allowing them to adapt their behavior outside of the initial protocol provided by the system designer, in reaction to unexpected system behavior.

As already mentioned in Section 1.1.2, agents in SASO systems are characterized by self-reflection and self-monitoring abilities, as well as by a priori reasoning capabilities, which allows them to do part of the system designer's work concerning Operational APrB updates. The maintenance of Operational APrBin SASO systems, therefore, differs greatly from its counterpart in traditional systems, since it (in principle) does not require the monitoring of the system designer for its well-functioning. Furthermore, there are no principal restrictions on the number of DAs: while it could be sufficient to introduce a single DA overseeing the whole system, fault-tolerant systems would require a more distributed setting: because the DAs must be able to update all the agents' Operational APrB, a single DA would constitute a single point of failure for the entire system. In the extreme case, every regular agent could be equipped with DA functionality.

### 2.2.1 A priori beliefs in the design of SASO systems

Designing SASO systems primarily requires a stratification of the Operational APrB already in the Development Phase, so as to allow DAs monitorning abilities, as illustrated in Figure 2.6.

As for the previous figures, rectangles represents any kind of knowledge and arrows represent the usage thereof. Compared to the traditional system depicted in Figure 2.2, in SASO systems the Operational APrB provided by the system designer consists of two distinct parts:

(1) The *DA APrB*, which contains all the self-* tools that allows DAs to monitor the system, doing a priori reasoning and performing a priori updates.[20] While this set is initially provided by the system designer, at least part of it can be autonomously modified by agents during the Operating Phase, as we will see in the next section.

[20]We explore the role of a priori reasoning in Section 2.3.

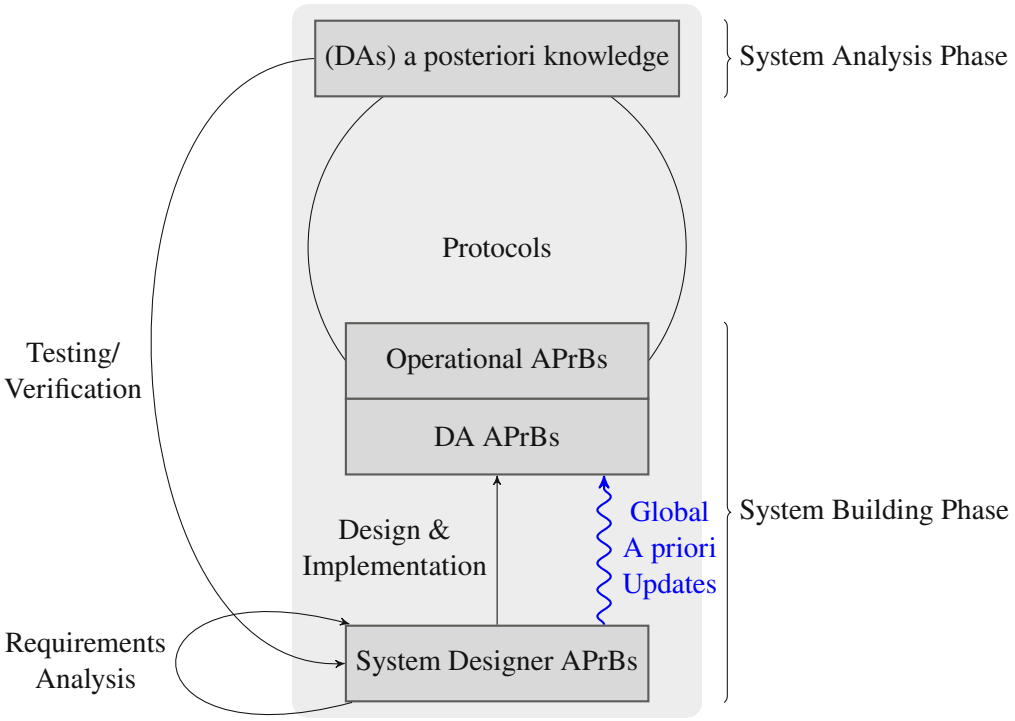



**Figure 2.6:** Illustration of the Development Phase of a SASO distributed system. Rectangles depict the various forms of knowledge, and edges with arrows the steps in the design process generating/using those.

(2) The Operational APrB of the regular agents, which, like in traditional systems, is initially a portion of the System Designer APrB but that can be later be modified not only by the system designer, but also autonomously by the DAs in the Operating Phase, e.g. by having DAs communicating their new a priori beliefs to standard agents.

Importantly, DAs usually execute very different protocols, which need to be designed and implemented correctly in the Development Phasein order to account for self-* abilities. For this reason, we also need to separate the epistemic knowledge maintained by the regular agents from the epistemic knowledge maintained by the DAs. Note that, as in traditional systems, the system designer can also intervene in SASO systems via global a priori updates and changing the underlying a priori belief of DAs and standard agents, as shown in Figure 2.6. While initially the system designer's a priori beliefs determines the a priori beliefs of agents, DAs might deviate from it via *Local a priori belief updates*, as we will see in the next section.

### 2.2.2 A priori beliefs in the operating phase of SASO systems

Whereas the Development Phase of SASO systems is not substantially different from its traditional counterpart (Figure 2.2), this is not the case for the Operating Phase, which is illustrated in

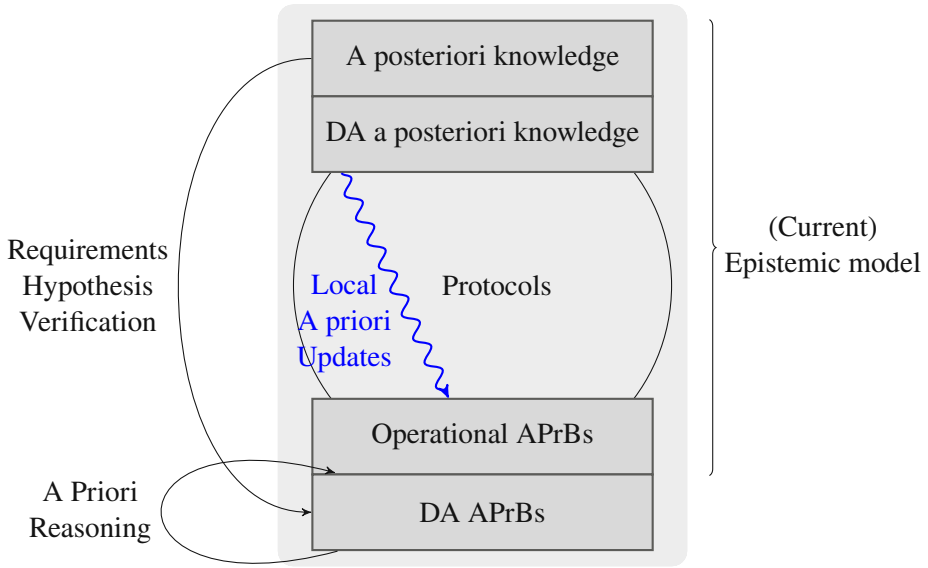



**Figure 2.7:** Illustration of the Operating Phase of a SASO distributed system. Rectangles depict the various forms of knowledge, and edges with arrows the steps in the design process modifying those.

Figure 2.7. While the regular agents usually just do what they do in traditional systems, namely, a posteriori reasoning for building up their epistemic knowledge based on the Operational APrB during the execution of their protocols, it may now happen that the DAs occasionally perform a priori updates, to be applied to some or all agents. This happens when the DAs, who monitor the behavior of the regular agents, possibly using tools for self-reflection, detect some violation of the current Operational APrB (in the *Requirements Hypothesis Verification* step in Figure 2.7). To distinguish them from their counterpart performed by the system designer, we call these updates *Local a priori updates*, as they are autonomously formulated and implemented by the DAs during Operating Phase.

The a priori reasoning process performed by DAs, which of course also involves the a posteriori DA epistemic knowledge built-up by the DAs in the run so far, results in a new Operational APrB' for the target agents, which is finally installed by a suitable a priori belief update. Importantly, all these activities of the DAs happen in the course of the execution of their DA protocols in the Operating Phase, and do not involve any intervention by the system designer.

Of course, the design of SASO systems is considerably more complex than the design of traditional systems: the DA APrB must be general enough to allow the DAs to come up with new variables and/or to modify the boundaries of the existing ones etc. Unfortunately, as of today, this is done purely *ad hoc*, since a solid conceptual (not to speak of formal) foundation and related tool support is lacking. Evidently, DAs needs to be able to adapt the current model to a new one, enriched with new (a priori) possibilities. This corresponds to the self-* abilities of such agents that allow them to learn from the current execution and to adjust the a priori hypothesis to the situation at hand,

| Agents | Can do A priori Updates? | Who can be affected? | When is performed? |
|---|---|---|---|
| System Designer | Yes | All | Development Phase |
| Traditional Agents | No | / | / |
| DAs | Yes | DAs | Operating Phase |

**Table 2.1:** Table illustrating the the different features of a priori updates depending on the kind of agent in question.

in ways not initially envisioned by the system designer. In addition, DAs need to be equipped with dedicated memory and monitoring abilities, in order to observe and store the results of this a priori belief dynamics.

Table 2.1 illustrates the a priori update abilities of agents considered so far in a compact way. Accordingly, agents in traditional systems are not capable of autonomous a priori reasoning and cannot perform a priori updates. The system designer has such abilities, and uses them during the Development Phase, either initially or after discovering some error during Operating Phase; the updates performed by the system designer can (in principle) affect all agents. DAs in SASO systems also posses these abilities; their updates are performed during Operating Phase and can affect only themselves as a result of a priori updates.[21]

We illustrate (DA) agents' reasoning using a variant of the Muddy Children Puzzle:[22]

**Example 50** (Muddy Children in SASO systems)**.** *Consider once again the problem of solving the Muddy Children puzzle of Example 49, but this time in a SASO system. For this scenario, assume that the system designer restricted the initial model so as to include always at least two muddy children. This design choice is reflected in the Operational APrB of the agents involved, in particular, also of the DAs. Suppose however that during one interation of the Muddy Children, there is only one muddy DA. In this instance, the muddy DA agent would immediately detect an inconsistency, simply by observing two clean agents, contradicting the assumption employed by the system designer. Hence, the DA must come up with a new a priori belief, perhaps less restrictive in the number of agents, and update its current a priori beliefs, so as to match the observations previously made.*

The a priori belief dynamics informally exposed in the example can be summed up in the following three steps:

- Inconsistency detection: An agent detects an inconsistency or an unexpected behavior in the system, which is typically traced back to some (now falsified) assumption(s) in the current Operational APrB. In epistemic terms, it means that the agent detecting it becomes *inconsistent*, formally $K_a\bot$, for some agent $a \in \mathcal{A}$.

[21] The fact that a priori updates are only self-updates does not preclude agents to communicate (or enforce) the newly acquired a priori beliefs to other agents. However, this operation corresponds to communicating new a priori beliefs which is an *a posteriori* update.

[22] In Chapter 5, we will show how these updates can be formalized, using the Muddy Children as a core example.

- A priori reasoning: DA agents come up with new suitable a priori belief hypotheses that could solve the inconsistency. This corresponds to the update synthesis task (Section 1.2.6).
- A priori belief updates: DA agents, after selecting new suitable a priori belief hypotheses, update their a priori beliefs (and potentially also the ones of other agents). If the result of the update does not solve the inconsistency, or a new one arises, then the process starts anew.

Notice that an inconsistent agent knows falsum $K_a\bot$, and since $\mathcal{M} \not\models \bot$, it means the accessible part of the Kripke model for an inconsistent agent is empty. Consequently, an a priori update should provide the agent with another model that does not lead to the unexpected behavior. There are no restrictions on how the DA a priori reasoning is performed, e.g. via statistical methods, using some predefined inference rules, using machine learning, etc. Consistently with our take on the role of the DA APrB, however, we stress that the real distinction between SASO systems and traditional adaptive systems is in terms of autonomous a priori belief dynamics. The ability of the DAs to correctly react to situations not initially envisioned by the system designer hence relies crucially on having provided sufficiently rich reasoning tools at the Development Phase.

As already mentioned in the previous section, the ability to dynamically perform Operational APrB updates in a SASO system is particularly interesting for fault-tolerant distributed systems. Since the Operational APrB in traditional systems is considered to be unquestionable, an agent faced with unexpected behaviour can only conclude that it is itself faulty. As stressed in [BG09], SASO systems represent a natural answer to some of the problems raised by byzantine fault-tolerant distributed systems: Their self-* abilities to dynamically react to unexpected changes reduces the hostility of byzantine settings, which are characterized by an elevated number of possible unexpected behaviours. Another advantage is that, instead of collecting all the possible misbehaviours of faulty agents in a gargantuan epistemic model, the initial Operational APrB just assumes that all agents were correct, while maintaining all the possible alternatives present (but not computed) in the meta-model. The "everyone is correct" assumption underlying the initial Operational APrB can be later modified by the DAs in the presence of some unexpected behavior, by initiating a priori reasoning and forging an alternative model that explains the unexpected behavior.

We finally note that the ability of a SASO system to adapt its specifications and goals during the Operating Phase also relates to the bootstrapping cycle: since the current a priori belief set has proved to be inadequate by the current state of the system, it is modified and/or expanded so that it fits the current system execution. Somewhat surprisingly, the bootstrapping relation in SASO systems is hence more close to the one proposed by Tahko, where a priori reasoning also amounts to establishing new a priori hypotheses that are better suited to explain the phenomenons observed during a posteriori verification, with some important differences:

- It is difficult, sometimes impossible, for the DAs in a SASO system to simultaneously update the Operational APrB at all agents. Moreover, like in Example 50, an Operational APrB update will only be performed for a proper subset of the regular agents. One way to

overcome this shortcoming is to allow DAs to communicate new a priori beliefs to other (non-updated) agents.

- Because of their autonomous a priori reasoning capabilities, SASO systems usually lack the convenient assumption of common knowledge of the Operational APrB by all agents: Even if it would be initially the case, the first private a priori update would destroy such assumption (even worse, other agents might believe that the model is still commonly known even if it is no longer the case).

## 2.3 A priori reasoning

The process of a priori reasoning is a crucial component of the bootstrapping loop introduced by Tahko: in this step, new a priori hypotheses are formulated, providing alternative explanations to the phenomena at hand. A priori reasoning is of course a complex process, involving a vast spectrum of skills, from problem-solving analysis to analogy, guessing and creative thinking. Clearly, at least on a first examination, a priori reasoning is not a deterministic process: there appear not to be any deterministic algorithm that, given as input the problems of physics of the early 20th Century, outputs the relativity theory. In particular, relativity theory is not the only physical theory that provides an explanation to the same physical phenomena, and a unifying theory is still lacking. Furthermore, as all scientific inquiries, relativity theory is subject to falsification, meaning that there might be a better theory that explains the target phenomena, albeit it is not yet known. At the same time, a priori reasoning is not made by purely random guesses, as the theories formulated through this process (should) take into account the problems left open by former theories. In other words, this process is guided by the information accumulated, both through experiments and by the information provided by former falsified a priori assumptions. A characterization of the a priori reasoning process is outside of the scope of this thesis: a first important obstacle is that identifying the exact set of assumptions that needs to be adjusted during the process is not easy, as stated by the *Duhem-Quine thesis*, for which it is often impossible to determine precisely which hypotheses are falsified by the falsifying experiment.

History of science has a number of cases where the assumptions that were put under discussion were also at the core of the scientific enterprise. One such example can be found in some interpretations of quantum physics, were the revision process of the a priori assumptions that should hold was not conservative at all, as they suggested that, under special circumstances, the principle of non-contradiction can be dropped, namely because two opposite quantum states can be true at the same time[23].

All the problems mentioned above in the field of scientific inquiry are also present in the design of distributed systems.[24] Specifically, we will identify the main components of this process and use them for differentiating between the kind of systems that we considered in this section,

[23]This issue is closely related to the issue of finding an a priori basis that cannot be revised under any circumstance. In fact, an early plausible candidate for this basis is the principle of non-contradiction [Tah11, p.162]. However, the quantum states example clearly shows that, at least for some well-defined instances, even this principle can be dropped.

[24]Probably the restricted field of distributed systems tones down the magnitute that this reasoning process can achive, as dropping the principle of non-contradiction in standard distributed systems sounds rather drastic.

namely traditional, adaptive systems and SASO systems. Recall that, in Section 2.1.4, we defined the Operational APrB of an agent to be the set of states of a system that the agent considers possible. In the epistemic analysis of distributed systems, this means that we can represent the Operational APrB of an agent as the set of states that are accessible to that agent[25]. Epistemic (a posteriori) reasoning of agents is based on the model they consider, and on its evolution resulting from (epistemic) actions. The a priori beliefs of the system designer and of DAs, on the other hand, is situated at the meta-level of reasoning, and is not directly represented in the Kripke model. Recall also that the system designer might not have complete knowledge of agent's Operational APrB this is what usually happens in the early phases of design, where she has yet not envisioned all possible states of the system she is designing. This is particularly true for SASO systems, where the system designer usually does not know all reachable states of the system, possibly not even initially.

Our focus will be on three main elements, the *Meta-model*, the *Background a priori beliefs* and the *A priori reasoning capabilities*. We start with informally defining these three elements:

**Definition 51** (Meta-model)**.** *The meta-model is a collection of possible models, each with its set of reachable states. Each model is characterized by different a priori assumptions. Hence, the meta-model collects the possible alternatives to the current model, obtained by modifying the underlying a priori assumptions. All these models do not need to be explicitly computed: rather, the meta-model provides the space where the current model can be modified and extended in (some of) its possible extensions during execution. This space can also be used to store information about previous models*[26].

The difference between a model and a meta-model can be expressed via the *actual versus potential* distinction: a model affects the actual epistemic knowledge of agents, while the meta-model affects it only potentially. This potentiality is then actualized when one of the models of the meta-model is chosen as the actual one.

**Definition 52** (Background a priori knowledge)**.** *The background a priori knowledge is the set of all a priori propositions that are not objects of the a priori reasoning, i.e., that would not be questioned in case of revision. The elements of this set depend on the agent in question. For a human system designer, it can include a priori assumptions about some parameters external to the system that still influences it e.g. a priori assumptions about mathematics.*

**Definition 53** (A priori reasoning capabilities)**.** *The a priori reasoning capabilities are the tools used for reasoning at the meta-level. They heavily depend on the agent considered, and they may include induction, hypothesis formulation, analogy, learning, guessing and many others.*

[25] Formally, we don't require the accessible worlds to be restricted to the worlds accessible to that agent only. For example, if the CKM assumption is in play, than the agent has access to all the worlds that are accessible also to other agents [Art20].

[26] The notion of meta-model can be thought as the modal equivalent of the notion of hypothesis space in learning theory [Jai09, KK07]: it represents the set of all possible hypotheses (or models in our case) that an agent might consider. As for the meta-model, the hypothesis space is determined by the protocols/algorithm's design, including the chosen model type (e.g., decision trees or neural networks) and other constraints and/or biases.

*Among these capabilities, it is required to meta-level agents to be able not only to conceive new possibilities, but also to update and to integrate the information learned from bootstrapping with the a posteriori knowledge obtained during execution. All these abilities are needed for maintaining an effective and useful meta-model.*

We will now study the shape and the role of these elements in different classes of distributed systems.

### Traditional systems

Figure 2.8 shows the interplay between the a priori reasoning elements in traditional systems, where the a priori belief dynamics is performed solely by the system designer during the design of a distributed system. There, the elements described above need not to be clearly and formally specified: the system designer may have only fuzzy and informal concepts in mind, and the wanted result is reached by trial and error. Hence, the meta-model is highly dynamic: The system designer constantly reformulates it through the interplay with the other meta-level components, based on the feedback given by the system (or a simulation thereof).

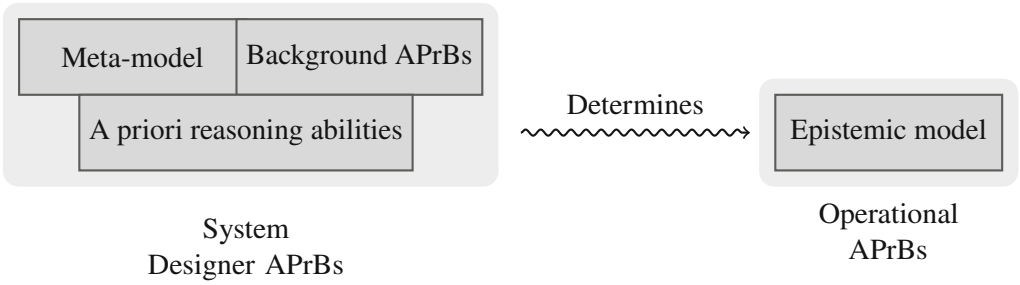



**Figure 2.8:** Interplay of a priori reasoning elements in traditional systems. The System Designer APrB is only used by the system designer and as such the meta-model can be though as being into the system designer's mind. The meta-model is not static, because of its interplay with the other components of System Designer APrB .

### Traditional adaptive systems

Traditional adaptive systems add an additional element to traditional systems, namely a *static meta-model*, which is provided by the system designer to the agents in the system, as showed in Figure 2.9. Usually the static meta-model is only a portion of the meta-model of the system designer, which is formally elaborated and implemented into the system. However, the meta-model it is static only from the viewpoint of the agents in the system, which are not able to modify it. The system designer could still intervene later on to change it, like in traditional systems.

The meta-level capabilities of the adaptive agents, when triggered, are restricted to choosing one set of reachable states (model) among the possibilities provided in advance by the system designer, which are to be found in the static meta-model.

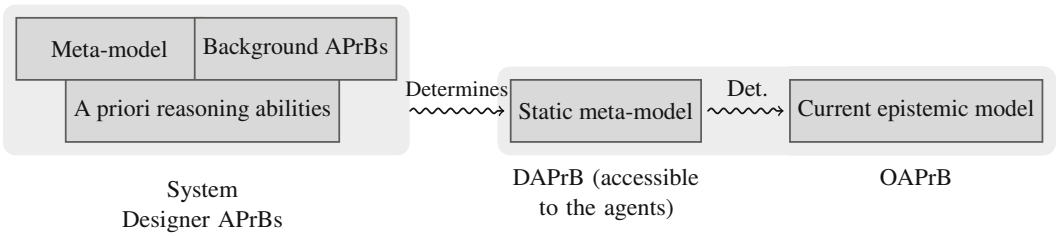



**Figure 2.9:** Interplay of a priori reasoning elements in adaptive systems. Part of the system designer's meta-model is provided to the agents in the system, via a static meta-model, which can be modified only by the system designer itself.

### SASO Systems

In SASO systems, the situation is more complex, since DAs are equipped with a priori reasoning tools of their own. As showed in Figure 2.10, the System Designer APrB determines the DA a priori beliefs, which must be implemented initially in full details by the system designer. This meta-level of a priori reasoning is what determines the key self-* abilities of DAs. The crucial element is that, despite being characterized by the same elements, System Designer APrB and DA APrB are not identical. Clearly, the system designer might have initially a richer set of a priori reasoning abilities w.r.t. DAs, which may be equipped only with a portion of it; the same can be said for their Background APrB. However, because of the self-* abilities, the DAs can reach significantly different DA APrB w.r.t. the System Designer APrB.

DAs are also be equipped with tools to modify the meta-model, setting them apart from TAS: agents in the former have the reasoning power to modify the meta-model, and come up with new models to make sense of the unexpected system behavior. Agents in TAS, on the other hand, lack this a priori reasoning abilities and are henceforth bounded to the meta-model provided by the system designer.

In fact, the meta-level a priori beliefs and capabilities of the system designer and of DAs are not essentially different, as the main elements involved are the same. This can also be extended to Tahko's example about scientific inquiry: what scientists do when performing a priori reasoning may not be substantially different from what the system designer or DAs do, despite the different subject of inquiry[27].

## 2.4 Conclusions and related work

In this chapter, we addressed the importance of the role of a priori beliefs in the Development Phase and Operating Phase of distributed systems, emphasizing the different types of a priori states and the different agents concerned. In the Development Phase, the System Designer APrBs, which comprise the system designer's (still not formalized) view of the system, together with its expected behavior and the assumptions about the environment, are established and used to

[27]This observations opens up a discussion about the role of creativity in coming up with new possible explanations for phenomena, and weather this process can be replicated by a machine.

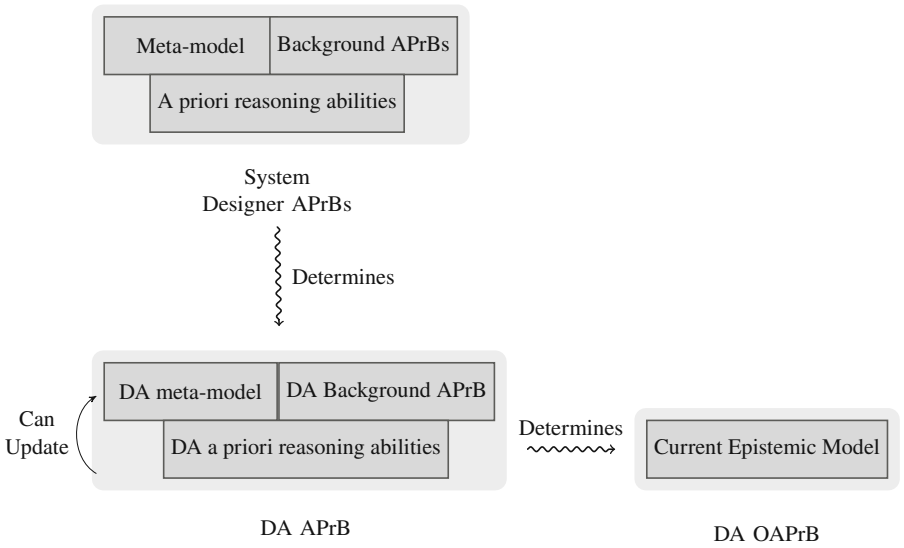



**Figure 2.10:** Interplay of a priori reasoning elements in SASO systems. The meta-model can be dynamically changed by DAs, as represented by the loop on the left.

determine the Operational APrB sets. The latter define the set of possible states of the system (based on the set of relevant variables) and, thus, forms the "arena" in which the a posteriori reasoning of the agents takes place during the Operating Phase. We have also explored the similarities and differences w.r.t. Tahko's account. We share the view that the a priori is a modal notion, albeit concerned with possibility rather than necessity, that it is fallible, i.e., that there is no guarantee of correctness of a priori beliefs, and that bootstrapping relations faithfully tie a priori beliefs and the a posteriori reasoning of the agents together. The main differences are the presence of different types of agents: namely, system designer and processes, the ability of the former to affect ("construct") the actual world, the sometimes different use of the a posteriori reasoning framework, and the fact that the agents do not necessarily have common knowledge of the a priori beliefs.

In terms of related work, Tahko's account of a priori knowledge bears clear similarities to the concept of abduction [Dou25]. In its standard formulation, abduction is an ampliative form of inference (i.e. its conclusions extend beyond what is logically entailed by the premises) and it is guided by explanatory considerations. This resemblance suggests a potential connection between the process of revising or expanding a priori assumptions and the inferential mechanisms underlying abductive reasoning. In other words, given some evidence and some candidate explanations, abduction allows one to determine which of the explanations is the best for the evidence under consideration. While there are clearly some similarities with the a priori account proposed by Tahko, such as the parallel between evidence and the a posteriori basis, or the analogy between explanations and a priori assumptions, there are also important differences: First, the a priori reasoning process involves more than simply selecting the best explanation for a given body of evidence. It also requires identifying which explanations are even possible or accessible based on that evidence. In this sense, abduction constitutes only one phase of a broader

bootstrapping cycle Second, in the field of distributed systems, it is not only the hypotheses that can be revised, but also the evidence itself (namely, the system under consideration) if it fails to meet the expectations of the system designer. This process can be seen as a form of (higher-order) abduction, where both the explanatory assumptions and the observational basis are subject to revision in order to achieve the desired system behavior. A deeper investigation into the precise role of abduction within the a priori update process is left for future work.

A closely related area is belief revision [AGM85], where the set of beliefs of an agent is modified based on some information received that must be integrated in the agent's set of beliefs in a consistent way. The belief revision framework presents some similarities with the a priori belief account introduced here. For instance, a priori beliefs can be seen as more deeply entrenched than their a posteriori counterparts: falsifying an a priori belief typically requires a more substantial revision than falsifying an a posteriori one. Conceptually, an a priori update could be described in belief revision terms as a mix of *expansion*, *revision*, *consolidation* and *replacement*: A new piece of information $p$ is added to the (a priori) belief base (expansion), which may render the set inconsistent. To restore consistency, certain elements may be discarded (consolidation), while others are introduced or restructured (replacement), ultimately yielding a revised and coherent (a priori) belief base (revision).

There are, however, significant differences between a priori updates and traditional belief revision. Notably, a priori updates can involve changes not only to beliefs but also to the very rules of reasoning. Moreover, a priori beliefs are often not explicitly represented in the belief base; rather, it is the role of the a priori reasoning process to determine which implicit assumptions must be questioned and which new a priori beliefs should be incorporated to resolve inconsistencies in a principled way. Finally, this thesis adopts a semantic approach to modeling a priori belief updates, in contrast to the predominantly syntactic focus of most belief revision frameworks. We will also examine existing semantic approaches to belief revision in logic, particularly plausibility models [BS06, BS16], and compare them to our framework in Chapter 4 and Chapter 5.

In the following chapters we integrate and apply the philosophical groundwork introduced in this chapter to the epistemic analysis of distributed systems.

CHAPTER 3

# Relaxing the common knowledge of the protocols: the creed modality

In this chapter[1], we start relaxing some of the most common a priori assumptions in fault-tolerant distributed systems. As we have seen in Section 1.2.3, the *hope* modality $H_i\varphi$ is needed for reasoning about systems with arbitrary faults. This modality is usually employed in systems where correct agents share common knowledge of their respective protocols and treat faulty agents as deviating from those. Since the protocol determines the set of possible states reachable by an agent, in the epistemic analysis of distributed systems, having a commonly known protocols is equivalent to having a commonly known model, wherein agents reason about the system execution. Naturally, some additional assumptions must be made to restrict the types and numbers of protocol-defying actions and messages. Without such restrictions, provably correct solutions for a distributed task do not exist. These assumptions must usually be commonly known by all agents as well, in conjunction with the basic communication mechanism, the protocol of all correct agents, the data encoding used in its messages, etc., constituting the corpus of the agents common a priori knowledge.

Since correct agents generally cannot distinguish a simple malfunction from a malintent, erroneous messages, i.e., messages sent in contravention of the commonly known joint protocol, are usually left uninterpreted. For instance, in the epistemic modeling and analysis framework [FKS21, KPSF19b, KPSF19a, KPS+19] for byzantine agents, message $\varphi$ received from agent $i$ is interpreted by means of the hope modality $H_i\varphi := correct_i \rightarrow B_i\varphi$, which is equivalent to a disjunction $\neg correct_i \vee (correct_i \wedge B_i\varphi)$, suggesting that the reception of a message $\varphi$ from $i$ at some other agent must be interpreted as involving the uncertainty between $i$ being faulty and the epistemic state of $i$ confirming $\varphi$ in case $i$ is a correct agent. Note that in the former case, the message carries no meaning whatsoever. Indeed, the axiomatization of hope in [vDFK22] takes $H_i\bot$

[1]The first part of this chapter is based on joint work with Hugo Rincon Galeana, Roman Kuznets and Ulrich Schmid, [CKRS23].

to be the definition of faulty agents because only a faulty agent can send contradictory messages. However, not all systems exhibit such a stark dichotomy between the commonly known and fully transparent *us* (correct processes) and the mysterious and uninterpretable *them* (faulty processes). *Rational agents* in blockchain settings [GKTZ12], for instance, do not necessarily share the same goals as the rest of the system. Nevertheless, neither their actions nor their communication are arbitrary, not to speak of adversarial. Consequently, game theoretic modeling, based on a model of their beliefs and goals, can be applied for the analysis of such systems [AGBPBTP20].

In this chapter, we extend this finer-grained view to the epistemic modeling of distributed systems and consider heterogeneous distributed systems (Section 1.1.1), where different processes may run different protocols and where the assumption that all protocols are commonly known is dropped. We assume that processes are partitioned into types (or roles, or classes) of agents, so that within one type the protocols are commonly known to the agents of that type. While such a strong assumption is not made for agents of different types, we do not assume them to have zero knowledge of each other's protocol either. In particular, we assume that each class is equipped with an *interpretation function* that encodes the amount of knowledge agents have regarding the preconditions for communication agents of a different type have. Since having no preconditions for sending a message is an allowed instance, this also covers the byzantine setting described earlier, where there are two types — correct and faulty agents — and only messages of correct agents have a non-trivial interpretation. These interpretation functions are formalized by means of the new family of modalities, which we will introduce in the following sections.

Agent-types in the field of epistemic logic were already formulated in [BCR21], where *names* are used as abstract roles for groups of agents, depending on their characteristics. From the dynamic epistemic logic [vDvdHK07] perspective, a public announcement logic with agent types was presented in [LW13], providing a dynamic framework to reason about uncertainty of agent-types that is used to formalize the knights and knaves puzzle. Due to the different motivations, while treating a closely related problems set, [BCR21] and [LW13] make different and at times incomparable choices regarding the postulates underlying the systems. For instance, a precondition for an announcement for an agent in [LW13] need not entail the agent knowing this precondition, which contradicts the fundamental Knowledge of Preconditions principle (Theorem 12) for distributed systems [Mos15]. On the other hand, all agents in [LW13] possess the same knowledge about each of the existing agent types, in particular, all agents share one common interpretation of messages from a particular type, an assumption in line with the rather centralized nature of updates in dynamic epistemic logic but less sensible for distributed systems. After introducing our framework, we explain in Section 3.4 how the above examples could be formalized using our new modality.

## 3.1 Epistemic logic for heterogeneous distributed systems

We introduce is the *creed* modality $\mathbb{C}_p^{A\setminus B}\varphi$, which generalizes the hope modality for the byzantine case and represents the information an agent of type $A$ can infer upon receiving message $\varphi$ from agent $p$ of type $B$. We consider a heterogeneous distributed system $\langle \mathcal{A}, \Pi, \mathcal{P}, \mathcal{C}, T\rangle$ according to Definition 1, where processes are partitioned into different types according to their protocol.

Agents of the same type share a common protocol, which also includes information on how to interpret messages from agents of various types. Recall that we assume that each process knows its own protocol/type, and, therefore, the protocol of all other agents of the same type, but not necessarily which agents are of this type. In particular, an agent may be unsure whether another agent belongs to its own type or not.

Agents interpret received messages by means of an interpretation function:

**Definition 54** (Interpretation function)**.** *Let $\mathbb{B}$ be the set of well-defined formulas used by agents to communicate. An* interpretation function *for type $A \in \Pi$ with respect to type $E \in \mathcal{A}$ messages is any function $f_{AE} : \mathbb{B} \to \mathbb{B}$.*

Intuitively, $f_{AE}(\varphi)$ corresponds to the knowledge that type $A$ agents (or simply $A$ agents) have about the preconditions for $E$ agents to send message $\varphi$. We assume that function $f_{AE}$, for every type $E$, is a priori known by every $A$ agent, as part of its protocol.

**Example 55.** *Interpretation function $f_{AE}(\varphi) := \top$ for all $\varphi \in \mathbb{B}$ corresponds to the case when $A$ agents have no knowledge about the communication protocol of $E$ agents. For instance, byzantine agents who can send any message at any time (send-unrestricted byzantine agents) can be captured by choosing $f_{\mathsf{Correct},\mathsf{Faulty}}(\varphi) = \top$ for partition $\Pi^E = \{\mathsf{Correct}, \mathsf{Faulty}\}$. The minimal requirement that all correct agents tell the truth translates into $f_{\mathsf{Correct},\mathsf{Correct}}(\varphi) = \varphi$.*

Since we want to be able to express partition membership in our language and formulas, we need to define partition membership atoms.

**Definition 56** (Propositional variables and partition atoms)**.** *We consider, for each process $p_i \in \mathcal{A}$, a finite set $Prop_i$ of* propositional variables*. In addition, for each agent type $A \in \Pi$, we consider the set $\Pi_A := \{A_p \mid p \in \mathcal{A}\}$ of* partition atoms*, expressing the fact that agent $p$ belongs to a certain partition $A$. The set of all* atomic propositions *is defined as the union of the sets of propositional variables and of partition atoms: $Prop := \bigcup_{i=1}^{n} Prop_i \cup \bigcup_{A \in \Pi} \mathcal{A}_A$.*

Since $\Pi$ is a partition, every agent belongs to one and only one type. For convenience, we denote the type of agent $p$ by $\bar{p}$. Furthermore, we will assume that each agent knows its own type, i.e., $K_p(\bar{p}_p)$.

Now that we have established the basics, we can proceed to define the language.

**Definition 57** (Language of EHL)**.** *The language $\mathcal{L}^{\mathsf{EHL}}$ of the epistemic heterogeneous logic extends the standard (multi-modal) epistemic language $\mathcal{L}$ by a new family of modalities called* creed *and is given by the grammar:*

$$\varphi ::= r \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid K_p\varphi \mid \mathbb{C}_p^{A\setminus E}\varphi, \tag{3.1}$$

*where $r \in Prop$ is an atomic proposition (i.e., propositional variable or partition atom), $p \in \mathcal{A}$ is an agent, and $A, E \in \Pi$ are agent types. Other boolean connectives, as well as boolean constants*

*$\top$ and $\bot$, are defined in the usual way. We use the following derived modalities: $\hat{K}_p\varphi := \neg K_p \neg\varphi$ and creed defined as*

$$\mathbb{C}_p^{A\backslash E}\varphi := E_p \to K_p f_{AE}(\varphi) \tag{3.2}$$

*for any agent $p \in \Pi$, agent types $A, E \in \Pi$ and $f_{AE} : \mathbb{B} \to \mathbb{B}$ with $\mathbb{B} \subseteq \mathcal{L}^{\mathsf{EHL}}$.*

Creed $\mathbb{C}_p^{A\backslash E}\varphi$ represents the amount of information an $A$ agent can extract from a message $\varphi$ received from agent $p$ under the assumption that $p$ belongs to type $E$ of the partition. It is based on the a priori knowledge $A$ agents possess of the preconditions for an $E$ agent to send message $\varphi$, as encoded in the interpretation function $f_{AE}$ from Definition 54, which is external to the language. This precondition already takes into account the Knowledge of Preconditions principle (Theorem 12), by assuming that the sender must know that the preconditions hold. We use the standard Kripke model semantics with additional restrictions for partition atoms:

**Definition 58** (Semantics)**.** *Let $\langle \mathcal{A}, \Pi, \mathcal{P}, C, T\rangle$ be a heterogeneous distributed system and $\{f_{AE} \mid A, E \in \Pi\}$ be the collection of interpretation functions for it. An* (epistemic) Kripke frame *$F = (W, \sim)$ is a pair of a non-empty set $W$ of worlds (or states) and a function $\sim : \mathcal{A} \to \mathcal{P}(W \times W)$ that assigns to each agent $p \in \Pi$ an equivalence relation $\sim_p \subseteq W \times W$ on $W$. A* Kripke model *$M = (W, \sim, V)$ is a triple where $(W, \sim)$ is an epistemic Kripke frame and $V : W \to \mathcal{P}(Prop)$ is a* valuation function *for atomic propositions. The truth relation $\vDash$ between Kripke models and formulas is defined as follows: $M, s \vDash r$ iff $r \in V(s)$ for any $r \in Prop$; cases for the boolean connectives are standard; $M, s \vDash K_p\varphi$ iff $M, t \vDash \varphi$ for all $t \in W$ such that $s \sim_p t$.*

*As usual, validity in a model, denoted $M \vDash \varphi$, means $M, s \vDash \varphi$ for all $s \in W$.*

*A Kripke model $M = (W, \sim, V)$ is called an* EHL model *iff the following two conditions hold:*

1. *For any state $s \in W$ and any agent $p \in \mathcal{A}$,*

$$\Big|V(s) \cap \{A_p \mid A \in \Pi\}\Big| = 1, \tag{3.3}$$

   *i.e., exactly one of partition atoms $A_p$ involving agent $p$ is true at state $s$.*

2. *For any agent $p$, any agent type $A$, and pair of states $s$ and $t$,*

$$s \sim_p t \quad \Longrightarrow \quad \Big(A_p \in V(s) \quad \Leftrightarrow \quad A_p \in V(t)\Big), \tag{3.4}$$

   *i.e., $p$ can distinguish worlds where it is of different type.*

*General validity, denoted $\vDash \varphi$, means $M \vDash \varphi$ for all* EHL *models.*

**Example 59.** *For the interpretation functions from Example 55 for send-unrestricted byzantine agents, $\mathbb{C}_p^{\mathsf{Correct}\backslash\mathsf{Faulty}}\varphi = \mathsf{Faulty}_p \to K_p\top$. For epistemic models, it is logically equivalent to $\top$, meaning that no information can be gleaned from a message under the assumption that it is sent by a fully byzantine agent. At the same time, for truth-telling correct agents $\mathbb{C}_p^{\mathsf{Correct}\backslash\mathsf{Correct}}\varphi = \mathsf{Correct}_p \to K_p\varphi$, which closely matches the hope modality $H_p\varphi = \mathsf{Correct}_p \to K_p(\mathsf{Correct}_p \to \varphi)$ from [vDFK22]. Indeed, since we assume agents to know their own type, $\mathsf{Correct}_p \to K_p\mathsf{Correct}_p$ holds, making $H_p\varphi$ equivalent to $\mathbb{C}_p^{\mathsf{Correct}\backslash\mathsf{Correct}}\varphi$.*

**Example 60.** *Apart from helping to understand messages, an interpretation function can be used to gain knowledge about the type of the sender. For instance, if $A$ agents know enough about the way $E$ agents communicate to conclude that a particular message $\varphi$ can never be sent by an $E$ agent, which corresponds to $f_{AE}(\varphi) = \bot$, then $\mathbb{C}_q^{A\setminus E}\varphi = E_q \to K_q\bot$. For epistemic models, such $\mathbb{C}_q^{A\setminus E}\varphi$ is logically equivalent to $\neg E_q$. In other words, having received $\varphi$ from agent $q$, an $A$ agent $p$ learns at least $K_p\neg E_q$.*

**Remark 61** (Information from message passing)**.** *Let $p, q \in \mathcal{A}$ be agents and $\Pi$ be a partition of $\mathcal{A}$. The knowledge gained by agent $p$ upon receiving a message $\varphi$ from agent $q$ can be described by $K_p\mathbb{C}_q^p\varphi$, where*

$$\mathbb{C}_q^p\varphi := \bigwedge_{E\in\Pi} \mathbb{C}_q^{\bar{p}\setminus E}\varphi \tag{3.5}$$

*In other words, knowing its own type, $p$ considers all possible types for the sender $q$ and for each type considers the respective interpretation of the message; the conjunction combined with the implications within creed make sure that the appropriate type is chosen. Note that the presence of send-unrestricted agents from Example 59 adds a conjunct to Equation* (3.5) *that is equivalent to* $\top$*. Hence, send-unrestricted agents can be safely ignored in determining the message meaning. By the same token, some conjuncts in Equation* (3.5) *can rule out a particular type for agent $q$ as in Example 60. Finally, if $p$ has already ruled out some type $E$, then $K_p\neg E_q$ logically implies $K_p(E_q \to K_q f_{AE}(\varphi))$ independent of the interpretation function. In this case, the $E$-conjunct of Equation* (3.5) *becomes redundant.*

**Example 62.** *In the system from Example 55 with send-unrestricted byzantine agents, upon receiving message $\varphi$ from agent $q$, agent $p$ can ignore the possibility of the sender being* Faulty *and conclude* $\mathsf{Correct}_q \to K_q\varphi$*, i.e., hope $H_q\varphi$ for the case of factive beliefs, in full accordance with [vDFK22]. Note also that $p$ may infer $K_q\varphi$ from this message if $p$ is sure that $q$ is correct.*

Now that we have established the basic definitions and semantics for the logic, we will provide an axiomatization that we prove sound and complete in the next section.

**Definition 63** (Logic EHL)**.** *Let $\langle \mathcal{A}, \Pi, \mathcal{P}, C, T\rangle$ be a heterogeneous distributed system. Logic* EHL *is obtained by adding to the standard axiomatization of modal logic of knowledge* S5 *the partition axioms* P1–P3*. The resulting axiom system is as follows: for all $p \in \mathcal{A}$, all $A \in \Pi$, and all $E \in \Pi$ such that $E \neq A$,*

Taut *All propositional tautologies in the language of* EHL*;*

k $K_p(\varphi \to \psi) \to (K_p\varphi \to K_p\psi)$*;*

4 $K_p\varphi \to K_pK_p\varphi$*;*

t $K_p\varphi \to \varphi$*;*

5 $\neg K_p\varphi \to K_p\neg K_p\varphi$*;*

(MP) *rule inferring $\psi$ from $\varphi \to \psi$ and $\varphi$;* (Nec) *rule inferring $K_p\varphi$ from $\varphi$;*

$$\text{P1} \bigvee_{A\in\Pi} A_p; \qquad \text{P2}\ A_p \to \neg E_p; \qquad \text{P3}\ A_p \to K_pA_p. \tag{3.6}$$

Partition axiom P1 states that each agent belongs to at least one of the types. Partition axiom P2 postulates that each agent belongs to at most one of the types. Together they imply that agent types partition the set of agent. Partition axiom P3 expresses that every process knows its own type.

## 3.2 Soundness and completeness of EHL

Since EHL is an extension of S5 with partition axioms governing the behavior of partition atoms while EHL models are instances of epistemic models, the soundness and completeness for EHL follows the standard proof for S5 (see, e.g., [vDvdHK07]), where additionally it is necessary to establish that the partition axioms are sound and that the canonical model satisfies the additional restrictions.

**Theorem 64** (Soundness)**.** *Logic* EHL *is sound with respect to* EHL *models, i.e.,* $\mathsf{EHL} \vdash \varphi$ *implies* $\vDash \varphi$.

*Proof.* We only establish the validity of partition axioms. Axioms P1 and P2 hold due to condition Equation (3.3). Similarly, P3 holds because of Equation (3.4). □

Completeness is proved by the standard canonical model construction, which requires several definitions. We omit the proofs of the following lemmas if completely standard and only treat the new cases covered by $\mathsf{P1} - \mathsf{P3}$ otherwise.

**Definition 65** (Maximal consistent sets)**.** *A set* $\Gamma \subseteq \mathbb{F}$ *of formulas is called* consistent *iff* $\mathsf{EHL} \nvdash \neg \bigwedge \Gamma_0$ *for any finite subset* $\Gamma_0 \subseteq \Gamma$. *A set* $\Gamma$ *is called* maximal consistent *iff* $\Gamma$ *is consistent but no proper superset* $\Delta \supsetneq \Gamma$ *is consistent.*

**Lemma 66** (Lindenbaum Lemma)**.** *Any consistent set* $\Gamma$ *can be extended to a maximal consistent set* $\Delta \supseteq \Gamma$.

**Definition 67** (Canonical model)**.** *The canonical model* $M^C = (S^C, \sim^C, V^C)$ *is defined as follows:*

- $S^C$ *is the collection of all maximal consistent sets;*
- $\Gamma \sim_p \Delta \quad$ *iff* $\quad \{K_p\varphi \mid K_p\varphi \in \Gamma\} = \{K_p\varphi \mid K_p\varphi \in \Delta\}$;
- $V^C(\Gamma) := \{r \in Prop \mid r \in \Gamma\}$.

**Lemma 68** (Truth Lemma)**.** *For any* $\varphi \in \mathbb{F}$ *and any* $\Gamma \in S^C$,

$$\varphi \in \Gamma \quad \Longleftrightarrow \quad M^C, \Gamma \vDash \varphi$$

**Lemma 69** (Correctness)**.** *The canonical model is an* EHL *model.*

*Proof.* That $S^C \neq \varnothing$ and $\sim_p$ is an equivalence relation for each $p \in \Pi$ is proved the same way as for S5. It remains to show that Equation (3.3) and Equation (3.4) hold.

(3.3) Consider any maximal consistent set $\Gamma \in S^C$ and any agent $p \in \Pi$. By the standard properties of maximal consistent sets, all theorems of EHL belong to each maximal consistent set, in particular, $\left(\bigvee_{A\in\Pi} A_p\right) \in \Gamma$ because of axiom P1. A disjunction belongs to a maximal consistent set iff one of the disjuncts does. Hence, there exists at least one type $A$ such that $A_p \in \Gamma$. At the same time, for any other type $E$, we have $(A_p \to \neg E_p) \in \Gamma$ because of axiom P2. Hence, $E_p \notin \Gamma$ because maximal consistent sets are consistent and closed with respect to (MP). It follows that there is exactly one partition atom of the form $A_p$ in $\Gamma$. Hence, by the definition of $V^C$,

$$\left|V^C(\Gamma) \cap \{A_p \mid A \in \Pi\}\right| = 1.$$

(3.4) Consider two maximal consistent sets $\Gamma \sim_p \Delta$. Let $A_p \in \Gamma$. By P3, also $K_pA_p \in \Gamma$. Hence, $K_pA_p \in \Delta$ by the definition of $\sim_p$. Finally, $A_p \in \Delta$ by axiom t. We proved that $A_p \in \Gamma$ implies $A_p \in \Delta$. The inverse implication is analogous.

□

**Theorem 70** (Completeness)**.** *Logic* EHL *is complete with respect to* EHL *models, i.e.,* $\mathsf{EHL} \vdash \varphi$ *whenever* $\vDash \varphi$.

*Proof.* We prove the contrapositive. Assume $\mathsf{EHL} \nvdash \varphi$. That means that $\{\neg\varphi\}$ is consistent. By the Lindenbaum Lemma 66, there exists a maximal consistent set $\Gamma \supseteq \{\neg\varphi\}$. Hence, $\Gamma \in S^C$ for the canonical model $M^C$ defined in Definition 67, which is an EHL model by Lemma 69. By the Truth Lemma 68, it follows that $M^C, \Gamma \vDash \neg\varphi$. Since $M^C, \Gamma \nvDash \varphi$ for some EHL model, $\varphi$ is not valid, i.e, $\nvDash \varphi$. □

## 3.3 Properties of creed

In this section, we derive several useful properties of the creed modality. The explicit assumption P3 that each agent knows which type it belongs to implies a complete knowledge of the own type, i.e., each agent $a$ knows whether it belongs to any type $A$:

**Theorem 71.** $\mathsf{EHL} \vdash \neg A_p \to K_p\neg A_p$ *for all* $p \in \mathcal{A}$, $A \in \Pi$, *i.e., agents know which type they do not belong to.*

*Proof.* By P1, agent $p$ must belong to one of the types in $\Pi$. Hence, if not type $A$, it must be one of the remaining types, i.e., $\neg A_p \to \bigvee_{E\neq A} E_p$. Therefore, we have $\neg A_p \to \bigvee_{E\neq A} K_pE_p$ due to P3. Given that $E_p \to \neg A_p$ for each $E \neq A$ by P2, also $K_pE_p \to K_p\neg A_p$ for each $E \neq A$ by standard modal reasoning. Hence, $\neg A_p \to K_p\neg A_p$. □

**Corollary 72.** $\mathsf{EHL} \vdash K_pA_p \vee K_p\neg A_p$ *for all* $p \in \mathcal{A}$, $A \in \Pi$.

*Proof.* Follows directly from P3 and Theorem 71 by propositional reasoning. □

The creed modality amounts to K45-belief:

**Theorem 73.** *Creed satisfies the normality, positive and negative introspection axioms if applied to statements already translated by an interpretation function. Formally, let* $[\![\varphi]\!]_{AE}$ *stand for any formula* $\xi$ *such that* $f_{AE}(\xi) = \varphi$*. Then the following formulas are derivable in* EHL*:*

$$\mathsf{k}_{\mathbb{C}} \vdash \mathbb{C}_p^{A\setminus E}[\![\varphi \to \psi]\!]_{AE} \to \left(\mathbb{C}_p^{A\setminus E}[\![\varphi]\!]_{AE} \to \mathbb{C}_p^{A\setminus E}[\![\psi]\!]_{AE}\right)$$

$$\mathsf{4}_{\mathbb{C}} \vdash \mathbb{C}_p^{A\setminus E}[\![\varphi]\!]_{AE} \to \mathbb{C}_p^{A\setminus E}\left[\!\left[\mathbb{C}_p^{A\setminus E}[\![\varphi]\!]_{AE}\right]\!\right]_{AE}$$

$$\mathsf{5}_{\mathbb{C}} \vdash \neg\mathbb{C}_p^{A\setminus E}[\![\varphi]\!]_{AE} \to \mathbb{C}_p^{A\setminus E}\left[\!\left[\neg\mathbb{C}_p^{A\setminus E}[\![\varphi]\!]_{AE}\right]\!\right]_{AE}$$

*Proof.* We start by deriving $\mathsf{k}_{\mathbb{C}}$:

1. $\mathbb{C}_p^{A\setminus E}[\![\varphi \to \psi]\!]_{AE} = E_p \to K_p(\varphi \to \psi)$ — definition of creed
2. $K_p(\varphi \to \psi) \to (K_p\varphi \to K_p\psi)$ — axiom k
3. $\mathbb{C}_p^{A\setminus E}[\![\varphi \to \psi]\!]_{AE} \to \big(E_p \to (K_p\varphi \to K_p\psi)\big)$ — prop. reasoning from 1.,2.
4. $\mathbb{C}_p^{A\setminus E}[\![\varphi]\!]_{AE} = E_p \to K_p\varphi$ — definition of creed
5. $\mathbb{C}_p^{A\setminus E}[\![\varphi \to \psi]\!]_{AE} \to \left(\mathbb{C}_p^{A\setminus E}[\![\varphi]\!]_{AE} \to (E_p \to K_p\psi)\right)$ — prop. reasoning from 3.,4.
6. $E_p \to K_p\psi = \mathbb{C}_p^{A\setminus E}[\![\psi]\!]_{AE}$ — definition of creed
7. $\mathbb{C}_p^{A\setminus E}[\![\varphi \to \psi]\!]_{AE} \to \left(\mathbb{C}_p^{A\setminus E}[\![\varphi]\!]_{AE} \to \mathbb{C}_p^{A\setminus E}[\![\psi]\!]_{AE}\right)$ — rewriting of 5. using 6.

The following is a derivation of $\mathsf{4}_{\mathbb{C}}$:

1. $\mathbb{C}_p^{A\setminus E}[\![\varphi]\!]_{AE} = E_p \to K_p\varphi$ — definition of creed
2. $K_p\varphi \to K_pK_p\varphi$ — axiom 4
3. $\mathbb{C}_p^{A\setminus E}[\![\varphi]\!]_{AE} \to (E_p \to K_pK_p\varphi)$ — prop. reasoning from 1.,2.
4. $K_p\varphi \to (E_p \to K_p\varphi)$ — prop. tautology
5. $K_pK_p\varphi \to K_p(E_p \to K_p\varphi)$ — normal modal reasoning from 4.
6. $\mathbb{C}_p^{A\setminus E}[\![\varphi]\!]_{AE} \to \left(E_p \to K_p\mathbb{C}_p^{A\setminus E}[\![\varphi]\!]_{AE}\right)$ — prop. reasoning from 3.,5. using 1.
7. $E_p \to K_p\mathbb{C}_p^{A\setminus E}[\![\varphi]\!]_{AE} = \mathbb{C}_p^{A\setminus E}\left[\!\left[\mathbb{C}_p^{A\setminus E}[\![\varphi]\!]_{AE}\right]\!\right]_{AE}$ — definition of creed

8. $\mathbb{C}_p^{A\setminus E}[\![\varphi]\!]_{AE} \to \mathbb{C}_p^{A\setminus E}\left[\!\left[\mathbb{C}_p^{A\setminus E}[\![\varphi]\!]_{AE}\right]\!\right]_{AE}$ rewriting of 6. using 7.

The following is a derivation of $5_{\mathbb{C}}$:

1. $\neg\mathbb{C}_p^{A\setminus E}[\![\varphi]\!]_{AE} \leftrightarrow E_p \wedge \neg K_p\varphi$ prop. reasoning from the definition of creed
2. $E_p \to K_pE_p$ axiom P3
3. $\neg K_p\varphi \to K_p\neg K_p\varphi$ axiom **5**
4. $\neg\mathbb{C}_p^{A\setminus E}[\![\varphi]\!]_{AE} \to K_p(E_p \wedge \neg K_p\varphi)$ normal modal reasoning from 1.–3.
5. $\neg\mathbb{C}_p^{A\setminus E}[\![\varphi]\!]_{AE} \to K_p\neg\mathbb{C}_p^{A\setminus E}[\![\varphi]\!]_{AE}$ normal modal reasoning from 1.,4.
6. $\neg\mathbb{C}_p^{A\setminus E}[\![\varphi]\!]_{AE} \to \left(E_p \to K_p\neg\mathbb{C}_p^{A\setminus E}[\![\varphi]\!]_{AE}\right)$ prop. reasoning from 5.
7. $\neg\mathbb{C}_p^{A\setminus E}[\![\varphi]\!]_{AE} \to \mathbb{C}_p^{A\setminus E}\left[\!\left[\neg\mathbb{C}_p^{A\setminus E}[\![\varphi]\!]_{AE}\right]\!\right]_{AE}$ rewriting of 6.

□

In addition, the "creed belief" is factive whenever the speaker type is correctly identified (cf. a similar conditional factivity for hope in [vDFK22]):

**Theorem 74.** $\mathsf{t}^*_{\mathbb{C}}$: $\mathsf{EHL} \vdash E_p \to \left(\mathbb{C}_p^{A\setminus E}[\![\varphi]\!]_{AE} \to \varphi\right)$.

*Proof.*
1. $\vdash \mathbb{C}_p^{A\setminus E}[\![\varphi]\!]_{AE} = (E_p \to K_p\varphi)$ definition of creed
2. $\vdash K_p\varphi \to \varphi$ axiom t
3. $\vdash E_p \to \left(\mathbb{C}_p^{A\setminus E}[\![\varphi]\!]_{AE} \to \varphi\right)$ prop. reasoning from 1.,2.

□

On the other hand, misidentifying the speaker's type may easily destroy factivity. Let $p \notin E$. Given that $\mathbb{C}_p^{A\setminus E}[\![\varphi]\!]_{AE} \to \varphi = (E_p \to K_p\varphi) \to \varphi$, we have that $E_p \to K_p\varphi$ is true simply because $E_p$ is false.

This provides a formal model of how a true statement can lead to false beliefs due to misinterpretation. Moreover, as Theorem 73 shows, such false beliefs cannot be detected solely by introspection.

## 3.4 Applications of creed

In this section we illustrate the usefulness of the creed modality by means of some examples.

**Example 75** ("The Murders in the Rue Morgue")**.** *This famous story by Edgar Allan Poe describes a murder mystery. While nobody saw the murderer (agent m), several witnesses heard m. The problem in interpreting their testimony is that they seem to contradict each other: for instance, a French witness f thinks m spoke Italian and is certain m was not French, whereas a Dutch witness d thought m was French, etc. Importantly, none of the witnesses could understand what was being said (f does not speak Italian, while d does not speak French, etc.). The standard byzantine framework considers the possibility of a faulty agent sending different messages to different agents to confuse them, but provides no means to describe one uncorrupted message being treated so differently by correct agents. Standard epistemic methods either accept all incoming information as being of equal value or make a priori preferential judgements. However, in the story, Monsieur C. Auguste Dupin correctly surmises that m spoke neither of the languages. Dupin neither dismisses witness accounts completely as lies nor accepts them completely. Instead he chooses some of the witness statements over others without prejudging them.*

Example 75 describes a situation where honest witnesses provide contradictory information that is, nevertheless, successfully filtered by Dupin. We show how his reasoning can be formalized and explained using the creed modality. Dupin reads all witness accounts from a paper. We assume no misinterpretation of what the witnesses said. In addition, the paper mentions the exact type of each witness (French not speaking Italian, Dutch not speaking French, etc.), which again is assumed to be factive, i.e., $f_{AE}(\varphi) = \top$ whenever $A \neq E$. Here $\varphi$ represents the sound that was heard by the witnesses, and we assume that they all heard the same sound, even if they have different interpretations for which language it was.[2] Hence, we use only one creed modality with the identity interpretation function per witness account read by Dupin. In other words, Dupin reasons about the available information without the need to interpret it. The crucial question is: Why does Dupin ignore some but not all of the information provided by each witness? The answer becomes clear if we view each witness account as one or several creed modalities regarding what this witness heard from $m$. Ignoring slight variations in details, all witness statements can be divided into two types: (a) $m$ did not speak the language I speak; (b) $m$ spoke a specific language I do not speak. Dupin accepts statements (a) but ignores statements (b). Even when statement (b) of a witness contradicts statement (a) of another witness, Dupin accepts statements (a) from both witnesses. This is how these statements of the French witness $f \in F$ regarding the utterance $\varphi$ of $m$ can be represented via the creed modality:

$$(a) \quad \mathbb{C}_m^{F\setminus F}\varphi = F_m \to K_m f_{FF}(\varphi) = F_m \to K_m\bot;$$
$$(b) \quad \mathbb{C}_m^{F\setminus I}\varphi = I_m \to K_m f_{FI}(\varphi) = I_m \to K_m\top.$$

Indeed, for (a), since the interpretation function from French to French is meaningful (in the simplest case, is the identity function), the fact that $f$ could not understand what $m$ was saying in

[2] In this example, $\varphi$ can be thought of the sequence of phonemes uttered by a speaker.

this case means that $f_{FF}(\varphi) = \bot$. On the other hand, for (b), since $f$ does not know Italian, he has $f_{FI}(\psi) = \top$ for all $\psi$. As discussed in Example 60, (a) yields $\neg F_m$. Similarly, (b) yields $\top$ as per Example 59. This rightfully leads Dupin to the conclusion $\neg F_m$, i.e., $m \notin F$. In other words, statements (b) are ignored because they are trivial, not because they are false. One might say that, for $f$, a stronger precondition for $m$ saying something in Italian is $m \in I$. But using $I_m \to K_m I_m$ in place of (b) would yield axiom P3, still a logically trivial statement.

In the story, $m$ was an orangutan, thus, fulfilling $m \notin A$ for any language $A$ discussed.

**Example 76** (Knights and Knaves puzzles)**.** *There is a series of logical puzzles, popularized by Smullyan [Smu78], about an island, all inhabitants of which are either* knights *who always tell the truth or* knaves *who always lie. One of the simplest ones ([Smu78, Puzzle 28]) is as follows:*

> *There are only two people, p and q, each of whom is either a knight or a knave. p makes the following statement: "At least one of us is a knave." What are p and q?*

Our goal is to incorporate the uncertainty about the mode of communication (knaves lie / knights tell the truth) into the logic. Fault-tolerant systems do not provide a satisfactory model since there information from faulty agents is either accepted (in case of benign faults) or ignored as completely unreliable (in case of byzantine faults). Instead, enough information is collected from correct agents (and they must constitute an overwhelming majority for most problems to be solvable). By contrast, knights and knaves puzzles are typically solvable even if all agents involved are knaves. The answer to the puzzle above, for instance, is that $p$ is a knight and $q$ is a knave. We would like to derive this answer fully within the logic.

Clearly the partition of the island from Example 76 involves two types: $I$ for knIghts and $A$ for knAves. Let $s$ be the reasoner and $L$ be his type. The puzzle postulates that $f_{LI}(\varphi) = \varphi$ and $f_{LA}(\varphi) = \neg\varphi$ for any formula $\varphi$. Accordingly, the full information agent $s$ receives from agent $p$'s statement that $\varphi$ is

$$\mathbb{C}_p^s\varphi = \mathbb{C}_p^{L\setminus I}\varphi \wedge \mathbb{C}_p^{L\setminus A}\varphi = (I_p \to K_p\varphi) \wedge (A_p \to K_p\neg\varphi).$$

In the puzzle in question, $p$ states that at least one of $p$ and $q$ is a knave, $A_p \vee A_q$ in formulas. Hence, agent $s$ learns

$$\mathbb{C}_p^s(A_p \vee A_q) = \Big(I_p \to K_p(A_p \vee A_q)\Big) \wedge \Big(A_p \to K_p\neg(A_p \vee A_q)\Big). \tag{3.7}$$

Here is how to derive in EHL that $p$ is a knight and $q$ is a knave, i.e., $I_p \wedge A_q$:

1. $A_p \to K_p\neg(A_p \vee A_q)$ — prop. reasoning from (3.7)
2. $K_p\neg(A_p \vee A_q) \to \neg A_p$ — t and prop. reasoning
3. $\neg A_p$ — prop. reasoning since $A_p \to \neg A_p$ follows from 1. and 2.
4. $\neg A_p \to I_p$ — P1 and prop. reasoning

5. $I_p$ (MP) from 3. and 4.
6. $I_p \to K_p(A_p \vee A_q)$ prop. reasoning from (3.7)
7. $I_p \to A_p \vee A_q$ t and prop. reasoning from 6.
8. $I_p \to A_q$ prop. reasoning from 7. since $I_p \to \neg A_p$ by P2
9. $I_p \wedge A_q$ prop. reasoning from 5. and 8.

Hence, $\mathsf{EHL} \vdash \mathbb{C}_p^s(A_p \vee A_q) \to I_p \wedge A_q$.

**Example 77** (Software Updates)**.** *In a highly available large scale distributed system like an ATM network, it is impossible to simultaneously update the software executed by the processes. Rather, processes are usually updated more or less sequentially during normal operation of the system, at unpredictable times. As a consequence, the joint protocol executed in the system while a software update is in progress might mix both old and new protocol instances. Existing solutions like [ALS06, SCF+16], which aim at updating complex protocols/software, typically provide "consistent update" environments that prevent such mixing.*

Thanks to our creed modality mixed joint protocols can be allowed, by explicitly considering those in the development of the new protocol instance: Indeed, when implementing a bug fix or feature update, the developer obviously knows the previous implementation. A message received at some process $p$ from some process $q$ in the new implementation just needs to be interpreted differently, depending on whether $q$ runs an old or a new protocol instance. Note that backward compatibility typically rules out incorporating a version number into the messages of the (new) protocol here, in which case $p$ would be uncertain about the actual status of $q$, despite having received a message from it. For light-weight low-level protocols, this approach might indeed constitute an attractive alternative to complex consistent update mechanisms.

Consider an heterogeneous distributed system with two agent-types, $U$ for the updated agents running the most recent software and $O$ for the agents running the old protocol, which is designed with the possibility of future updates in mind. Since the new protocols are designed by taking into account the existence of processes running the old protocol, the interpretation functions can be built asymmetrically. Each type interprets information from its own type directly: $f_{UU}(\varphi) = \varphi$ and $f_{OO}(\varphi) = \varphi$. $U$ agents can interpret messages from $O$ agents using backward compatibility $f_{UO}(\varphi) = g\big(f_{OO}(\varphi)\big)$, where $g$ translates into the updated system language.

The opposite is not always possible as $O$ agents have no knowledge of the new protocols. Accordingly, messages $\varphi$ compatible with the old protocol will be processed as before, i.e., using $f_{OO}(\varphi)$. But if $\varphi$ is unknown to the old protocol, i.e., $f_{OO}(\varphi) = \bot$, the creed under the assumption that sender $s \in O$ would yield $\mathbb{C}_s^{O\setminus O}\varphi \leftrightarrow \neg O_s$. In this case, receiver $r$ can conclude that the sender process $s$ does not conform to the old protocol. Since this error flagging disappears when $r$ is also updated, however, it may very well be the case that this does not violate the fault resilience properties of the old protocol, in particular, when not too many processes are updated simultaneously. In this case, $r$ could be guaranteed to always compute a correct result.

In the examples mentioned before, most of the interpretation functions were relatively simple, in the sense that they mapped a formula either to itself, its negation or bottom. We conclude this section by briefly mentioning two other modelings with a more elaborated interpretation function:

- According to Grice's maxims, in a conversation one should provide enough information, but not too much. If agent $q \in A$ asks agent $p \in B$ whether there is a gas station nearby ($\varphi$), and $p$ reply positively, then if $q$ believes $p$ to adehere to Grice's maxims, the interpretation function $f_{AB}(\varphi) = \varphi \wedge \psi$, where $\varphi \wedge \psi$ stands for there is a gas station nearby and is open;
- A distracted agent $q \in A$ can be modeled as having an interpretation function that, no matter the partition of the sender, interprets the received formula $\varphi$ into a disjunction of $\varphi \vee \psi$, for some other formula $\psi$.

## 3.5 Conclusions and future work

The idea of generalizing agents and group of agents has already been introduced in [GH93] and then extended in [BCR21]. This chapter follows the same tradition by associating a type to each agent in a static framework. From the dynamic epistemic logic [vDvdHK07] perspective, a public announcement logic with agent types is presented in [LW13], providing a dynamic framework to reason about uncertainty of agent-types that is used to formalize the knights and knaves puzzle. Due to the different motivations, while treating a closely related problems set, [BCR21] and [LW13] make different and at times incomparable choices regarding the postulates underlying the systems. For instance, a precondition for an announcement for an agent in [LW13] need not entail the agent knowing this precondition, which contradicts the fundamental Knowledge of Preconditions principle (Theorem 12) for distributed systems [Mos15]. On the other hand, all agents in [LW13] possess the same knowledge about each of the existing agent types, in particular, all agents share one common interpretation of messages from a particular type, an assumption in line with the rather centralized nature of updates in dynamic epistemic logic but less sensible for distributed systems.

We plan to extend the current framework in a twofold way: on the one hand, it would be extremely interesting to relax the assumption that interpretation functions are *static* and *commonly known*. Including interpretation functions in the language and allowing the possibility to dynamically change them allows representing agents adjusting (potentially in a private manner) to situations (messages) they cannot fully (or correctly) interpret. On the other hand, the creed modality assume the viewpoint of the receiver of information. Considering the viewpoint of the sender indeed could be fruitful for characterizing the expectations of an agent w.r.t. to how others might interpret its information, which allows to represent goal-oriented communication typical of heterogeneous systems. Finally, from a logic modeling perspective, the creed modality can be employed in the runs and systems framework (Section 1.2.3) making it the perfect candidate for modeling distributed algorithms that rely on heterogeneous protocols.

CHAPTER 4

# Global A priori Belief Updates

In the previous chapter, we studied what happens when relaxing a core a priori system assumption, namely common knowledge of (all) the protocols, leading to different logics. In the remaining chapters we will focus on a core mechanism of the bootstrapping cycle involved in the design of distributed systems, namely, *updating* of a priori beliefs.

In this chapter[1], we introduce a global a priori update mechanism[2], which simulates the intervention of the system designer aiming at correcting some unexpected behavior of some agents. In a nutshell, its goal is to correct the epistemic state of the involved agent(s), in a way that leaves the rest of the system untouched. The update mechanism proposed here is a synthesis method (Section 1.2.6), which, given a goal formula $\varphi$ in $\mathcal{K}45$, determines whether there is a suitable update model that makes $\varphi$ true, and that constructs such update model from $\varphi$.

While existing synthesis methods typically work in a language extended with quantifiers over updates, such as the Arbitrary Action Modal Logic [Hal13] and Arbitrary Arrow Update Modal Logic [DDvdHKK20] , we use a restriction of the basic doxastic language $\mathcal{L}^B$ (Equation (1.1)). In more detail, we assume that our system contains some fault-detection mechanism, which is able to detect some unexpected behavior, as well as some a priori reasoning capabilities which are able to encode the change of the model needed for correcting the erroneous behavior in a goal formula $\varphi$. Our update mechanism is responsible for changing the model such that $\varphi$ is true after the update. Since the goal formula represent an epistemic change of some agents' beliefs, we want the update to have *minimal side effects* w.r.t. (higher order) beliefs of other agents, with which we mean that: (i) if an agent is not involved at the first level of modal depth of a goal formula, then its beliefs are unchanged (ii) if an agent is involved in the goal formula but no propositional component is (for that agent), then that agent's propositional beliefs are unchanged.

[1] This chapter is based on joint work with Thomas Schögl and Roman Kuznets, [SKC24].

[2] By contrast, in the next chapter we will consider updates that are initiated by agents based on their *local* perspective.

For example, suppose that the system designer discovers a design mistake in the system, by checking the a posteriori beliefs of processes, because an agent, at some point of the execution, should have acquired a certain belief but failed to do so. The update modify the beliefs of that agent only. In Section 2.1, we called these updates global a priori updates, which are a priori only from the viewpoint of the agents involved (provided that they can detect such change, something that we do not assume here).

In such a synthesis task, one trivial way to make a formula true for an agent is to make that agent inconsistent. However, this is not a desired feature of our update mechanism, as it preserves consistency of agents whenever possible. To obtain this effect, we need to privatize the model. Unsurprisingly, this not only changes the underlying model substantially, but also breaks the CKM (Section 1.2.2) assumption. While it is possible to construct action model logic update models representing completely private communication [BR16] (see, e.g., Figure 4.3), a standardized update synthesis procedure does not exist. In addition, most existing synthesis methods do not address belief consistency preservation and minimal change [DDvdHKK20]. Finally, to simulate global a priori updates, fully private update models (akin to the ones of Figure 1.5 ) are required, which are not used in existing synthesis methods.

Our privatization is obtained by structuring the model in a very specific way: we use directed acyclic graphs, inspired by the completely private action models of Figure 1.5 . We first separate the actual world from the rest of the model (the former becomes the root of our update models) and stratify agents' beliefs starting from that point. Intuitively, the children of the actual world represent the direct beliefs of an agents, and the children of those represent the first-order beliefs of that agent about other agents, and so on. In our stratified models, not only we clearly distinguish between actually possible worlds and virtually possible worlds (Section 1.2.2) for any agent, but also avoid the CKM assumption by letting each agent access only a limited and private part of it[3], without losing the ability to reason about higher-order beliefs of other agents of arbitrary depths. In addition, the stratified structure complies with the crucial notion of local view of an agent, typical of distributed systems, which is only a limited portion of the global view of the system usually inaccessible to agents.

Our update mechanism also employs a novel update operation called *pointed updates*, which, compared to the product update operation of AML (Definition 21) does not apply to the full product of the Kripke model and the action model rather, we exploit our stratification method (and the distinction between actually possible worlds and virtually possible worlds) to determine when to form a world-event tuple and when not. Our updates are hence not applied to the whole model at once, as they are progressively applied only to those worlds that match a certain structure (other than a certain precondition).

Consequently, stratification not only represents the idea of each agent having a partial view of a Kripke model, but also distinguishes the actual world from any other worlds, by making it inaccessible, therefore endorsing a *fallibilistic* assumption, in accordance with the proposed a priori knowledge framework proposed in this thesis (Section 1.3): Making the actual world inaccessible enforces fallibility from an agent perspective, as no agent can be sure that the global

[3]A formal definition of privatization is provided in Section 4.3.

state of the system is captured in their limited view of it. This fallibilistic assumption allows agents to have false beliefs, but does not force them to: agents might still have an accurate view of the system during the execution, for example, because they consider worlds that are bisimilar to the actual world in their private cluster.

A downside of our global a priori update mechanism is that it accounts only for a limited range of goal formulas, restricted to *deterministic belief increase* (DBI) only, i.e., to conjunctions of (positive) modal operators. This limitation is due to the fact that preserving minimality and consistency becomes highly problematic for unrestricted goal formulas: on the one hand, negations of belief operators might introduce ignorance, potentially contradicting previously held beliefs. On the other hand, disjunctions of modal operators have multiple realizations, making the update non-deterministic. On the other hand, the proposed update mechanism is generally more efficient than AML and GAUL updates, as it deletes all states not reachable from the actual world, making the growth in size of the model at worst linear in the size of the goal formula after one update and decreasing starting from the second update based on the same modal syntactic tree (see Definition 100). By contrast, it is well known that repeated applications of AML and GAUL updates lead to the exponential growth of the model. It is also known that the model checking problem is PSPACE-complete here [AS13].

## 4.1 Formal preliminaries and motivation

The aim of this section is to illustrate what kind of updates are we interested in and what are the basic building blocks to their formalization. We introduce two motivational examples, a more informal one, representing a complex update scenario involving the nested belief of one agent, and a second one describing the use of these updates in distributed settings.

**Example 78** (Trojan example)**.** *Exhausted by the very long siege of Troy, the Aches started to doubt whether they will ever be victorious, and they are all waiting for someone to take action and come up with some plan q. In the camp, o approaches p, who was very discouraged after the latest failure. o, trying to lift up p' spirit, tells him that he overheard a boasting to Agamennones that he had a plan q. p knows that o would resort to lie in order to make him feel better, and dismisses the claims of the king of Ithaca, leaving him thinking that the lie had no effect. However, thinking about it twice (and knowing a very well), he chooses to believe o's words, and is now under the impression that o knows about the plan as well. However, he himself does not believe that a actually has a plan, as it was probably a boast in the face of Agamennones' arrogance. Thus, using p, o and a for the three heroes, o lie should change p beliefs to achieve the goal formula:*

$$\varphi = B_p(B_a q \wedge B_o B_a q \wedge B_o q) \tag{4.1}$$

In particular, this update should neither affect $a$' or $o$' actual beliefs nor making $B_p q$ true.

Since, ignorance about $q$ was common belief in the initial model $\mathcal{M}$, the public announcement of $\varphi$ would cause everybody's beliefs to become inconsistent, whereas a private message $B_a q \wedge B_o B_a q \wedge B_o q$ would cause the same inconsistency, but for $p$' beliefs only. This happens irrespectively of whether one uses the world-removing or arrow-removing updates.

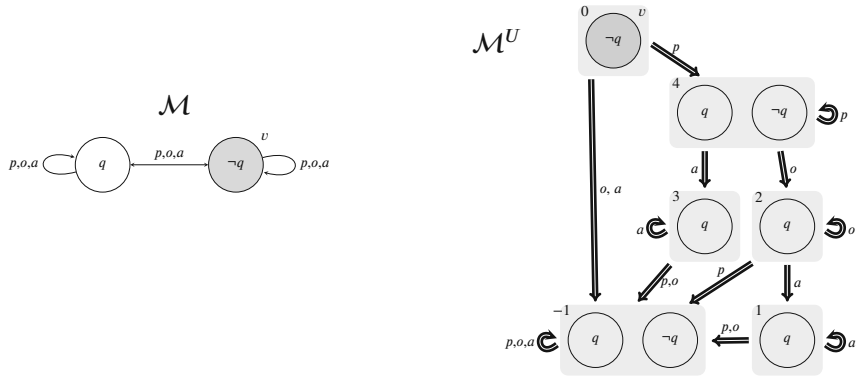



**Figure 4.1:** Left: Initial pointed Kripke model $\mathcal{M}$ for the trojan Example 78. Right: Updated model $\mathcal{M}^U$ (the actual world $v$ is dark gray). A formula $\varphi$ within a circle representing a world means that $\varphi$ is true in this world. A light-gray rectangle with $n$ in the top left corner is the cluster that is numbered $n$. A double arrow from such cluster $n$ to cluster $m$ represents a set of accessibility arrows with the same agent's label from every world of cluster $n$ to every world of cluster $m$.

We therefore propose an update method that stratifies the initial model $\mathcal{M}$ into several clusters representing individual beliefs and updating only some of these clusters, depending on the modal structure of the goal formula, resulting in model $\mathcal{M}^U$ in Figure 4.1, where updated clusters are represented by light-gray rectangles. Cluster 0 contains only the actual world $v$, which no agent considers possible. Cluster $-1$ (the *sink*) is the copy of the initial model representing (higher-order) beliefs of agents unaffected by the update, including $a$'s and $o$'s actual beliefs. In particular, we want the ignorance about $q$ still to be common belief between $a$ and $o$. $p$ is also ignorant about $q$: his beliefs are represented by cluster 1, so $\mathcal{M}^U, v \not\models B_p q \vee B_p \neg q$. But his ignorance is not anymore common with $p$ and $a$. Indeed, according to Cluster 2, which represents $p$' beliefs about $a$' beliefs, $\mathcal{M}^U, v \models B_p B_a q$. Cluster 3 plays the same role for $p$' beliefs about $o$' beliefs, making $\mathcal{M}^U, v \models B_p B_o q$. Thus, while himself being ignorant of the answer, $p$ thinks that $a$ and $o$ have chosen for themselves. Finally, Cluster 4 is $p$' beliefs about $o$' beliefs about $a$' beliefs so that $\mathcal{M}^U, v \models B_p B_o B_a q$. Overall, $\mathcal{M}^U, v \models \varphi$. The $p$-labeled double arrow from cluster 3 to the sink $-1$ represents two $p$-labeled arrows from the only world of cluster 3 to each of the two worlds of the sink and ensures that $p$ believes that $o$ believes that $p$ has not changed his beliefs, $\mathcal{M}^U, v \models B_p B_o(\neg B_p q \wedge \neg B_p \neg q)$. Similarly, the $p, o$-labeled double arrows from Clusters 2 and 4 to the sink signify that in no scenario where $a$ knows whether $q$, does she expect others to do the same (or even be aware of his choice). It is easy to check that no agent has inconsistent beliefs, $\mathcal{M}^U \models \neg B_p \bot \wedge \neg B_o \bot \wedge \neg B_a \bot$ and that $a$' and $o$' beliefs are the same as in the initial model, $M, v \models B_i \psi$ iff $\mathcal{M}^U, v \models B_i \psi$ for $i \in \{o, a\}$.

**Example 79** (Critical system)**.** *Suppose that in a simple asynchronous distributed system, two agents a and b do not initially know the value of a variable p, which, if it was believed to hold by all agents, would compromise the safety of the system. Suppose that due to an unforeseen error in the system b believes that p is true. The system designer immediately notices the error and*

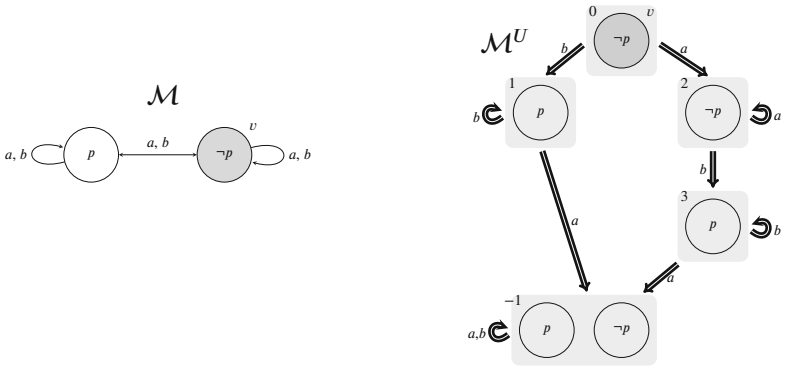



**Figure 4.2:** Left: Initial pointed Kripke model $\mathcal{M}$. Right: Updated model $\mathcal{M}^U$ (the actual world $v$ is dark gray). As in the previous figure a formula $\varphi$ within a circle representing a world means that $\varphi$ is true in this world. A light-gray rectangle with $n$ in the top left corner is the cluster that is numbered $n$. A double arrow from such cluster $n$ to cluster $m$ represents a set of accessibility arrows with the same agent's label from every world of cluster $n$ to every world of cluster $m$.

*performs an a priori update on a informing her $\neg p$ is the case, to prevent disastrous situations. Since it is common knowledge that truth of p for both should be avoided, the system designer decides also to inform a that b believes variable p is true, giving more reasons to a not to believe p even if b were to communicate so. At the same time, b does not know about the a priori update on a. The wanted update can be modeled with the goal formula $\theta := B_b p \wedge B_a(\neg p \wedge B_b p)$. Notice that we are modeling also the system error as an a priori update, and this operation is legitimate because from the viewpoint of the the agents there is virtually no difference (see Section 2.1).*

As for the previous example, we show the initial pointed Kripke model and the desired updated model in Figure 4.2, showing the stratification in the wanted clusters, represented by light-gray rectangles. We later show how the corresponding action model is constructed from the goal formula in Example 79. Cluster 0 contains only the actual world $v$, which no agent considers possible. Cluster $-1$ (the sink) is the copy of the initial model representing (higher-order) beliefs of agents unaffected (or believed to be unaffected) by the update. In this example, no agent is ignorant about the value of $p$ after the update (as represented by Cluster 1 for $b$ and Cluster 2 for $a$). Cluster 3 represents $a$'s beliefs about $b$ concerning the truth of $p$. In this case $b$ believes $a$'s beliefs are unaffected, and $a$ believes that $b$ believes that $a$'s beliefs are unaffected, as represented by the double arrows from cluster 1 to the sink $-1$ and from cluster 3 to the sink $-1$ for $b$ and $a$ respectively.

We now proceed with some preliminary definitions.

**Definition 80.** *The set of* target agents *of a modal formula $\varphi$ is defined as follows:* $\mathsf{ta}(p) := \emptyset$*;* $\mathsf{ta}(\neg\varphi) := \mathsf{ta}(\varphi)$*;* $\mathsf{ta}(\varphi \wedge \psi) := \mathsf{ta}(\varphi) \cup \mathsf{ta}(\psi)$*; finally* $\mathsf{ta}(B_i\varphi) := \{i\}$.

E.g., $\mathsf{ta}\big(B_p(B_a q \wedge B_o B_a q \wedge B_o q)\big) = \{p\}$ even though $\mathsf{ta}(B_a q \wedge B_o B_a q \wedge B_o q) = \{a, o\}$.


**Figure 4.3:** Left: Initial (pointed) Kripke model $\mathcal{M}$ where agent $a$ and $b$ are uncertain about whether $p$ or $\neg p$ is true. Right: (Pointed) action model $U$ for a private announcement $p$ to $a$: agent $a$ learns $p$, while $b$ believes that nothing happened. A formula $\varphi$ within a square representing an event $\beta$ is the precondition for this event, i.e., $\mathsf{pre}(\beta) = \varphi$.

It is well known that repeatedly applying product updates for multi-round communication can cause exponential blow up of the domain of the model, even for simple models and action models. Consider for example Figures 4.3–4.5: Indeed, every updated model has exactly one world where $p$ is false. Hence, a further update of a model of $N$ worlds with $U$ creates a model with $2N - 1$ worlds, as we create two copies of the initial domain, one that is identical and one where all the $\neg p$ worlds are deleted.

The standard solution is to reduce the size of the resulting model by using bisimulation contraction [WYL13], making the model smaller again. For example, bisimulation contraction deletes all worlds that are unreachable from the resulting actual world, resulting in a bisimilar and hence, modally equivalent [BdRV01], model.

To avoid this two-step procedure, we introduce the notion of *pointed updates*, which incorporates this world-removing operation:

**Definition 81** (Pointed update)**.** *Let* $(\mathcal{M}, w) = (\langle S, R, V\rangle, w)$ *be a pointed Kripke model and* $(U, \alpha) = (\langle E, Q, \mathsf{pre}\rangle, \alpha)$ *be a pointed action model. If* $\mathcal{M}, w \vDash \mathsf{pre}(\alpha)$*, then we define the updated pointed Kripke model* $\big(\mathcal{M} \odot U, (w, \alpha)\big)$ *where* $\mathcal{M} \odot U := \langle S^U, R^U, V^U\rangle$ *as follows. Let* $T^U := \big\{(v, \beta) \in S \times E \mid \mathcal{M}, v \vDash \mathsf{pre}(\beta)\big\}$*. Note that* $(w, \alpha) \in T^U$*.*

- *The domain* $S^U$ *is the smallest subset of* $T^U$ *containing* $(w, \alpha)$ *such that: if* $(v, \beta) \in S^U$*,* $(u, \gamma) \in T^U$*, and both* $vR_iu$ *and* $\beta Q_i\gamma$ *for some agent* $i$*, then* $(u, \gamma) \in S^U$*;*
- $R_i^U := \big\{\big((v, \beta), (u, \gamma)\big) \in S^U \times S^U \mid (v, u) \in R_i \text{ and } (\beta, \gamma) \in Q_i\big\}$*;*
- $V^U(p) := \big\{(v, \beta) \in S^U \mid v \in V(p)\big\}$*.*

Similarly to what is done in [BHH21], we show that pointed updates produce the same result as product updates (but with smaller models) using bisimulation (Definition 24)

**Theorem 82** (Product and pointed updates are bisimilar)**.** *Given a pointed Kripke model* $(\mathcal{M}, w)$ *and pointed action model* $(U, \alpha)$ *such that* $\mathcal{M}, w \vDash \mathsf{pre}(\alpha)$*, we have* $\mathcal{M} \odot U, (w, \alpha) \leftrightarroweq \mathcal{M} \otimes U, (w, \alpha)$*.*

The theorem follows by construction of the pointed update: given $\mathcal{M} = \langle S, R, V\rangle$, $U = \langle E, Q, \mathsf{pre}\rangle$, $\mathcal{M} \odot U = \langle S^U, R^U, V^U\rangle$ and $\mathcal{M} \otimes U = \langle S', R', V'\rangle$, $(w, \alpha) \in S^U \subseteq S'$, and $R^U = R'_{\restriction (S^U \times S^U)}$, and $V^U = V'_{\restriction S^U}$. Thus, $\mathscr{B} := \big\{\big((v, \beta), (v, \beta)\big) \mid (v, \beta) \in S^U\big\}$ is the requisite bisimulation.


**Figure 4.4:** Left: First product update $\mathcal{M} \otimes U$ of Kripke model $\mathcal{M}$ with action model $U$. The label of the resulting worlds is made by the label of the world of the Kripke model plus the label of the event applied to that world. Right: Second product update $(\mathcal{M} \otimes U) \otimes U$ with the same action model $U$. $p$ is true in all worlds that start from $p$ and false in all worlds that start from $\neg p$.

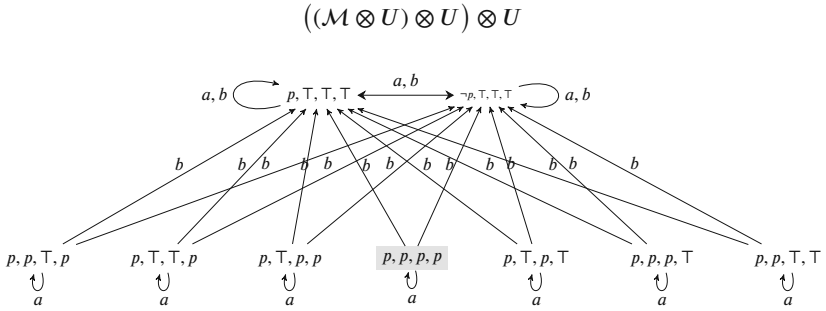



**Figure 4.5:** Third product update $\big((\mathcal{M} \otimes U) \otimes U\big) \otimes U$ with the same action model $U$.

**Theorem 83** (Updates preserve bisimilarity)**.** *Given bisimilar pointed Kripke model $\mathcal{M}, w \leftrightarroweq \mathcal{M}', w'$ and bisimilar pointed action models $U, \alpha \leftrightarroweq U', \alpha'$ such that $\mathcal{M}, w \vDash \mathsf{pre}(\alpha)$, we have $\mathcal{M} \otimes U, (w, \alpha) \leftrightarroweq \mathcal{M}' \otimes U', (w', \alpha')$ and $\mathcal{M} \odot U, (w, \alpha) \leftrightarroweq \mathcal{M}' \odot U', (w', \alpha')$. The same holds for G-bisimilarity (Definition 24).*

*Proof.* For product updates, the proof can be found in [vDvdHK07]. For pointed updates, it follows from Theorem 82. The statements for $a$-bisimilarity (Definition 24) easily follow. □

While our method cannot prevent an exponential blowup for all classes of models, it has the clear advantage that the output model is no bigger than the one obtained via standard product update. This advantage is clear in case of iterated updates: while repeated product updates of $\mathcal{M}$ from Figure 4.1 with $U$ from Figure 4.3 lead to the exponential growth of the domain, pointed updates yield the three-world pointed Kripke model depicted left in Figure 4.4, no matter how many times the pointed update is performed:

$$\big((\mathcal{M} \odot U) \odot U\big) \odot \cdots \odot U = \mathcal{M} \otimes U.$$

While bisimulation contraction [WYL13] can shrink the model obtained by the product update operation, it needs to be invoked on the resulting model in a second step. In other words, first the model needs to be computed (with a potential exponential blowup) and then the bisimulation contraction runs on that model, making the operation computationally costly. On the other hand,

the pointed update operation that we introduce runs only once and it produces a model which is no bigger than the one that would be obtained by the product update operation.

In the following section, we use the newly introduced mechanism to solve the consistent update synthesis problem.

## 4.2 Consistent update synthesis: deterministic belief increase

This section is dedicated to the consistent update synthesis task for *goal formulas* $\varphi$ representing *global* a priori updates. Recall that we restrict these kind of updates to *deterministic belief increase* (DBI), which means that we adopt only conjunction of beliefs as a goal formula.

**Definition 84** (DBI goal formulas, DBI normal form)**.** DBI goal formulas $\varphi$*, or* DBI formulas *for short, are of the following form:*

$$\varphi ::= B_i\xi \mid B_i(\xi \wedge \varphi) \mid (\varphi \wedge \varphi) \mid B_i\varphi, \tag{4.2}$$

*where $\xi$ is any purely propositional formula and $i \in \mathcal{A}$. In addition, we further restrict the set of allowed formulas by disallowing nesting of belief operators for the same agent. Thus, each DBI goal formula is a non-empty conjunction of belief operators.* Formulas in DBI normal form *are DBI formulas $\varphi$ obtained by restricting the construction as follows: $B_i\xi$ is always DBI normal; $B_i\varphi$ and $B_i(\xi \wedge \varphi)$ are DBI normal iff $\varphi$ is DBI normal and $i \notin \mathsf{ta}(\varphi)$; $\varphi \wedge \psi$ is DBI normal iff $\varphi$ and $\psi$ are DBI normal and $\mathsf{ta}(\varphi) \cap \mathsf{ta}(\psi) = \varnothing$.*

**Lemma 85.** *For any DBI goal formula $\varphi$, there exists a DBI normal formula $\varphi'$ such that $\mathcal{K}45 \vDash \varphi \equiv \varphi'$, where $\mathcal{K}45$ is the class of all transitive and euclidean frames.*

*Proof.* The proof is by induction on the construction of the formula. **Base case:** It follows by definition that $\theta := B_i\xi$ is always DBI normal form. If $\theta := B_i(\xi \wedge \varphi)$ then since $B_i\theta \wedge B_i\eta \equiv B_i(\theta \wedge \eta)$ in $\mathcal{K}45$, then $B_i\xi$ is in DBI normal form by previous case and since $i \notin \mathsf{ta}(\varphi)$ (by the above restriction), by IH $\theta$ is in DBI normal form. If $\theta := B_i(\varphi \wedge \psi)$ then since $\mathsf{ta}(\eta) \cap \mathsf{ta}(\psi) = \varnothing$ and $i \notin \mathsf{ta}(\varphi), \mathsf{ta}(\psi)$ (by the above restriction), by IH $\theta$ is in DBI normal form. Finally if $\theta := B_i\varphi$ then $\theta$ is in DBI normal form by IH. □

**Lemma 86** ($G$-bisimilarity and goal formula preservation)**.** *If $\mathcal{M}, v \leftrightarrows_G \mathcal{M}', v'$ and $\varphi$ is a DBI formula with $\mathsf{ta}(\varphi) \subseteq G$, then $\mathcal{M}, v \vDash \varphi$ iff $\mathcal{M}', v' \vDash \varphi$.*

*Proof.* The statement follows from Theorem 26 (*2*) and from the fact that goal formulas are conjunction of $B_i\psi_i$ for $i \in \mathsf{ta}(\varphi) \subseteq G$. □

The overall model update mechanism consists in generating a pointed action model $U_\varphi$ such that the result of a pointed update of any given pointed Kripke model with $U_\varphi$ satisfies $\varphi$, without the introduction of (potentially also higher-order) inconsistent beliefs. Indeed,the main novelty in our update synthesis method is the aim to preserve consistency whenever possible. In the rest of the chapter we only consider formulas in DBI normal form. For instance, formula (4.1) is a DBI

formula but not DBI normal. Its normal form would be $B_b\big(B_t p \wedge B_l(p \wedge B_t p)\big)$. Given such a DBI normal formula $\varphi$, we now construct a pointed action model $(U_\varphi, 0)$ such that for any pointed Kripke model $(\mathcal{M}, w)$, the pointed update $\mathcal{M} \odot U_\varphi$ is defined, $\mathcal{M} \odot U_\varphi, (w, 0) \vDash \varphi$, while other epistemic differences between $(\mathcal{M}, w)$ and $\big(\mathcal{M} \odot U_\varphi, (w, 0)\big)$ are minimized. Informally, this means that an update should not influence any beliefs except for those explicitly stated in $\varphi$ and logically following from $\varphi$ based on the agents' pre-update beliefs [4]. In particular, if $j \notin \mathsf{ta}(\varphi)$, then $j$'s beliefs should not be affected even if $B_j$ occurs in $\varphi$: updating somebody else's beliefs about $j$'s beliefs should not affect actual $j$'s beliefs.

The following definition explicitly formulates how a pointed action model is generated from a formula in DBI normal form. Every such action model has (i) an actual event labeled 0, that is inaccessible to every agent, and (ii) a sink event labeled $-1$ that represents the previous state of affairs. As 0 only has outgoing arrows (and no reflexive loops) and $-1$ only has incoming arrows (excluding reflexive loops) an arrow from 0 to $-1$ represents the fact that the beliefs of that agent are unchanged. In between 0 and $-1$ there are clusters of (stratified) events as there are (nested) belief operators in the goal formula: intuitively, each operator forms a cluster of events (the ones that will form the possible worlds where the beliefs of that agent hold). For each nested belief operator, another layer of events between the previous cluster and the $-1$ event is introduced. We now introduce the formal definition and then proceed with further explanations constructively with an example.

**Definition 87** (Consistent DBI update synthesis)**.** *Given a formula $\varphi$ in DBI normal form, the pointed action model $(U_\varphi, 0) = (\langle E^\varphi, Q^\varphi, \mathsf{pre}^\varphi\rangle, 0)$ is constructed recursively based on the structure of $\varphi$. In all cases, $E^\varphi = \{0, -1\} \sqcup D^\varphi$ with $0 \neq -1$ for some $\varnothing \neq D^\varphi \subseteq \mathbb{N}$, event 0 representing the actual event, unknowable for all, has no incoming arrows, i.e., not $\alpha Q_i^\varphi 0$, and event $-1$ representing status quo/no change has no outgoing arrows other than reflexive loops, i.e., $-1 Q_i^\varphi \alpha$ iff $\alpha = -1$. $\mathsf{pre}^\varphi(0) = \mathsf{pre}^\varphi(-1) := \top$. Let $\underline{Q}_j^\varphi := Q_j^\varphi \cap \big((E^\varphi \setminus \{0\}) \times (E^\varphi \setminus \{0\})\big)$ be the restriction of $Q_j^\varphi$ onto $E^\varphi \setminus \{0\}$ for any agent $j$.*

**1** *If $\varphi = B_i\xi$ for a propositional formula $\xi$, then $D^{B_i\xi} = \{m\}$ for a fresh $m \geq 1$, $\mathsf{pre}^{B_i\xi}(m) := \xi$, $Q_j^{B_i\xi} := \{(0,-1),(m,-1),(-1,-1)\}$ for any $j \neq i$, while at the same time $Q_i^{B_i\xi} := \{(0,m),(m,m),(-1,-1)\}$.*

**2** *If $\varphi = B_i\psi$ for a DBI normal formula $\psi$, then $U_\psi = \langle \{0,-1\} \sqcup D^\psi, Q^\psi, \mathsf{pre}^\psi\rangle$ has already been constructed. Let $1 \leq m \notin D^\psi$ be fresh. Then $D^\varphi := D^\psi \sqcup \{m\}$, $\mathsf{pre}^\varphi := \mathsf{pre}^\psi \sqcup \{(m, \top)\}$, i.e., $\mathsf{pre}^\varphi$ agrees with $\mathsf{pre}^\psi$ and extends it to $\mathsf{pre}^\varphi(m) := \top$, and the accessibility relations are defined as follows:*

$$Q_j^\varphi := \underline{Q}_j^\psi \sqcup \{(0,-1)\} \sqcup \{(m,k) \mid (0,k) \in Q_j^\psi\} \qquad \text{for any } j \neq i;$$
$$Q_i^\varphi := \underline{Q}_i^\psi \sqcup \{(0,m),(m,m)\}.$$

[4] We adopt the minimality principle in the same spirit of the standard AGM approach [vBS15], for which an update should lead to the loss of as few previous beliefs as possible [Han22].

**3** *If $\varphi = B_i(\xi \wedge \psi)$ for a DBI normal formula $\psi$ and propositional formula $\xi$, the construction is the same as in the preceding case with the only change being that* $\mathsf{pre}^{\varphi}(m) := \xi$.

**4** *If $\varphi = \psi \wedge \theta$ for DBI normal formulas $\psi$ and $\theta$, then $U_{\psi} = \langle \{0,-1\} \sqcup D^{\psi}, Q^{\psi}, \mathsf{pre}^{\psi} \rangle$ and $U_{\theta} = \langle \{0,-1\} \sqcup D^{\theta}, Q^{\theta}, \mathsf{pre}^{\theta} \rangle$ have already been constructed. We can assume w.l.o.g. that $D^{\psi} \cap D^{\theta} = \varnothing$ (otherwise, just rename events in one of them). Take $D^{\varphi} = D^{\psi} \sqcup D^{\theta}$, set $\mathsf{pre}^{\varphi} = \mathsf{pre}^{\psi} \cup \mathsf{pre}^{\theta}$ and, for any $j$,*

$$Q_j^{\varphi} := \underline{Q}_j^{\psi} \cup \underline{Q}_j^{\theta} \sqcup \{(0,k) \mid (0,k) \in Q_j^{\psi} \cup Q_j^{\theta}, k \in D^{\psi} \sqcup D^{\theta}\} \sqcup \{(0,-1) \mid (0,k) \notin Q_j^{\psi} \cup Q_j^{\theta} \text{ for any } k \in D^{\psi} \sqcup D^{\theta}\}.$$

To better illustrate our update synthesis method, we show how it applies to our motivational examples.

We start by illustrating how we construct an action model from a goal formula using Definition 87 for our Example 79. We then show the whole update synthesis method for our Example 78.

**Example 88** (Constructing a pointed action model in the Critical system example)**.** *For the DBI normal form $\theta = B_b p \wedge B_a(\neg p \wedge B_b p)$ from Example 79, the stages of construction of the pointed action model $(U_{\theta}, 0) = (\langle E^{\theta}, Q^{\theta}, \mathsf{pre}^{\theta} \rangle, 0)$ constructed according to Definition 87 can be found in Figure 4.6. To illustrate how the construction of the update model works we show the procedure step by step. In the first instance, events 0 and $-1$ are added to $E^{\theta}$, and $pre^{\theta}(0) = pre^{\theta}(-1) = \top$, with $-1 Q_i^{\theta} - 1$ for all $i \in \mathcal{A}$. We then proceed following the structure of the formula. The formula falls under item **4** of Definition 87. This means that we have to construct an action model for each conjunct separately and then go back to item **4**.*

*(i) We start by constructing the action model $U_{\theta_1}$ for the first conjunct $\theta_1 = B_b p$ falls under item **1** of Definition 87. then $D^{B_b p} = \{1\}$, $pre^{B_b p}(1) = p$, which means that we add a new event 3 and we set its precondition to p. As for relations, $Q_a^{B_b p} = (0,-1),(1,-1),(-1,-1)$ i.e., beliefs of a are unchanged, so they range from the actual event 0 to the sink, and the higher order beliefs of b about a range from the new event 1 to the sink; while $Q_b^{B_b p} = (0,1),(1,1),(-1,-1)$, representing b beliefs from 0 to 1, and its previous beliefs are preserved in the sink, used to represent a's higher order beliefs. The resulting action model $U_{\theta_1}$ is shown in Figure 4.2 (left).*

*(ii) The second conjunct $\theta_2 = B_a(\neg p \wedge B_b p)$ falls under item **3** of Definition 87. (ii.a) The first part of the second conjunct $B_a \neg p$ is analogous to the previous case (i), except that $D^{B_a \neg p} = \{2\}$ $pre^{B_a \neg p}(2) = \neg p$. Relations are defined similarly and follow similar justifications. (ii.b) The second part of the second conjunct $B_a B_b p$ falls into case **2** of Definition 87, which then calls case **1** again for the formula $B_b p$, making it analogous to case (i) but with $D^{B_b p} = \{3\}$ and then back again to case **2**, so we add a new event to the one of the previous case (ii.a) $D^{B_a B_b p} = \{2\} \sqcup \{3\}$, and $pre^{B_a B_b p} = pre^{B_b p} \sqcup \{(3,\top)\}$ meaning that we only change the precondition of 3 and we leave the precondition of 2 untouched. Accessibility relations are defined as follows: $Q_b^{B_a B_b p} = Q_b^{B_b p} \sqcup \{(0,-1)\} \sqcup (2,3)$, i.e. the relations for agent b in the formula $B_a B_b p$ are given by the relations of constructed event from formula $B_b p$ of the preceding case, namely $(3,3)$ plus a b-relation connecting the nested event 3 to its parent 2 $(2,3)$ and the b-relation to the sink*

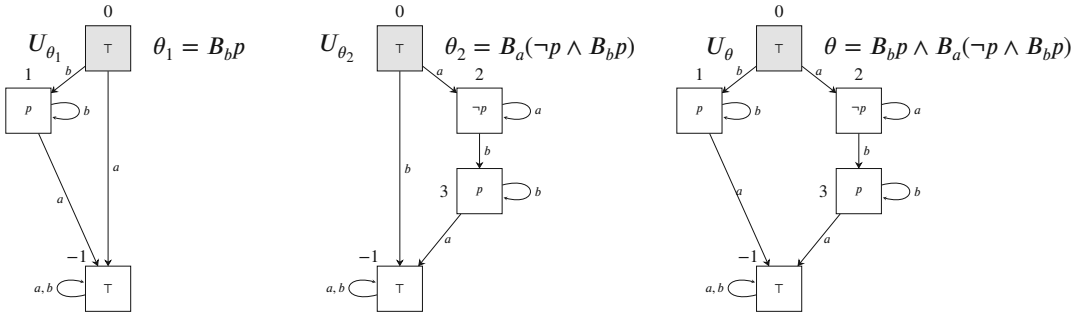



**Figure 4.6:** Intermediate stages (left, middle) and final result (right) in constructing action model $U_\theta$ for DBI formula $\theta$ from Example 79.

$(0,-1)$. *For agent a* $Q_a^{B_a B_b p} = Q_a^{B_b p} \sqcup \{(0,2)\} \sqcup (2,2)$, *i.e., the relations for a in the formula* $B_a B_b p$ *are given by the a-relations of the formula* $B_b p$ *of the preceding case, so only* $(3,-1)$ *plus a relation* $(0,2)$ *and corresponding reflexive loop* $(2,2)$. *The resulting action model* $U_{\theta_2}$ *is shown in Figure 4.2 (middle).*

*Finally, we can go back to the starting point item* ***4*** *of Definition 87. The domain* $D^\theta = D^{\theta_1} \sqcup D^{\theta_2}$, *and the resulting preconditions are preserved. Finally, the resulting relations for all agents are made by the previous relations excluding the ones from* $(0,-1)$ *(as indicated by the* $\underline{Q}_i$ *restriction), unless i is an agent not involved in the update, for which relations from* $(0,-1)$ *are preserved and/or added (no agents in this example), resulting in the model Figure 4.6 (right).*

**Example 89** (Trojan example continued)**.** *For the DBI normal form* $\varphi = B_p(B_a q \wedge B_o(q \wedge B_a q))$ *of formula* (4.1) *from Example 78, the pointed action model* $(U_\varphi, 0)$ *constructed according to Definition 87 can be found in the left part of Figure 4.8 (the stages of construction are depicted in Figure 4.7). The result of the pointed update of* $(\mathcal{M}, v)$ *from the left part of Figure 4.1 with this* $(U_\varphi, 0)$ *is shown on the right of Figure 4.8 (and is isomorphic to the right part of Figure 4.1). It is easy to see that* $\mathcal{M} \odot U_\varphi, (v,0) \vDash \varphi$. *Since all accessibility relations in the updated model* $\mathcal{M} \odot U_\varphi$ *are serial, i.e., for every world w there is a world u such that* $wR_i u$, *it follows that it is common belief that all agents have consistent beliefs.*

*In the resulting model, since no agent ended up with false beliefs, and since o's and a's views of the model are unchanged, this update synthesis has been achieved with minimal change w.r.t. agents' beliefs*[5]*: it is easy to check using G-bisimilarities for appropriate groups of agents that*

$$\mathcal{M}, v \equiv_{o,a} \mathcal{M} \odot U_\varphi, (v,0) \qquad \mathcal{M}, v \equiv_{p,o} \mathcal{M} \odot U_\varphi, (u,1)$$
$$\mathcal{M}, v \equiv_{p} \mathcal{M} \odot U_\varphi, (u,2) \qquad \mathcal{M}, v \equiv_{p,o} \mathcal{M} \odot U_\varphi, (u,3)$$

*where v is the actual world and u is the other world of* $\mathcal{M}$ *(we omit set braces in the subscript of* $\equiv$*). The statements in the first column mean that the update is imperceptible for agents o and a and that agent p believes that agent o does not think p's beliefs have changed. Similarly, the*

[5] We make this claim more precise in Theorem 95.

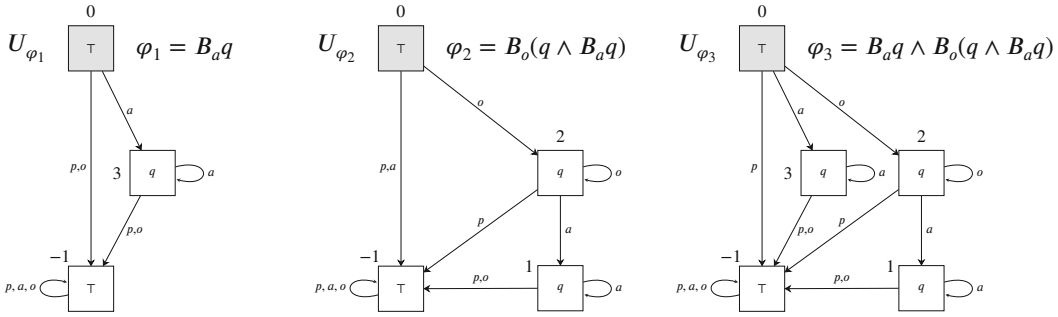



**Figure 4.7:** Intermediate stages in constructing action model $U_\varphi$ for DBI formula (4.1) from Example 78 based on its DBI normal form $\varphi = B_a q \wedge B_o(q \wedge B_a q)$.

*second column testifies that p believes himself and thinks that o believes that a does not think either p or o changed their beliefs. In other words, the only higher-order beliefs that are affected by the update are those explicitly dictated by φ.*

We illustrate the advantage of using pointed updates over the product update operation by applying the latter to our motivational example:

**Example 90** (Trojan example in DEL)**.** *For comparison purposes, we consider Example 78, but using the product update of standard DEL: The result of updating the initial model with the action model obtained from φ is shown in Figure 4.9. In particular, the result of the product update has only one additional world w.r.t. the model obtained via pointed updates $\mathcal{M} \odot U_\varphi$ of Figure 4.8 (left). However, applying again the product update operation to the resulting model using with the same action model leads to the Kripke model sketched in Figure 4.10. Compare this to the pointed update operation, whose result is the same for any number of iterations for the same action model (Theorem 99).*

Let us now prove that $U_\varphi$ performs a correct update synthesis for any DBI normal $\varphi$, minimizes belief change of other agents, and preserves consistency whenever possible.

First, note that updates do not affect propositional formulas:

**Lemma 91** (Propositional invariance)**.** *Let $(w, \alpha)$ belong to the domain of $\mathcal{M} \odot U$ for some Kripke model $\mathcal{M}$ and action model $U$. Then, for any propositional formula $\xi$,*

$$\mathcal{M}, w \vDash \xi \qquad \Longleftrightarrow \qquad \mathcal{M} \odot U, (w, \alpha) \vDash \xi. \tag{4.3}$$

*Proof.* For product updates $\otimes$, this is well known (see, e.g., [vDvdHK07]). For pointed updates, it follows from Theorem 82. □

**Theorem 92** (Correct update synthesis)**.** *For any DBI normal formula φ and any pointed Kripke model $(\mathcal{M}, v)$, the pointed update of $(\mathcal{M}, v)$ with $(U_\varphi, 0)$ is defined and*

$$\mathcal{M} \odot U_\varphi, (v, 0) \vDash \varphi. \tag{4.4}$$

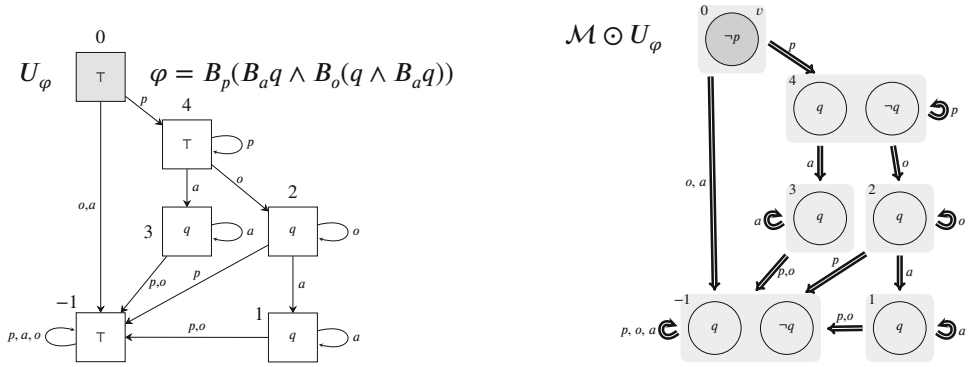



**Figure 4.8:** Successful update synthesis $\mathcal{M} \odot U_\varphi, (v, 0) \vDash \varphi$ by applying the pointed update with pointed action model $(U_\varphi, 0)$ for DBI formula (4.1) from Example 78 to $(\mathcal{M}, v)$ from Figure 4.1 (left). The real event and world are dark gray; light gray rectangles are a compact way for representing arrows in the updated model: a double arrow with label $a \in \mathcal{A}$ from one rectangle to another represents a set of accessibility arrows with label $a$ from every world in the source rectangle to every world in the target rectangle.

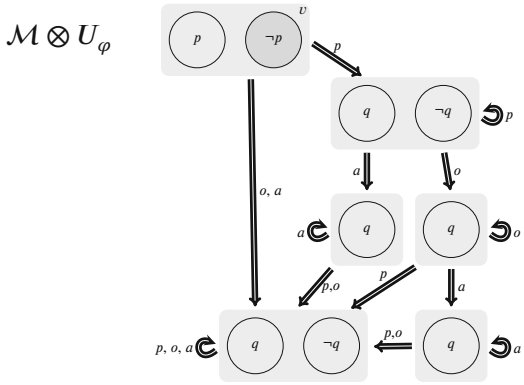



**Figure 4.9:** Updated model from Example 90 using product updates $\mathcal{M} \otimes U_\varphi$. The real world $v$ is underlined. Grey rectangles are a compact way for representing arrows: A double arrow from a rectangle to another represents a set of accessibility arrows with the same agent's label from every world in the source rectangle to every world in the target rectangle.

*Proof.* Let $\mathcal{M} = \langle S, R, V \rangle$ with $v \in S$. Per Definition 87, $U_\varphi = \langle E^\varphi, Q^\varphi, \mathsf{pre}^\varphi \rangle$. The pointed update $\mathcal{M} \odot U_\varphi, (v, 0)$ is defined because $\mathsf{pre}^\varphi(0) = \top$ always holds. Let $\mathcal{M} \odot U_\varphi = \langle W^\varphi, R^\varphi, V^\varphi \rangle$. We prove (4.4) by induction on the construction of $\varphi$. For $\varphi = B_i \xi$ with propositional $\xi$, whenever $(v, 0) R_i^\varphi (u, m)$, then $\mathcal{M}, u \vDash \xi$ because $\mathsf{pre}^\varphi(m) = \xi$. Hence, $\mathcal{M} \odot U_\varphi, (u, m) \vDash \xi$ by Lemma 91. We conclude (4.4) for $\varphi = B_i \xi$ since $(u, m)$ was chosen arbitrarily.

For $\varphi = B_i \psi$, since $\psi$ is a simpler formula than $\varphi$, then the action model constructed by definition 87 $U_\psi$ is a simpler action model than $U_\varphi$: they both have identical 0 and the $-1$ events, and since each modal operator correspond to an event, $U_\varphi$ has exactly one more event than $U_\psi$. In particular, the nesting of belief operators correspond to a stratification in the respective action

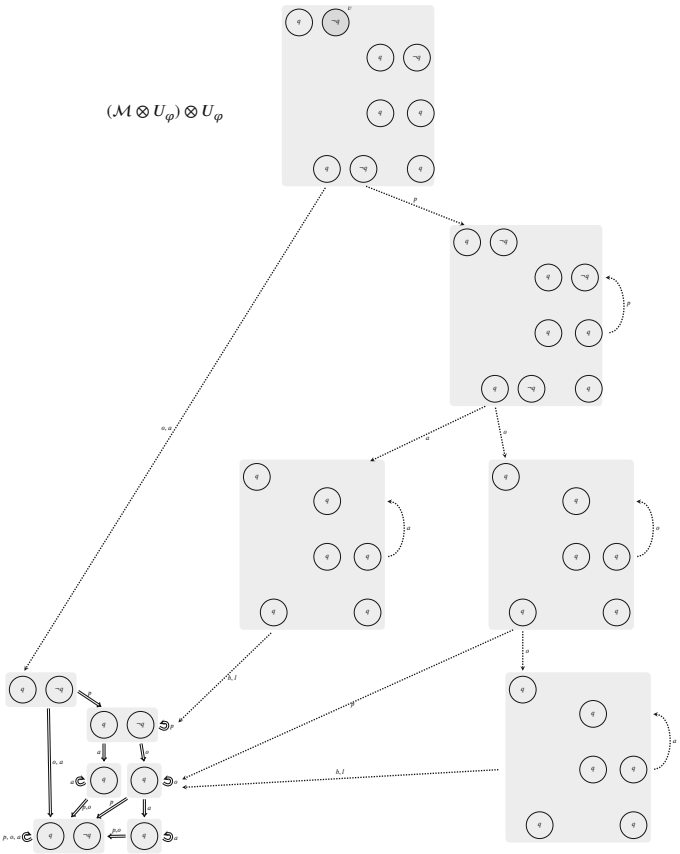



**Figure 4.10:** Sketch of the result of the iterated product update operation $(\mathcal{M} \otimes U_\varphi) \otimes U_\varphi$. Grey rectangles are a compact way to represent sub-models originating from worlds in the previous model. Dotted arrows are meant to capture the direction of the arrows and the agents involved. Unlike double arrows, they do not represent arrows from and to all worlds in the rectangle, but only from and to some of these worlds. We leave to the reader their identification. Finally, note that the sink is a copy of the previous model.

models, which means that $U_\varphi, m \leftrightarrows_{\mathcal{A}\setminus\{i\}} U_\psi, 0$ for any $m \neq 0$. Whenever $vR_iu$, since pointed updates preserve bisimilarity $\mathcal{M} \odot U_\varphi, (u, m) \leftrightarrows_{\mathcal{A}\setminus\{i\}} \mathcal{M} \odot U_\psi, (u, 0)$ by Theorem 83. By IH, $\mathcal{M} \odot U_\psi, (u, 0) \vDash \psi$. By normality of $\varphi$, $i \notin \mathsf{ta}(\psi)$. Hence, $\mathcal{M} \odot U_\varphi, (u, m) \vDash \psi$ by Lemma 86. Now (4.4) for $\varphi = B_i\psi$ follows since $(v, 0)R_i^\varphi(u, \alpha)$ iff $vR_iu$ and $\alpha = m$ due to $\mathsf{pre}^\varphi(m) = \top$. The case of $\varphi = B_i(\xi \wedge \psi)$ is analogous, except additionally that $\mathcal{M} \odot U_\varphi, (u, m) \vDash \xi$ because $\mathsf{pre}^\varphi(m) = \xi$, as in the case of $B_i\xi$.

For $\varphi = \psi_1 \wedge \psi_2$, by construction and normality of $\varphi$ we have $U_\varphi, 0 \leftrightarrows_{\mathsf{ta}(\psi_j)} U_{\psi_j}, 0$ and, hence, $\mathcal{M} \odot U_\varphi, (v, 0) \leftrightarrows_{\mathsf{ta}(\psi_j)} \mathcal{M} \odot U_{\psi_j}, (v, 0)$ for $j = 1, 2$. Since $\mathcal{M} \odot U_{\psi_j}, (v, 0) \vDash \psi_j$ for $j = 1, 2$ by IH. $\mathcal{M} \odot U_\varphi, (v, 0) \vDash \psi_j$ for $j = 1, 2$ by Lemma 86. Now (4.4) for $\varphi = \psi \wedge \theta$ follows immediately. □

We call a sequence of modal operators of a formula $\varphi$ $B_{i_1} \dots B_{i_k}$ a ordered list of modal operators occurring in that formula, including the empty sequence. For example, given $\varphi = B_i(B_j p \wedge B_r q)$, the set of sequences of modal operators in $\varphi$, labeled $\mathcal{B}^\varphi$, is $\epsilon$, $B_i$, $B_i B_j$, $B_i B_r$.

To formulate statements about minimal change, we introduce the notion of independent formulas.

**Definition 93** (Independent formulas)**.** *Let $\varphi$ be a DBI normal formula, and its corresponding action model $U_\varphi = \langle E^\varphi, Q^\varphi, \mathsf{pre}^\varphi \rangle$ be constructed according to Definition 87. A modal formula $\theta$ is in* $\top$*-shape w.r.t. event $\alpha_0 \in E^\varphi$ iff for each sequence of modal operators $B_{i_1} \dots B_{i_k}$ used in the construction of $\theta$, including the empty sequence $\epsilon$ with $k = 0$, and for the unique*[6] *corresponding sequence $\alpha_0 Q^\varphi_{i_1} \alpha_1 Q^\varphi_{i_2} \alpha_2 \dots \alpha_{k-1} Q^\varphi_{i_k} \alpha_k$ of events $\alpha_j \in E^\varphi$ all $\mathsf{pre}^\varphi(\alpha_j) = \top$ for $0 \leq j \leq k$. Formula $\theta$ is called* independent *of $\varphi$ iff it is in $\top$-shape w.r.t. $0 \in E^\varphi$. If $\theta$ is not independent of $\varphi$, it is* dependent *on $\varphi$.*

In other words, given a formula $\varphi$ in DBI normal form, a formula $\theta$ is independent of $\varphi$ if the sequences of modal operators in the formula $\theta$ have a corresponding sequence of events in $U^\varphi$ and the precondition of those events is always $\top$. A formula is then dependent in case at least one of the preconditions for those events is different from $\top$. The main idea is that the pointed update $U_\varphi$ will keep the formulas that are independent of $\varphi$ true after the update.

**Example 94.** *For formula $\varphi = B_p\big(B_a q \wedge B_o(q \wedge B_a q)\big)$ from Example 78, we have that $\theta = \neg B_p B_p q \vee B_a B_o q$ is independent from $\varphi$ because the modality sequences of $\mathcal{B}^\theta$ $\epsilon$, $B_p$, $B_p B_p$, $B_a$, and $B_a B_o$ correspond to event sequences $0$, $0Q^\varphi_p 4$, $0Q^\varphi_p 4 Q^\varphi_p 4$, $0Q^\varphi_a {-}1$, and $0Q^\varphi_a {-}1 Q^\varphi_o {-}1$, and the preconditions for events $0$, $4$, and $-1$ are all $\top$. On the other hand, $\eta = \neg B_p B_a q$ is dependent on $\varphi$, because the modality sequence of $\mathcal{B}^\eta$ $B_p B_a$ corresponds to event sequence $0Q^\varphi_p 4 Q^\varphi_a 3$, and $\mathsf{pre}^\varphi(3) = q \neq \top$. Both $\theta$ and $\eta$ were true in $\mathcal{M}, v$ according to Figure 4.1, left. While the pointed update with $(U_\varphi, 0)$ keeps $\theta$ true, formula $\eta$ becomes false in $\mathcal{M} \odot U_\varphi, (v, 0)$ as shown in Figure 4.8, right.*

**Theorem 95** (Update synthesis minimality)**.** *If formula $\theta$ is independent of a DBI normal formula $\varphi$, then for any pointed Kripke model $(\mathcal{M}, v)$,*

$$\mathcal{M}, v \vDash \theta \qquad \Longleftrightarrow \qquad \mathcal{M} \odot U_\varphi, (v, 0) \vDash \theta. \tag{4.5}$$

*Proof.* Let $\mathcal{M} = \langle S, R, V \rangle$. We prove by induction on the construction of $\theta$ that, for any $\alpha \in E^\varphi$ and any $u \in S$, if $\theta$ is in $\top$-shape w.r.t. $\alpha$, then $\mathcal{M}, u \vDash \theta$ iff $\mathcal{M} \odot U_\varphi, (u, \alpha) \vDash \theta$. (4.5) is an instance of this induction statement for $\alpha = 0$ and $u = v$. Note that if $\theta$ is in $\top$-shape w.r.t. $\alpha$, then $\mathsf{pre}^\varphi(\alpha) = \top$ because of the empty sequence $\epsilon$, hence, the pointed update of $(\mathcal{M}, u)$ with $(U_\varphi, \alpha)$ is defined for all $u \in S$. For propositional atoms, the statement follows from the definition of pointed updates. The cases for Boolean connectives are straightforward. It remains to show the induction statement for $\theta = B_i \eta$. Let $\beta \in E^\varphi$ be the unique event such that $\alpha Q^\varphi_i \beta$. Then $\mathsf{pre}^\varphi(\beta) = \top$ because of sequence $B_i$ of $\theta$ and, additionally, $\eta$ is in $\top$-shape w.r.t. $\beta$. By IH, for any $u \in S$,

[6] The uniqueness of this sequence is given by the construction of the action model in Definition 87: intuitively, the 0 and $-1$ events are fixed, and for each (nesting of) belief operator(s) there is a corresponding event layered based on its modal depth.

we have $\mathcal{M}, u \vDash \eta$ iff $\mathcal{M} \odot U_\varphi, (u, \beta) \vDash \eta$. We have $\mathcal{M}, u \nvDash B_i\eta$ iff $\mathcal{M}, w \nvDash \eta$ for some $uR_iw$. By IH, this is equivalent to $\mathcal{M} \odot U_\varphi, (w, \beta) \nvDash \eta$, which is equivalent to $\mathcal{M} \odot U_\varphi, (u, \alpha) \nvDash B_i\eta$ because $(u, \alpha)R_i^\varphi(w, \gamma)$ iff $uR_iw$ and $\gamma = \beta$. □

**Corollary 96.** *Let $\varphi = \bigwedge_{j\in G} B_j\omega_j$ be a DBI normal formula, where $\varnothing \neq G \subseteq \mathcal{A}$.*

1. *If $i \notin \mathsf{ta}(\varphi)$, i.e., if $i \notin G$, then $i$'s beliefs are unaffected by the pointed update $U_\varphi$, i.e., $\mathcal{M}, v \vDash B_i\sigma$ iff $\mathcal{M} \odot U_\varphi, (v, 0) \vDash B_i\sigma$ for any formula $\sigma$ and any pointed Kripke model $(\mathcal{M}, v)$.*
2. *If $i \in \mathsf{ta}(\varphi)$, but $\omega_i = \bigwedge_{j\in H} B_j\pi_j$, i.e., $\omega_i$ is a higher order belief formula, then $i$'s propositional beliefs are unaffected by $U_\varphi$, i.e., $\mathcal{M}, v \vDash B_i\chi$ iff $\mathcal{M} \odot U_\varphi, (v, 0) \vDash B_i\chi$ for any propositional formula $\chi$ and any pointed Kripke model $(\mathcal{M}, v)$.*

**Theorem 97** (Update synthesis consistency preservation)**.** *Let $\varphi$ be a DBI normal formula. A pointed update with $U_\varphi$ can only cause (higher-order) inconsistent beliefs if the respective (higher-order) beliefs originally excluded the respective propositional preconditions: for any modality sequence $B_{i_1} \dots B_{i_k}$ with $i_j \neq i_{j+1}$ for any $j$ and any pointed Kripke model $(\mathcal{M}, v)$,*

$$\mathcal{M}, v \vDash \widehat{B}_{i_1}\left(\mathsf{pre}^\varphi(\alpha_1) \wedge \widehat{B}_{i_2}\left(\mathsf{pre}^\varphi(\alpha_2) \wedge \dots \widehat{B}_{i_k}\mathsf{pre}^\varphi(\alpha_k)\right)\right) \iff \mathcal{M} \odot U_\varphi, (v, 0) \nvDash B_{i_1} \dots B_{i_k}\bot \quad (4.6)$$

*for the unique sequence $0Q^\varphi_{i_1}\alpha_1Q^\varphi_{i_2}\alpha_2 \dots Q^\varphi_{i_k}\alpha_k$ of events $\alpha_j \in E^\varphi$.*

*Proof.* Let $\mathcal{M} = \langle S, R, V\rangle$ and $\mathcal{M} \odot U_\varphi = \langle S^\varphi, E^\varphi, V^\varphi\rangle$ for the pointed update of $(\mathcal{M}, v)$ with $(U_\varphi, 0)$. The left statement holds iff there is a sequence $vR_{i_1}s_1R_{i_2}s_2 \dots R_{i_k}s_k$ of states from $S$ such that $\mathcal{M}, s_j \vDash \mathsf{pre}^\varphi(\alpha_j)$ for $j = 1, \dots, k$. It is easy to observe (by induction on $k$) that this is equivalent to $(s_j, \alpha_j) \in E^\varphi$ for $j = 1, \dots, k$ and $(v, 0)R^\varphi_{i_1}(s_1, \alpha_1)R^\varphi_{i_2}(s_2, \alpha_2) \dots R^\varphi_{i_k}(s_k, \alpha_k)$. The equivalence to the right statement of Equation (4.6) now follows from the fact that $\mathcal{M} \odot U_\varphi, (s_k, \alpha_k) \nvDash \bot$ and the uniqueness of the sequence of $\alpha_j$'s such that $0Q^\varphi_{i_1}\alpha_1Q^\varphi_{i_2}\alpha_2 \dots Q^\varphi_{i_k}\alpha_k$. □

**Corollary 98.** *Let $\varphi$ be a DBI normal formula.*

1. *If $\varphi$ fits either of the clauses of Corollary 96 for agent $i$, then $U_\varphi$ preserves $i$'s consistency, i.e., $\mathcal{M}, v \nvDash B_i\bot$ iff $\mathcal{M} \odot U_\varphi, (v, 0) \nvDash B_i\bot$ for any $(\mathcal{M}, v)$.*
2. *If $\varphi = B_i\left(\xi \wedge \bigwedge_{k\in H} B_k\pi_k\right) \wedge \bigwedge_{j\in G} B_j\omega_j$ where $i \notin G$ and formula $\xi$ is propositional, then $U_\varphi$ makes $i$'s beliefs inconsistent if and only if $i$ originally does not consider $\xi$ to be possible, i.e., $\mathcal{M}, v \vDash \widehat{B}_i\xi$ iff $\mathcal{M} \odot U_\varphi, (v, 0) \nvDash B_i\bot$ for any $(\mathcal{M}, v)$.*

With the above corollary, we proved in which sense our synthesized updates have minimal side effects w.r.t. (higher order) beliefs of other agents in the sense introduced at the beginning of this chapter: if an agent is not involved at the first level of modal depth of a goal formula, then its beliefs are unchanged. At the same time, if an agent is involved in the goal formula but no

propositional component is (for that agent), then that agent's propositional beliefs are unchanged as well.

We use the notion of idempotence (Definition 27) to show that iterated pointed updates do not lead to a growth of the model[7]:

**Theorem 99.** *For any DBI normal formula $\varphi$ and any pointed Kripke model $(\mathcal{M}, v)$, the pointed action model $U_\varphi$ is idempotent.*

*Proof.* The updates $\mathcal{M}\odot U, (v,0)$ and $(\mathcal{M}\odot U)\odot U, ((v,0),0)$ are always defined by Theorem 92.

Since $U_\varphi$ has an out-tree structure, we conduct the proof by induction over this structure beginning from the root 0.
**Ind. hyp.:** For every state $((w,\alpha),\beta)$ in $(\mathcal{M}\odot U)\odot U$, $\beta = \alpha$.
**Base case:** Point $(v,0)$ is the only state that can be combined with 0, since by Definition 87, 0 has no incoming edges in $U_\varphi$.
**Ind. step:** Suppose the IH holds for state $((w,\alpha),\alpha)$, but not for its $i$-accessible state $((x,\beta),\gamma)$, meaning $\gamma \neq \beta$. Note that this $i$-accessible state must exist, since $(\mathcal{M}\odot U)\odot U, ((v,0),0)$ could not be smaller than $\mathcal{M}\odot U, (v,0)$, as all preconditions in $U_\varphi$ are propositional by Definition 87. Since $((w,\alpha),\alpha)R_i^{U\,U}((x,\beta),\gamma)$ it must be the case that also $(w,\alpha)R_i^U(x,\beta)$ by Definition 81. Since $\varphi$ is DBI normal, every state in $U_\varphi$ has no more than one $j$-accessible state, hence $\beta$ is the only $i$-accessible state from $\alpha$. Therefore if $((w,\alpha),\alpha)R_i^{U\,U}((x,\beta),\gamma)$ it must be that $\gamma = \beta$. □

## 4.3 The role of privatization

As mentioned at the beginning of this chapter, the minimality of our update synthesis method relies on a particular structure of the action model used for the update, which we call *privatized*. In this section, we give an explicit definition and show why it serves the purpose of preserving beliefs whenever possible.

We begin with the notion of a modal syntactic tree, which represents the nesting of modalities within a formula as a tree. As we will see soon, this is exactly the structure that provides the stratification and consequently the privatization. Note that the structure of the action model from Definition 87 (almost) exactly follows this tree structure already, so there was no need to introduce it explicitly already there.

**Definition 100** (Modal syntactic tree)**.** *The* modal syntactic tree $\mathcal{T}_\varphi$ *of a goal formula $\varphi$ is an out-tree with a single unlabeled root, while all non-root nodes are labeled with modal operators.*

Some examples of modal syntactic trees are provided in Figure 4.11: In $\mathcal{T}_{B_i\xi}$, the root has one child-leaf labeled $B_i$. Both $\mathcal{T}_{B_i\varphi}$ and $\mathcal{T}_{B_i(\xi\wedge\varphi)}$ are obtained by labeling the root of $\mathcal{T}_\varphi$ with $B_i$ and making it the only child of the new root. Finally, $\mathcal{T}_{\varphi\wedge\psi}$ is obtained by taking the disjoint union of $\mathcal{T}_\varphi$ and $\mathcal{T}_\psi$ and identifying their roots.

[7] A proof of a similar statement can be found in [LPW11]

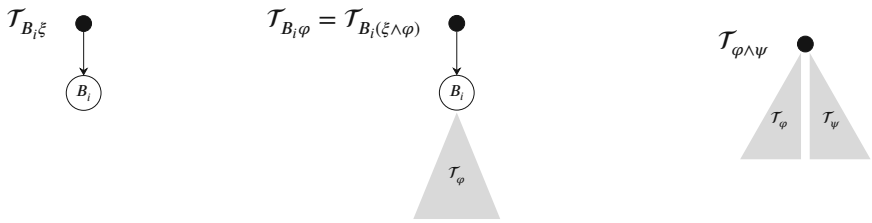



**Figure 4.11:** Modal syntactic trees

To formally describe this privatized structure, we introduce some preparatory definitions:

**Definition 101** (Modal-syntactic-tree root paths)**.** *For formula $\varphi$ we define $RootP(\varphi)$ as the set of paths $\big((root, \alpha_1, i_1), \dots, (\alpha_{l-1}, \alpha_l, i_l)\big)$ of length $l \geq 0$ in $\mathcal{T}_\varphi$ starting from the root, where $i_k$ is the label of $\alpha_k$ for $k = 1, \dots, l$.*

**Definition 102** (Kripke-frame root walks)**.** *For a pointed Kripke frame $(\langle S, R\rangle, w)$ we similarly define the set of root walks $RootW(\langle S, R\rangle, w)$ as the set of all walks starting from the root (point) $w$. We further define the restriction $RootW_{\text{nsr}}(\langle S, R\rangle, w)$ as the subset of $RootW(\langle S, R\rangle, w)$, where we exclude walks that contain at least two successive edges for the same agent.*

**Definition 103** (Agent sequence)**.** *For a walk $\sigma = \big((\alpha_1, \alpha_2, i_1), \dots, (\alpha_l, \alpha_{l+1}, i_l)\big)$ from Definitions 101 or 102 we define $AgSeq(\sigma) := (i_1, \dots, i_l)$. In particular, $AgSeq(\varepsilon) := \varepsilon$. We extend this definition to sets, where for a set of walks $\Sigma$, $AgSeq(\Sigma)$ is the set of corresponding agent sequences.*

*The set of all agent sequences of length $l$ we denote by $\mathcal{A}^l$. The set of all agent sequences of length $l$ without any successively repeating agents we denote by $\mathcal{A}^l_{\text{nsr}}$.*

We now define agent-clusters and agent-accessible parts of the model.

**Definition 104** (Clusters and reachable states)**.** *For a Kripke frame $\mathcal{F} = \langle W, R\rangle$, world $w \in W$, and sequence $(i_1, \dots, i_l) \in \mathcal{A}^l$ of agents, we introduce the* cluster of path-accessible worlds

$$C^{i_1,i_2,\dots,i_l}_{\mathcal{F},w} := \big\{u \in W \mid (\exists u_2, \dots, u_l \in W)\ wR_{i_1}u_2R_{i_2}\dots u_lR_{i_l}u\big\}. \tag{4.7}$$

*In particular, for the empty sequence, $C^{\varepsilon}_{\mathcal{F},w} = \{w\}$.*

Finally, we have the necessary tools to formally describe the notion of privatization.

**Definition 105** (Privatized)**.** *We call a pointed Kripke frame $(\mathcal{F}, w) = (\langle S, R\rangle, w)$* privatized *w.r.t. DBI formula $\varphi$ iff for any root path $\sigma \in RootP(\varphi)$*

$$C^{AgSeq(\sigma)}_{\mathcal{F},w} \neq \varnothing \text{ and } \left(\forall agSeq \in \bigcup_{l=0}^{\infty} \mathcal{A}^l_{\text{nsr}} \setminus \{AgSeq(\sigma)\}\right) \left(C^{AgSeq(\sigma)}_{\mathcal{F},w} \cap C^{agSeq}_{\mathcal{F},w}\right) = \varnothing. \tag{4.8}$$

**Definition 106** (Weakly Privatized)**.** *Similarly we call a pointed Kripke frame* $(\mathcal{F}, w) = (\langle S, R\rangle, w)$ weakly privatized *w.r.t. DBI formula* $\varphi$ *iff for any root path* $\sigma \in Root P(\varphi)$

$$\left(\forall agSeq \in \bigcup_{l=0}^{\infty} \mathcal{A}^{l}_{\text{nsr}} \setminus \{AgSeq(\sigma)\}\right) \left(C^{AgSeq(\sigma)}_{\mathcal{F},w} \cap C^{agSeq}_{\mathcal{F},w}\right) = \varnothing. \tag{4.9}$$

**Definition 107** (Walk accessibility)**.** *For a pointed Kripke frame* $(\mathcal{F}, w) = (\langle W, R\rangle, w)$ *we define* walk accessibility operator

$$W Acc_{\text{nsr}}(u, \mathcal{F}, w) := \left\{\sigma \in Root W_{\text{nsr}}(\mathcal{F}, w) \mid \pi_2 \pi_{|\sigma|}\sigma = u\right\} \tag{4.10}$$

*to be the set of agent-alternating walks starting from* $w$ *and ending in* $u$*.*

Walk accessibility yields an alternative equivalent definition of privatized frames:

**Corollary 108** (Privatized alternative)**.** *A pointed Kripke frame* $(\mathcal{F}, w)$ *is privatized w.r.t. DBI* $\varphi$ *iff for any root path* $\sigma \in Root P(\varphi)$

$$C^{AgSeq(\sigma)}_{\mathcal{F},w} \neq \varnothing \text{ and } \left(\forall s \in C^{AgSeq(\sigma)}_{\mathcal{F},w}\right) \left|AgSeq\big(W Acc_{\text{nsr}}(s, \mathcal{F}, w)\big)\right| = 1. \tag{4.11}$$

*Proof.* Follows from Definitions 103, 105 and 107. □

**Corollary 109** (Weakly privatized alternative)**.** *A pointed Kripke frame* $(\mathcal{F}, w)$ *is privatized w.r.t. DBI* $\varphi$ *iff for any root path* $\sigma \in Root P(\varphi)$

$$\left(\forall s \in C^{AgSeq(\sigma)}_{\mathcal{F},w}\right) \left|AgSeq\big(W Acc_{\text{nsr}}(s, \mathcal{F}, w)\big)\right| = 1. \tag{4.12}$$

*Proof.* Follows from Definitions 103, 106 and 107. □

**Theorem 110.** *For DBI formula* $\varphi$*, the pointed action model* $(U_\varphi, 0)$ *is privatized w.r.t.* $\varphi$*.*

*Proof.* By induction on the recursive construction of $U_\varphi$ as by Definition 87.
**Base case:** For formula $B_i\xi$, where $\xi$ is propositional, since $AgSeq(Root P(B_i\xi)) = \{\varepsilon, (i)\}$, we only need to check all states that are walk-accessible via these two sequences. Since in $U_{B_i\xi}$ the state $m \notin \{0, -1\}$ is the only state accessible via an $i$-edge from the root 0, as the 0 itself has no incoming arrows, $AgSeq(W Acc_{\text{nsr}}(m, U_{B_i\xi}, 0)) = \{i\}$. Furthermore, the root 0 is only walk-accessible via the empty sequence $\varepsilon$, therefore $AgSeq(W Acc_{\text{nsr}}(0, U_{B_i\xi}, 0)) = \{\varepsilon\}$. Lastly, as the sink $-1$ is not walk-accessible via any of the two sequences $\varepsilon$ and $(i)$ in $AgSeq(Root P(B_i\xi))$, we conclude that $(U_{B_i\xi}, 0)$ is indeed privatized w.r.t. $B_i\xi$.
**Ind. step:** If $\varphi = B_i\psi$ for DBI normal formula $\psi$, then $(U_\psi, 0)$ is privatized w.r.t. $\psi$. By the $Root P(\varphi)$ extends the paths in $Root P(\psi)$ right after the root with an edge to $m$: $Root P(\varphi) = \{((root, m, i), (m, \alpha_1, j_1)...(\alpha_{l-1}, \alpha_l, j_l)) \mid ((root, \alpha_1, j_1), ...(\alpha_{l-1}, \alpha_l, j_l)) \in Root P(\psi) \text{ and } l \geq 0\} \cup \{\varepsilon\}$. Since the set of walks in $Root W_{\text{nsr}}(U_\psi, 0)$ is extended in exactly the same way, $(U_\varphi, 0)$ must also be privatized w.r.t. $\varphi$.

If $\varphi = B_i(\xi \wedge \psi)$ for a propositional formula $\xi$ and a DBI normal formula $\psi$, the reasoning is analogous to the previous case, as privatization is a frame property and thus independent of the additional precondition added.

If $\varphi = \psi \wedge \theta$, for DBI normal formulas $\psi$ and $\theta$, then $RootP(\varphi) = RootP(\psi) \cup RootP(\theta)$. Again since by Definition 87 $RootW(U_\varphi) = RootW(U_\psi) \cup RootW(U_\theta)$ is extended the same way, $(U_\varphi, 0)$ is again privatized w.r.t. $\varphi$ and we are done. Note that in $U_\varphi$ not only the root 0, but also the sink nodes $-1$ of $U_\psi$ and $U_\theta$ are unified. □

We will sometimes slightly abuse the notation by applying properties defined on Kripke frames to Kripke models $\langle S, R, V\rangle$ and action models $\langle E, Q, \mathsf{pre}\rangle$. In all such cases, we mean the property to be applied to frames $\langle S, R\rangle$ and $\langle E, Q\rangle$ respectively.

**Lemma 111.** *Let the pointed update $\big(\mathcal{M} \odot U, (w, \alpha)\big)$ of a pointed Kripke model $(\mathcal{M}, w)$ with a pointed action model $(U, \alpha)$ be defined. For any $agSeq \in \bigcup_{l=0}^{\infty} \mathcal{A}^l$,*

$$(x, \beta) \in C^{agSeq}_{\mathcal{M}\odot U,(w,\alpha)} \quad \Longrightarrow \quad \beta \in C^{agSeq}_{U,\alpha} \text{ and } x \in C^{agSeq}_{\mathcal{M},w}. \tag{4.13}$$

*Proof.* Let $\mathcal{M} = \langle S, R, V\rangle$, $U = \langle E, Q, pre\rangle$, and $\mathcal{M} \odot U = \langle S', R', V'\rangle$. We use induction on $l = |agSeq|$.
**Base case** $l = 0$**:** Since $\mathcal{A}^0 = \{\epsilon\}$ the statement follows trivially from Definitions 104 and 81.
**Ind. step:** Consider any agent sequence $\overline{agSeq} = agSeq \circ i$ of length $l + 1$ and state $(x, \beta) \in C^{\overline{agSeq}}_{\mathcal{M}\odot U,(w,\alpha)}$. By Definition 104, there must exist a state $(v, \gamma) \in C^{agSeq}_{\mathcal{M}\odot U,(w,\alpha)}$ such that $(v, \gamma) R'_i (x, \beta)$, in particular, $v R_i x$ and $\gamma Q_i \beta$ by Definition 81. By IH for $agSeq$ of length $l$, we have $v \in C^{agSeq}_{\mathcal{M},w}$ and $\gamma \in C^{agSeq}_{U,\alpha}$. Thus, by Definition 104 both $x \in C^{\overline{agSeq}}_{\mathcal{M},w}$ and $\beta \in C^{\overline{agSeq}}_{U,\alpha}$. □

**Theorem 112.** *For pointed action model $(U, \alpha)$ privatized w.r.t. DBI formula $\varphi$ and pointed epistemic Kripke model $(\mathcal{M}, w)$, if their pointed update exists, then $(\mathcal{M} \odot U, (w, \alpha))$ is weakly privatized w.r.t. $\varphi$.*

*Proof.* Suppose by contradiction that the pointed action model $(U, \alpha)$ is privatized w.r.t. $\varphi$, but for some pointed Kripke model $(\mathcal{M}, v)$, the update $(\mathcal{M} \odot U, (w, \alpha))$ exists, however is not weakly privatized w.r.t. $\varphi$. This means that there exists a root path $\sigma \in RootP(\varphi)$, an agent sequence $agseq \in \bigcup_{l=0}^{\infty} \mathcal{A}^l_{nsr} \setminus AgSeq(\sigma)$ and states $(x, \beta) \in C^{agseq}_{\mathcal{M}\odot U,(w,\alpha)}$, $(x, \beta) \in C^{AgSeq(\sigma)}_{\mathcal{M}\odot U,(w,\alpha)}$. By Lemma 111, this implies that $\beta \in C^{AgSeq(\sigma)}_{U,\alpha}$ and $\beta \in C^{agSeq}_{U,\alpha}$. However this contradicts the assumption that $(U, \alpha)$ is privatizing w.r.t. $\varphi$, as $C^{AgSeq(\sigma)}_{U,\alpha} \cap C^{agSeq}_{U,\alpha} = \varnothing$ □

**Theorem 113.** *For DBI formula $\varphi$ and any pointed Kripke model $(\mathcal{M}, v)$, the pointed update between pointed action model $(U_\varphi, 0)$ and $(\mathcal{M}, v)$ is defined and $(\mathcal{M} \odot U, (w, \alpha))$ is weakly privatized.*

*Proof.* Follows immediately from Theorem 112. □

## 4.4 Conclusions and future work

In this chapter, we presented a way to synthesize a pointed action model for DBI goal formulas that: (i) makes the goal formula true in the resulting model; (ii) privatizes any Kripke model to which it is applied to (thus breaking the common knowledge of the model assumption and simulating totally private communication); (iii) preserves consistency whenever possible (marking a difference from, e.g., [DDvdHKK20]); and (iv) has minimal side effects (in the sense that it changes as few beliefs unspecified in the goal formula as possible).

In addition, the synthesized pointed action models are combined with pointed Kripke models via a novel pointed update operation, which does not apply to the whole model globally, but rather respects the structure of the privatization introduced in the synthesis. This mechanisms can simulate global a priori updates, i.e., the intervention of a system designer aiming at modifying the (a priori) beliefs of agents in a system as a reaction to unexpected behavior. Since the pointed update is a subset of the full product update operation, the proposed update mechanism is quite efficient: only the initial update increases the size of the model at worst linearly in the size of the goal formula, while all subsequent updates via goal formulas with the same belief structure (or substructure thereof) only decrease the model size. This compares favorably with the exponential blow up after iterated updates using the standard product updates, which is a topic also relevant in epistemic planning. In particular, our pointed update operation has a clear advantage over the bisimulation contraction operation[BdRV01]: in the latter case, the resulting model first needs to be produced and then it can be contracted; while the resulting model is smaller, the operation is still computationally costly. Our pointed update operation takes care of (some) bisimilar worlds on the fly, without going through the model again.

The main difference of our framework w.r.t. plausibility models [BS16] is that we do not fix an order of preference in the worlds in advance. In other words, we do not make any assumption on which situations are more likely than others, as it is against the open-world assumption of a priori updates. At the same time, we do not list the set of possible actions in advance as in epistemic planning, because it would take away the a priori flavor of our updates. How the system designer comes up with the goal formula is a distinct problem that we do not address here.

Existing synthesis methods typically work in a language extended with quantifiers over updates, such as the Arbitrary Action Modal Logic [Hal13] and Arbitrary Arrow Update Modal Logic [DDvdHKK20] (see Section 1.2.1). In particular, the quantification over updates allows to express the synthesis operation within the logical language. However, these update operations do not focus on issues of privatization and of minimal change, which is a strong shortcoming when considering applications to distributed systems, especially when aiming at modeling a priori updates. Recall that the goal of the synthesis tasks is, given a goal formula $\varphi$, to output an update model that, applied to any initial model, outputs a model where $\varphi$ holds. For some models however, $\varphi$ could hold vacuously, in case e.g., there are no relations in the resulting model for a given agent involved in the formula. In our case, this is not possible: because of the proposed stratification of the model, an agent does not result in believing inconsistencies unless precisely instructed so by a sequence of updates. In other words, our synthesis method addresses belief consistency preservation, while other existing methods do not. While it is possible to construct

AML or GAUL update models representing completely private communication [BR16] (see, e.g., Figure 4.3), and thus it is possible to synthesize them, there is no standardized update synthesis procedure for it. Finally, AAUL was proved to be undecidable [vvK17]. On the other hand, our language is a restriction of the standard doxastic language, and is hence decidable. Consequently, while not as powerful, it is computationally more appealing for the synthesis task.

One natural extension of the current framework is to allow a wider range of goal formulas, incorporating the non-determinism of the $\hat{B}_i$ operator, dual to modality $B_i$, and in disjunctions of $B_i$ operators, as well as enabling the formalization of the conditions of consistency for each level of iterated beliefs and outlining the exact set of beliefs unaffected by the update based on a given goal formula. We also plan to provide a mechanism to represent public announcements and other kinds of updates in our framework, providing full granularity in the update design.

CHAPTER 5

# Agent-based Local A priori Belief Updates

In this chapter[1], we introduce an update mechanism for local a priori updates, which are updates initiated autonomously by designer agents in SASO systems. As stressed several times already, the epistemic analysis of distributed systems [FHMV95] and of epistemic puzzles [vDK15] routinely relies on agents' common knowledge of the model [Art20]. In effect, this provides the agents' common *a priori* assumptions and enables them to reason about (higher-order) reasoning of other agents. This type of knowledge is very different from the a posteriori knowledge acquired by agents through communication, and is usually static. Accordingly, dynamic epistemic logic implements knowledge change by model modifications that reorganize and shrink the already available possibilities, in contrast to the initial epistemic model creating the common space of these possibilities for the agents out of the puzzle description or distributed system specification.

Indeed, the system designer's task of creating a distributed system satisfying a given specification can be viewed as creating the adequate common a priori knowledge for the agents (Section 2.1.4). As argued in Chapter 2, design mistakes discovered in the simulation/execution phase trigger a bootstrapping cycle, where the system assumptions are re-considered based on observed errors. The new assumptions are then implemented in the updated system, in an operation that can be viewed as a priori belief updates: In the case of traditional distributed systems, aberrant a posteriori behavior prompts the system designer to trigger a new iteration of the design cycle, where she will re-adapt the a priori system assumptions so as to match the intended system behavior, redesign the affected parts of the implementation and finally deploy and restart them, initializing the agents with updated a priori assumptions. A model update mechanism for such global a priori updates has been described in Chapter 4.

Given the trend towards more and more complex and growing distributed systems, however, discovering and recovering from such design errors is increasingly becoming prohibitively costly:

[1] This chapter is based on joint work with Roman Kuznets [CK24].

the ability to predict and/or monitor possible behaviors of such a system decreases exponentially, whereas the redesign costs increase dramatically. SASO [BG09, THvM+14] systems, on the other hand, allow processes to access and operate with their own representation of the system, which in turn also enables them to update certain design assumptions on their own. In other words, in addition to the conventional a posteriori belief updates, agents in SASO systems are allowed to also perform a priori belief updates, with the goal of self-correcting their behavior in response to situations not envisioned by the system designer.

This chapter is devoted to an epistemic formalization of a mechanism for local and agent-based a priori belief updates, implemented in the form of DEL-inspired updates. With *local* and *agent-based*, we mean that these updates are initiated by the agent(s) autonomously and independently when discovering some design error. As such, they are applied only to the portion of the model accessible to that agent, without involving anybody else. Agents mainly discover inconsistencies at the system level by reaching an inconsistent state of beliefs: as such, we focus on the self-recovery of these agents in an attempt to solve the task despite adverse circumstances. Following Plaza [Pla89], we illustrate our methods using (variants of) standard epistemic puzzles such as the consecutive numbers and muddy children puzzle introduced in Section 1.2.4.[2]

The goal of this chapter is to describe a mechanism that applies a given a priori update to a certain model. The question of coming up (or synthesizing) a local a priori update is outside of the scope of this thesis. For this reason, we do not impose any restriction on the kinds of models used in the updates, as we want to describe the mechanism in the most general way possible. In Section 5.5 we consider some heuristics to bound the a priori reasoning process, but we leave their formalization for future work.

Let us first illustrate how faulty a priori assumptions can derail the progress, say, in the consecutive numbers puzzle and how human agents might still be able to find a solution by adjusting their a priori beliefs.

**Example 114** (Consecutive numbers)**.** *Recall the consecutive numbers puzzle exposed in Section 1.2.4: Two agents a and b are privately told two natural numbers. In addition, they are publicly told that the two numbers are consecutive (making it common knowledge). Suppose that a is told number 1 and b is told number 2. They are allowed to state whether they know the other's number or not, but not allowed to communicate their own number. In the standard formulation, the puzzle is solved by the following dialog:*

- *a: I don't know your number.*
- *b: I don't know your number either.*
- *a: Now I know your number.*
- *b: Now I know yours too.*

[2]We focus on epistemic puzzles instead of examples from simple distributed systems because they are more intuitive, and because of their popularity in the epistemic analysis of distributed systems [HM90, FHMV95].

*Here the first statement by a is uninformative. The first statement by b makes it clear that b's number is not 0, which enables a to conclude that b's number must be 2. Since this determination would not have happened were a to hold number 3, now b can conclude that a's number is 1. The standard epistemic modeling of this example involves a Kripke model that each agent is supposed to build based on the rules of the puzzle in a way that makes this model commonly known to both agents. This commonality is based, in Lewis's telling, on the "suitable ancillary premises regarding [agents'] rationality, inductive standards, and background information" ([Lew69, p. 53]). While agents in epistemic puzzles are routinely considered to be perfect reasoners, which takes care of rationality and inductive standards, the question of background information is much less clear cut.*

*For our twist on the original formulation, imagine that, unknown to each other, a and b learned different definitions of natural numbers: a starts them from 1, while for b number 0 is also natural. In other words, their a priori beliefs are not shared by the agents preventing them from achieving common knowledge. Since natural numbers are routinely assumed (including in the formulation of the consecutive numbers puzzle) to be a well-defined object, each agent incorrectly believes that the other agent shares their definition of natural numbers. As is to be expected, the common knowledge of the model and of the situation at hand shatters, leading to the following possible conversation:*

- *a: I know your number.*
- *b: Wait, what? But that is impossible, unless... Ah, I see. Then I know your number too.*

*Here agent a does not consider* $(1,0)$ *to be a legitimate pair, hence, (correctly) concludes that the numbers are* $(1,2)$*. Agent b, on the other hand, expects a to consider* $(1,0)$ *and, hence, does not understand a's reasoning. Indeed, according to b, if a had 1, he would have hesitated between* $(1,0)$ *and* $(1,2)$*, while if a had 3, he would have hesitated between* $(3,2)$ *and* $(3,4)$*. Thus, a's statement is incompatible with b's view of the world. In the proposed conversation, b does what is natural for a human reasoner: she questions her a priori assumptions, conceives that there is an alternative understanding of natural numbers as starting from 1, realizes that this is compatible with a's behavior, and updates her a priori beliefs (about a's a priori beliefs).*[3] *Moreover, after this update, a's claim to knowing b's number is only compatible with a having 1: were a's number 3, the hesitation between* $(3,2)$ *and* $(3,4)$ *would have remained. This update of b's a priori beliefs enables her to both explain the situation and arrive at the correct conclusion.*

Clearly, standard (dynamic) epistemic reasoning would not provide an adequate explanation. Epistemically, *b*'s beliefs are supposed to become inconsistent. In fault-tolerant systems, this

[3]Indeed, there is an unbounded number of explanations that might be considered possible by *b*. Here we focus on a particular instance because our aim is to show what kind of updates we are interested in. The problem of coming up with the right update is outside the scope of this thesis. To partially address this issue, we suggest an iterative method that allows an agent to try multiple explanations and to change them whenever they are not fit. Finally, if we were to model this example with plausibility models [BS16] we would have to provide agents some a priori knowledge about the alternative possible explanations, that is something we do not assume.

often translates in *b* considering herself and/or the other agent fully byzantine and thus completely unreliable [KPSF19a], making the puzzle not solvable. The goal of the present chapter is to model a priori belief updates epistemically, thus, providing agents with self-correction capabilities. While the process remains highly non-deterministic in general, as there are multiple possible ways to resolve design mistakes, we provide some general guidelines and heuristics to guide possible future implementations of such self-recovery operations in SASO systems.

## 5.1 Formal preliminaries

In this section we introduce the necessary tools for formally describing a priori updates. They constitute a refinement of the standard definitions already provided in Section 1.2.4.

**Definition 115** (Binary relation types)**.** *Recall from Section 1.2.1, that a binary relation $R \subseteq W \times W$ is called* reflexive *iff $wRw$ for all $w \in W$;* transitive *iff for all $w, v, u \in W$ we have $wRu$ whenever $wRv$ and $vRu$;* euclidean *iff for all $w, v, u \in W$ we have $vRu$ whenever $wRv$ and $wRu$.*

*In addition, a binary relation is* symmetric *iff for all $w, v \in W$ we have $vRw$ whenever $wRv$. Relation $R$ is called an* equivalence relation *iff it is reflexive, transitive, and euclidean; a* partial equivalence relation *iff it is transitive and symmetric; an* introspective relation *iff it is transitive and euclidean.*

**Proposition 116.** *An equivalence relation is also symmetric, hence, introspective and a partial equivalence relation. A partial equivalence relation is also euclidean, hence, introspective.*

**Definition 117** (Model types)**.** *We call a Kripke model $\langle W, R, V \rangle$* epistemic *iff all $R_a$ are equivalence relations;* introspective *iff all $R_a$ are introspective;* quasi epistemic *iff all $R_a$ are partial equivalence relations.*

**Proposition 118.** *An equivalence relation $R_a \subseteq W \times W$ partitions $W$ into equivalence classes, or $a$-*clusters*, such that for each equivalence class $E \subseteq W$ we have $uR_av$ for any $u, v \in E$ and neither $uR_au'$ nor $u'R_au$ for any $u \in E$ and $u' \in W \setminus E$. A partial equivalence relation produces a similar partition but of a subset $W \setminus I$, where $I \subseteq W$ consists of isolated worlds, i.e., neither $iR_aw$ nor $wR_ai$ for any $w \in W$ and $i \in I$.*

**Definition 119** (Composition and iteration of relations)**.** *For any binary relations $Q, Q' \subseteq W \times W$ on a set $W$, their* composition $Q \circ Q' := \{(w, v) \in W \times W \mid (\exists u \in W)(wQu \text{ and } uQ'v\}$. *Let $Q^k$ for $k \geq 0$ be defined recursively by $Q^0 := \{(w, w) \mid w \in W\}$ and $Q^{k+1} := Q \circ Q^k$.*

Analogously to Section 1.2.1, we define mutual and common accessibility relations but for beliefs:

**Definition 120** (Mutual and common accessibility)**.** *For a Kripke model $\langle W, R, V \rangle$, we define the*

- mutual accessibility relation $R_{\mathcal{A}} := \bigcup_{a \in \mathcal{A}} R_a$ *that corresponds to the mutual belief of all agents;*

- common accessibility relation $R^*_{\mathcal{A}} := \bigcup_{k=1}^{\infty} R^k_{\mathcal{A}}$ *that corresponds to the common belief of all agents.*

We now define the agent's submodel, which follows closely the definition of the submodel generated by an agent's accessibility relation from the actual world [BdRV01]:

**Definition 121** (Agent's submodel)**.** *Let* $(\mathcal{M}, v)$ *for* $\mathcal{M} = \langle W, R, V \rangle$ *be a pointed model and* $i \in \mathcal{A}$ *be an agent with consistent beliefs, i.e., such that* $R_i(v) \neq \emptyset$*. The* submodel accessible by $i$ at $(\mathcal{M}, v)$*, or* $i$'s part/submodel of $(\mathcal{M}, v)$ *is the Kripke model* $\mathcal{M}^i_v := \langle W', R', V' \rangle$ *such that*

- $W' \subseteq W$ *is the set of all worlds* $u$ *such that* $vR_iu_1R_{j_1}u_2R_{j_2}\dots u_kR_{j_k}u$ *for some* $k \geq 0$*, worlds* $u_1, \dots, u_k \in W$*, and agents* $j_1, \dots, j_k \in \mathcal{A}$*;*
- $R'_j := R_j \cap (W' \times W')$ *for each* $j \in \mathcal{A}$*;*
- $V'(p) := V(p) \cap W'$ *for each* $p \in Prop$*.*

*Note, in particular, that* $R_i(v) \subseteq W'$*.*

*If* $R_i(v) = \emptyset$*, we say that* $i$*'s submodel is empty.*

In the a priori update mechanism proposed in this chapter, we also employ some novel ideas that are a core of this thesis, namely:

- *Fallibilism*, which we represent in epistemic logic by considering pointed models $(\mathcal{M}, v)$, where the real world $v$ is not considered possible by any of the agents. This allows false beliefs without preventing agents to have an accurate view of the system e.g., by considering an exact duplicate of the actual world as a possibility (as already used in Chapter 4).
- We use the distinction between actually possible worlds and virtually possible worlds (Section 1.2.2), distinguishing which worlds are considered to be actual candidates for the real world from those that are considered possible for representing higher order reasoning.

## 5.2 A priori beliefs in epistemic logic

In Section 2.1.4, we established a connection between epistemic logic and a priori beliefs by equating the latter, for an agent, to the sets of worlds that that agent consider possible (both actually and virtually). Since this meta-level of beliefs concerns sets of possible worlds and not (directly) formulas, it is important to differentiate between a priori beliefs that can and cannot be represented syntactically. We call the former *explicit a priori beliefs* and define them as beliefs that can be represented by one epistemic formula in the object language. Thus, beliefs that can be represented by a finite set of formulas are explicit because the conjunction of the set provides an equivalent description. However, not all a priori beliefs are explicit. For instance, factivity of everyone's knowledge is usually described as either a frame property (reflexivity) or an infinite

set of formulas ($B_a\varphi \to \varphi$ for all $a$ and $\varphi$) but cannot be reduced to the truth of one formula only. We call such a priori beliefs *implicit*.[4,5]

In as far as epistemic puzzles and distributed systems deal with a priori beliefs (usually without calling them that), it is done by providing the initial Kripke models that agents proceed to modify based on the information they receive. The implicit form of a priori beliefs is dominant, and converting it into an explicit representation may well be an unrealistic task even for simple epistemic puzzles [Art22].

### 5.2.1 Privacy of a priori beliefs

In the typical modeling of epistemic puzzles, agents are assumed to have common and factual a priori beliefs, as represented by the commonly known epistemic model [Art20]. This ensures the homogeneity of reasoning by all agents and creates no problems as long as agents have no need to modify their a priori beliefs. At the same time, for our setting, it is unreasonable to assume that the internal reasoning process leading one agent to modify its explicit a priori beliefs would be noticeable by other agents, let alone be commonly known among them, even in cases where they start with commonly known priors.

Hence, we abandon the assumption of the common knowledge of the model and treat each agent as having its own private a priori beliefs represented by the submodel of this agent. If submodels of several agents overlap, this is treated as a coincidence, especially in view of the fact that each agent, being only aware of its own submodel, would be oblivious of any such overlaps.

Note that if the initial model is epistemic, the submodel of each agent is the whole model. We interpret this situation as each agent believing to have the common model that they have, but leave the possibility of one or several agents being wrong, typically as a result of later a priori public updates.

To summarize, (i) the agents' reasoning is represented by a pointed Kripke model $(\mathcal{M}, v)$, but instead of assuming common knowledge of $(\mathcal{M}, v)$, an agent $i$ is only guaranteed to know $i$'s submodel $\mathcal{M}^i_v$, which may or may not be different from submodels $\mathcal{M}^j_v$ visible to other agents $j$; and (ii) in addition to the standard public update mechanism for public announcements, we employ a *private* model update mechanism for a priori belief updates of an agent $i$ that results in $i$'s submodel becoming disjoint from other agents' submodels after $i$ performed an a priori update.

[4]While clearly axiom schemes cannot be reduced to a single (first-order) formula, they can be treated as second-order formulas. Interestingly, this suggests that some implicit a priori beliefs might be expressible in a second-order language, which is, however, outside of the scope of this thesis.

[5]Awareness logic [FH87] distinguishes between explicit beliefs (what an agent is explicitly aware of and believes) and implicit beliefs (what the agent could believe through inference from their explicit beliefs). Here the difference is rather about representability: a priori beliefs are explicit if they can be represented in the object language, and are implicit otherwise. In other words, our a priori beliefs are not implicit because agents are not logical omniscient, but because these model properties cannot be expressed in the object language.

### 5.2.2 A priori belief updates as conflict resolution

We still maintain the agents' reasoning within a pointed Kripke model, i.e., each agent $i$ is still logically omniscient w.r.t. the a posteriori information contained in $i$'s submodel $\mathcal{M}^i_v$ of the given pointed Kripke model $(\mathcal{M}, v)$. The observations that the agent $i$ is making during the run may come into a conflict with its a priori beliefs, which would manifest as $R_i(v) = \varnothing$ causing $i$'s beliefs to become inconsistent. As long as such a contradiction does not arise, $i$ continues a posteriori reasoning in the standard epistemic manner or DEL manner for received information. In light of the Knowledge of Preconditions Principle (Theorem 12), inconsistent beliefs make it impossible for the agent to act correctly. Indeed, even if the agent is supposed to choose between mutually exclusive actions $A$ if $\varphi$ holds or $B$ otherwise, the inconsistent agent would have to perform both due to believing everything, including both $\varphi$ an $\neg\varphi$. This provides a good incentive for the agent to reexamine its a priori beliefs and try updating them in such a way as to restore consistency. Semantically, this is achieved by $i$ creating a new submodel $\mathcal{M}^i_v$ (disjoint from the existing model) for itself in place of the current empty one. This is exactly the desired functionality for SASO systems: if an agent finds out during runtime that its a priori beliefs are inadequate, i.e., if an a posteriori epistemic update of an agent violates some of its a priori beliefs, the agent may initiate a priori reasoning, aiming at finding new a priori beliefs that comply with the a posteriori epistemic status. If an appropriate solution is found, an *a priori belief update* is privately invoked for installing a new private submodel within the current Kripke model.

Needless to say, there are many conceivable ways for resolving conflicts arising from inconsistencies in SASO systems, mainly because, in accordance with the Duhem–Quine thesis [Qui51], it is not always possible to isolate the specific hypothesis (a priori belief in our case) as the culprit for an inconsistency, even for a restricted set of explicit a priori beliefs. Thus, we will not try to provide deterministic algorithms for choosing an appropriate a priori update. At the same time, we do not leave this process completely *ad hoc*. A new submodel is constructed from several building blocks representing the agent's new guesses regarding the actually possible worlds, the virtually possible worlds, and their relationship, as described in Section 5.3.[6]

### 5.2.3 Higher-order a priori reasoning

The question of higher-order belief updates remains outside the scope of this thesis. In other words, if an agent $a$ detects some inconsistency in the beliefs of another agent $b$, we do not force $a$ to try and replicate an a priori belief update by $b$, even if agent $b$ is likely do it, in $a$'s estimation. Since a priori updates might be non-deterministic, there is little reason to assume that $a$ would be able to exactly match the thought processes of another agent $b$. In Lewis's terminology, once the belief in shared "inductive standards and background information" fails, so does the ground for a common understanding of the situation. In practice, this means that $a$ would generally lose the ability to interpret $b$'s actions or gain information from them[7]. In effect, from this moment on, $a$

[6] Providing a Kripke model synthesis procedure to produce actually and virtually possible worlds to satisfy specific explicit a priori assumptions is left for future work.

[7] It might be possible to actually communicate an agent's a priori beliefs, but of course, this communication action correspond to an a posteriori update which is about a priori beliefs. Modeling this complex interaction is left for future work.

would treat $b$ as a fully byzantine agent i.e., having inconsistent beliefs. [KPSF19a].

## 5.3 A priori belief updates

The aim of this section is to describe the semantic mechanism an agent $a$ can use to perform a priori belief updates, in order to attempt self-recovery when $a$ discovers that its a posteriori beliefs came into conflict with its a priori beliefs. Note that this means that $a$'s part of the model is empty, making it impossible for $a$ to use the current pointed model to recover a consistent epistemic state (Section 2.2.2). Much like epistemic puzzles are described by semantical models, the a priori belief updates are also semantic: agent $a$ tries to re-imagine the epistemic situation as a *trial model* that is based on the agents's previous experiences, modifications of explicit a priori beliefs, and/or ad hoc guesses. However, $a$ generally has no reason to ascribe these internal attempts to the thinking of other agents. Hence, to model higher-order beliefs for all other agents, $a$ uses some *backup model* that is primarily derived from $a$'s previous experiences and knowledge of a priori beliefs of other agents.

### 5.3.1 General update mechanism

We first formulate this update mechanism to be as general as possible, requiring only the basic coherency restrictions on the trial and backup models. In Section 5.5, we will discuss some strategies autonomous agents may employ to generate these models. We do stress, however, that trial models are intended to be ad hoc guesses. The amount of restrictions on the trial model should be inversely proportional to how wrong agent's beliefs are expected to be: the further away from reality the agent may have strayed, the fewer restrictions should be imposed on its imagination in the process of recovering a consistent state.

For instance, in the consecutive numbers puzzle (Example 114), the trial model should not include non-integer numbers or violate laws of arithmetic. On the other hand, if agents are expected to not be fully attentive to the formulation of the puzzle, a trial model may include pairs of numbers that are not consecutive to account for possible misunderstandings.

**Definition 122** (A priori belief update)**.** *An a priori belief update for agent $a$ is a tuple*

$$U = (\mathcal{M}^a, U^a, \mathcal{M}^{\neg a}, \mapsto) \tag{5.1}$$

*where*

- *the* trial model $\mathcal{M}^a = \langle W^a, R^a, V^a \rangle$ *is a quasi-epistemic Kripke model,*
- $U^a \subseteq W^a$ *is an $a$-cluster within $\mathcal{M}^a$ (note that $U^a \neq \emptyset$),*
- *the* backup model $\mathcal{M}^{\neg a} = \langle W^{\neg a}, R^{\neg a}, V^{\neg a} \rangle$ *is a quasi-epistemic Kripke model,*
- $S \mapsto W^{\neg a}$ *is a* correspondence function *from some subset $S \subseteq W^a$ of the domain of the trial model into the domain of the backup model, i.e., a partial function from $W^a$ to $W^{\neg a}$ that identifies some of the trial worlds with backup worlds*

*such that for any atomic proposition $p \in Prop$ and for any trial worlds $u, v \in W^a$ and backup worlds $u', v' \in W^{\neg a}$, the following* coherency conditions *are fulfilled:*

1. atomic coherency*: if $u \mapsto u'$, then $u \in V^a(p) \iff u' \in V^{\neg a}(p)$, i.e., only propositionally equivalent worlds can be identified;*
2. reasoning coherency*: for each agent $b \neq a$, if $u \mapsto u'$ and $v \mapsto v'$, then $u R_b^a v \iff u' R_b^{\neg a} v'$, i.e., the identification respects indistinguishabilities of all agents but $a$;*
3. simulation coherency*: for each agent $b \neq a$, if $u \mapsto u'$ and $u' R_b^{\neg a} v'$, then there exists $v \in W^a$ such that $u R_b^a v$ and $v \mapsto v'$, i.e., the trial model simulates the backup model for all agents but $a$.*

Intuitively, $U^a$ represents the local state of $a$, i.e., those worlds that $a$ considers actually possible. Relations $R_b^a$ for $b \neq a$ determine whether these new worlds constructed by $a$ would have been distinguishable for other agents, were they aware of $a$'s new trial vision of the world. However, since they are unaware of $a$'s trial model $\mathcal{M}^a$, these indistinguishabilities need to be transferred to the backup model $\mathcal{M}^{\neg a}$, which represents $a$'s virtually possible worlds, used for understanding of how the other agents imagine the epistemic situation. The coherency conditions on $\mapsto$ ensure compatibility between $a$'s actually possible worlds and the virtually possible worlds considered by $a$. In particular, the trial model should simulate the backup model for agents other than $a$ because $a$'s understanding of their epistemic state cannot be worse than $a$'s impression of their own understanding.

The result of applying an a priori belief update to a pointed Kripke model is described by the following definition:

**Definition 123** (Result of a priori belief update)**.** *Let $U = (\mathcal{M}^a, U^a, \mathcal{M}^{\neg a}, \mapsto)$ be an a priori belief update with $\mathcal{M}^a = \langle W^a, R^a, V^a \rangle$ and $\mathcal{M}^{\neg a} = \langle W^{\neg a}, R^{\neg a}, V^{\neg a} \rangle$ and $(\mathcal{M}, v)$ be a pointed Kripke model with introspective model $\mathcal{M} = \langle W, R, V \rangle$ such that $R_a(v) = \varnothing$. The result of agent $a$ applying $U$ to $(\mathcal{M}, v)$ is a pointed Kripke model $(\mathcal{M} \circledcirc_a U, v)$ where $\mathcal{M} \circledcirc_a U := \langle W', R', V' \rangle$ such that:*

- $W' := W \sqcup U^a \sqcup W^{\neg a}$;
- $R'_a := R_a \sqcup R_a^{\neg a} \sqcup (\{v\} \times U^a) \sqcup (U^a \times U^a)$;
- $R'_b := R_b \sqcup R_b^{\neg a} \sqcup \{(u, v') \in U^a \times W^{\neg a} \mid (\exists v \in W^a) u R_b^a v \mapsto v'\}$ *for each agent $b \neq a$;*
- $V'(p) := V(p) \sqcup V^{\neg a}(p) \sqcup (V^a(p) \cap U^a)$ *for any $p \in Prop$.*

Intuitively, the domain of the resulting model is made by the original domain $W$, the a-cluster $U^a$ and the domain of the backup model $W^{\neg a}$. The accessibility relations remain the same in the original model and in the backup model for all agents. The agent performing the update also has corresponding relations from the actual world to $U^a$ and from elements of $U^a$ to elements of $U^a$.

All agents not involved in the update have additional relations from the world $u \in U^a$ to $v \in W^{\neg a}$ whenever there is a correspondence function $\mapsto$ between them. Finally, the evaluation function remain unchanged, but is restricted to the worlds of the a-cluster instead of the whole domain of the trial model for the updating agent.

**Remark 124.** *The requirement of simulation coherency is asymmetric. It is worth asking why it should not be a bidirectional bisimulation instead. Recall that both the trial and backup models describe a's attempt to provide an alternative explanation for the apparent inconsistency of its beliefs. The difference is that the trial model $\mathcal{M}^a$ is a's suggestion of how things "actually are" whereas the backup model $\mathcal{M}^{\neg a}$ is a's attempt to imagine how this new picture of the world is viewed by all other agents based on some standard, default, or past views a expects of them. This puts a in a position of intellectual superiority, akin to that of the system designer of a distributed system. Within this new point of view of a's creation, agent a* does *know better than the other agents, much like the system designer knows better than distributed agents. Indeed, as mentioned before, it is the system designer who performs a priori belief updates in traditional distributed systems, which makes it reasonable for an agent to adapt a similar attitude for the same task. In a's view, $\mathcal{M}^a$ is an "improved" description of the world, which a hopes to be accurate, whereas $\mathcal{M}^{\neg a}$ may well be an erroneous description that a is aware of by rejects. If a thinks that other agents should consider certain possibilities, it is only rational for a to embed these possibilities into its own view of the world to ensure its accuracy by better describing the epistemic reasoning of other agents. At the same time, a can imagine others having blind spots and missing some possibilities a is now considering. A simulation in the opposite direction, from $\mathcal{M}^{\neg a}$ to $\mathcal{M}^a$ would have made such blind spots impossible and, hence, is not required by our definition. At the same time, if a's update is guided by the idea that other agents knew the true state of affairs all along and it is a who has been missing something, then the simulation in the opposite direction would make sense. Hence, while not requiring it, our definition does not precludes a bisimulation between $\mathcal{M}^a$ and $\mathcal{M}^{\neg a}$ either.*

The properties of a priori belief updates are captured by the following theorem:

**Theorem 125.** *Let $U = (\mathcal{M}^a, U^a, \mathcal{M}^{\neg a}, \mapsto)$ be an a priori belief update and $(\mathcal{M}, v)$ be a pointed Kripke model with introspective model $\mathcal{M} = \langle W, R, V \rangle$ such that $R_a(v) = \varnothing$. Then*

1. *$\mathcal{M} \circledcirc_a U$ is an introspective model.*
2. *For any purely propositional formula $\varphi$,*

$$\mathcal{M}, v \vDash \varphi \qquad \Longleftrightarrow \qquad \mathcal{M} \circledcirc_a U, v \vDash \varphi.$$

3. *For any formula $\varphi$ and any agent $b \neq a$*

$$\mathcal{M}, v \vDash B_b\varphi \qquad \Longleftrightarrow \qquad \mathcal{M} \circledcirc_a U, v \vDash B_b\varphi.$$

4. *$\mathcal{M} \circledcirc_a U, v \nvDash B_a\bot$.*

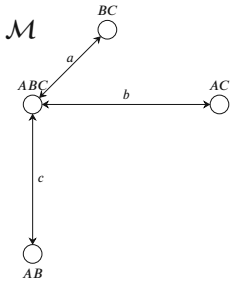



**Figure 5.1:** Kripke model $\mathcal{M}$ where all agents have the common a priori belief $APB_2$.

*Proof.* Let $\mathcal{M} \circledcirc_a \mathsf{U} := \langle W', R', V' \rangle$.

1. Transitivity and euclideanity of $R'_a$ and $R'_b$ for $b \neq a$ follow by construction due to all involved models being introspective. We demonstrate only several non-trivial cases. (a) It cannot happen that $wR_av$ and $vR'_au$ for some $u \in U^a$ because $wR_av$ would imply $vR_av$ by euclideanity of $R_a$, whereas we assumed $R_a(v) = \varnothing$. (b) Let $b \neq a$. If $uR'_bv'$ because $uR^a_bv{\mapsto}v'$ and $v'R^{\neg a}_bz'$ for some $u \in U^a$, $v \in W^a$, and $v', z' \in W^{\neg a}$, then there exists $z \in W^a$ such that $vR^a_bz{\mapsto}z'$ by the simulation coherency condition. Hence, $uR^a_bz$ by transitivity of $R^a_b$, which implies $uR'_bz'$, as required. (c) Let $b \neq a$. If $uR'_bv'$ because $uR^a_bv{\mapsto}v'$ and $uR'_bz'$ because $uR^a_bz{\mapsto}z'$ for some $u \in U$, $v, z \in W^a$, and $v', z' \in W^{\neg a}$, then $vR^a_bz$ by euclideanity of $R^a_b$. Hence, $v'R^{\neg a}_bz'$ by the reasoning coherency condition, as required.

2. Easily follows from the fact that the propositional valuation of $v$ does not change.

3. Easily follows from $R'_b(v) = R_b(v)$.

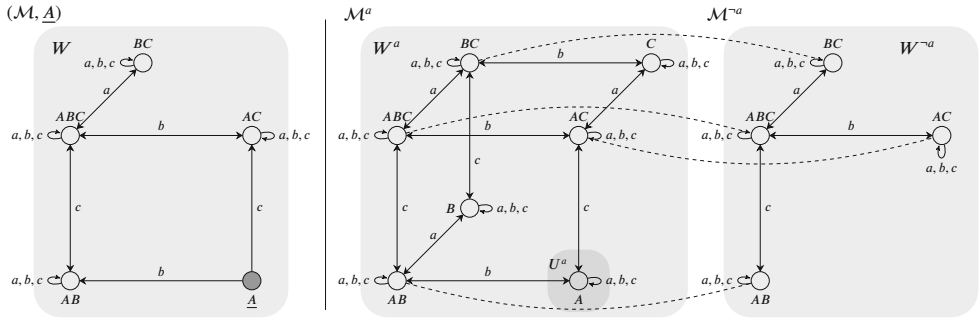
4. Easily follows from the fact that $R'_a(v) = U^a \neq \varnothing$. □

**Remark 126.** *Theorem 125.3 says that a performing a successful update of its a priori beliefs cannot resolve inconsistent beliefs of other agents (though a may erroneously believe to have resolved them). It is reasonable to expect each of the agents with inconsistent beliefs to perform such an operation. We will describe how to do it and why updates of different agents are completely independent from each other in Section 5.4.2.*

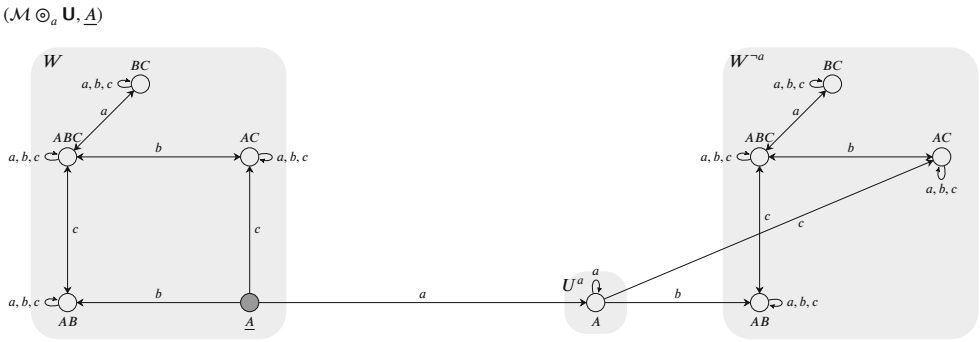
## 5.4 Application to the muddy children and consecutive numbers puzzles

We now illustrate this a priori update mechanism by considering muddy children puzzle (Section 1.2.4) variants and with various additional a priori assumptions.

**Example 127** (MCP with false a priori assumption resolved by a priori belief update)**.** *Consider again the variant of the muddy children puzzle ( Example 50) where, after playing the same game*




**(a)** Initial pointed Kripke model $\mathcal{M}, \underline{A}$ restricted to $APB_2$ (Left) and elements (Middle and Right) of a priori belief update $\mathsf{U} = (\mathcal{M}^a, U^a, \mathcal{M}^{\neg a}, \mapsto)$ for agent $a$ from Example 127. The correspondence relation $\mapsto$ is represented by dashed lines.




**(b)** Updated Kripke model $(\mathcal{M} \odot_a \mathsf{U}, \underline{A})$ from Example 127. In particular, arrows from $U^a$ to worlds in $W^{\neg a}$ are drawn in accordance to the correspondence relation $\mapsto$, and all worlds in $U^a$ are reachable from $\underline{A}$ via $a$-arrows as for Definition 123.

**Figure 5.2**

*every day, three children a, b, and c have come to the conclusion that at least two of them always get muddy, i.e., they formed a common a priori belief that*

$$APB_2 = (m_a \wedge m_b) \vee (m_a \wedge m_c) \vee (m_b \wedge m_c).$$

*(Note that $APB_2$ does not represent* all *common a priori beliefs of the children. Most of those are encoded in epistemic model $\mathcal{M}_0$.)*

*In terms of Kripke modeling, this can be viewed as a public announcement of $APB_2$, which results in model $\mathcal{M}_0$ from Figure 5.1 they used to commonly consider as the base shrinking to the Kripke model of four worlds only (AB, AC, BC, and ABC), making it unnecessary for Father to announce anything. Suppose that this time around, contrary to habit, only a is muddy, i.e., $m_a \wedge \neg m_b \wedge \neg m_c$. Thus, their common a priori beliefs $APB_2$ turn out to be false. This scenario is depicted as pointed model $(\mathcal{M}, \underline{A})$ in Figure 5.2a (Left), which is obtained from the four-world model by adding a real world $\underline{A}$[8] and drawing arrows from it to the four worlds in accordance*

[8] It is named $A$ because its propositional valuation is the same as that of $A$ in $\mathcal{M}_0$, while the underline means it is

*with what children would consider possible. Note that there are now one-directional arrows due to the fact that children have false beliefs as the result of false a priori assumptions.*

*While beliefs of all agents are false, only a can detect this because a's beliefs are inconsistent, unlike those of b and c:*

$$\mathcal{M}, \underline{A} \vDash B_a\bot, \text{ while } \mathcal{M}, \underline{A} \nvDash B_b\bot \text{ and } \mathcal{M}, \underline{A} \nvDash B_c\bot.$$

*Thus, a is the only child who is justified to update her a priori beliefs. Note that no public announcement was made, and that a's inconsistency is the result of a's reasoning about the epistemic situation.*

*The introduction of one-directional arrows changes the whole interpretation of the epistemic situation. Indeed, using the assumption of the common knowledge of the model, it could be tempting to suggest that agents b and c can detect a's inconsistency. It would not, however, match the underlying scenario. For instance, when b sees that a is muddy while c is not, we would expect b to conclude that b is the second muddy child and that a sees the mud on b's forehead (the model says as much by providing a unidirectional b-arrow from $\underline{A}$ to $AB$). To match this intuition, we are forced to abandon the assumption of common knowledge of the model. For the pointed model $(\mathcal{M}, \underline{A})$, agent b is only aware of the world $AB$, the only possible for b, and all worlds accessible from it by a sequence of arrows, agent c is only aware of the world $AC$, the only possible for c, and all worlds accessible from it by a sequence of arrows, while a is not aware of any worlds, resulting in inconsistent beliefs. In particular, a has lost the ability to examine the reasoning of other agents. While it is natural for a to assume that they still operate within the last commonly considered model, i.e., the four-world $\mathcal{M}^{\neg a}$, connecting this assumption to a's own reality requires this reality to be fleshed out by a, which is the purpose of $\mathcal{M}^a$ (and, of its restriction $U^a$).*

*According to Definition 122, to perform an a priori belief update, agent a must come up with a trial model $\mathcal{M}^a$. Treating the process as ultimately a creative one, we eschew an explanation of how to do that (providing some heuristics in Section 5.5). Instead we consider several alternative trial models in this and following examples. The success of the a priori update depends largely on how good the choices made by a are, including crucially the choice of the trial model. If agent a manages to guess a trial model reflecting the actual world well (recall that the actual world is not considered possible by a), the a priori update is likely not only to resolve the inconsistency of a's beliefs, but also result in new beliefs that are factive. This is the case we demonstrate in the current example. Other examples will demonstrate that a bad choice may not remove the inconsistency, or worse still, create new consistent beliefs that are very far from reality. Let $\mathcal{M}^a$ be as depicted in Figure 5.2a (Middle), which fits well with what she observes, if the singleton cluster $U^a = \{A\}$ of worlds from $\mathcal{M}^a$ is to represent the possibilities a considers actually possible based on her observations. As already mentioned, for the backup model, which a uses for computing what is considered by b and c, agent a takes the four-world model $\mathcal{M}^{\neg a}$ she herself used until recently. The elements of the a priori belief update $U = (\mathcal{M}^a, U^a, \mathcal{M}^{\neg a}, \mapsto)$ are depicted in Figure 5.2a (Right). In particular, since $\mathcal{M}^a$ extends $\mathcal{M}, \underline{A}$ with three additional worlds where only one agent is muddy, agent a establishes the correspondence function $\mapsto$ to connect each world from $\mathcal{M}^a$ to a*

the actual world.

*propositionally equivalent world of $\mathcal{M}^{\neg a}$ if such a world exists. It is easy to see that the coherency conditions for the a priori belief update are fulfilled. According to Definition 122, the result of applying the update to the initial model $\mathcal{M} \circledcirc_a U, \underline{A}$ is shown in Figure 5.2b. The two arrows from $A \in U^a$ for b and c to worlds in $W^{\neg a}$ are added because of the correspondence relation $\mapsto$. For instance, the b-arrow from A in $U^a$ to AB in $\mathcal{M}^{\neg a}$ of the updated model is due to the b-arrow from A to AB in $\mathcal{M}^a$ and the correspondence from AB in $\mathcal{M}^a$ to AB in $\mathcal{M}^{\neg a}$. As a result*

$$\mathcal{M} \circledcirc_a U, \underline{A} \vDash B_b(m_a \wedge m_b \wedge \neg m_c), \qquad \mathcal{M} \circledcirc_a U, \underline{A} \vDash B_a B_b(m_a \wedge m_b \wedge \neg m_c),$$
$$\mathcal{M} \circledcirc_a U, \underline{A} \vDash B_c(m_a \wedge \neg m_b \wedge m_c), \qquad \mathcal{M} \circledcirc_a U, \underline{A} \vDash B_a B_c(m_a \wedge \neg m_b \wedge m_c).$$
$$\mathcal{M} \circledcirc_a U, \underline{A} \vDash B_a(m_a \wedge \neg m_b \wedge \neg m_c), \qquad \mathcal{M} \circledcirc_a U, \underline{A} \vDash B_b B_a(m_a \wedge m_b \wedge \neg m_c),$$
$$\mathcal{M} \circledcirc_a U, \underline{A} \vDash B_c B_a(m_a \wedge \neg m_b \wedge m_c),$$

*In other words, each agent believes to know the actual situation and to be muddy. All three of them will step forward when Father prompts. Moreover, the a priori update restored a's ability to understand the reasoning of others. Because a guessed $\mathcal{M}^{\neg a}$ correctly, she can correctly interpret b and c stepping forward. Agents b and c expect a to step forward, but for a wrong reason. On the other hand, b cannot interpret c stepping forward because $\mathcal{M} \circledcirc_a U, \underline{A} \vDash B_b(\neg B_c m_c \wedge \neg B_c \neg m_c)$ and vice versa.*

### 5.4.1 A priori belief update triggered by public announcement

One could say that Example 127 should have been modeled according to Figure 5.2b from the very beginning because it fits better with the evidence observed by the agents than model $\mathcal{M}^{\neg a}$ from Figure 5.2a they attempt to use. That would not address the question of how to turn an epistemic description of agents' observations and beliefs in complex epistemic scenarios, possibly with false common beliefs of a subset of agents. Our a priori belief updates provide a mechanism for building models for more complex epistemic situations based on existing models for simpler scenarios.

It is harder to find an alternative to a priori belief updates when the inadequacy of the a priori assumptions cannot be observed initially and is only uncovered through communication. As already mentioned, the idea of scrapping the original model and starting anew corresponds to the development cycle of ordinary distributed systems, where it is performed by a system designer. We aim at developing a mechanism for individual agents to do it, as befits SASO systems. Moreover, as we saw in Example 127, different agents are likely to discover the flaw in their assumptions at different times, which puts the idea of scrapping the *whole* model at odds with the Knowledge of Preconditions Principle [Mos15] that requires an agent performing an action to know the reason for this action.

This was the situation in Example 114 where it was *a*'s public announcement of knowing *b*'s number that caused *b* to realize that some a priori assumptions must have been wrong. We now show how to perform a priori belief update in such a situation by representing *b*'s reasoning in Example 114 in our formal framework.

**Example 128** (Consecutive numbers with false a priori assumptions formalized)**.** *Let us describe the twist version of Example 114 as a Kripke model. As before, we label worlds according to*

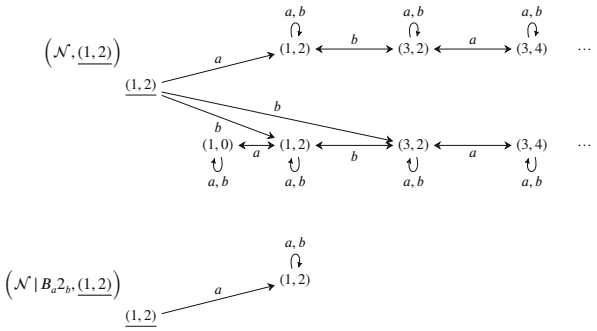
*pairs of numbers held by the agents. Although this results in several worlds having the same label, we prefer to disambiguate when necessary rather than creating an overly complex naming system for worlds. As before, the actual world* $\underline{(1,2)}$ *is distinguished by the underline. Unlike the MCP, agents a and b do not have common a priori assumptions here: a starts natural numbers from* 1 *while b also considers* 0 *to be natural. At the same time, none of them are initially aware of this disagreement. Hence, pointed Kripke model* $\left(\mathcal{N}, \underline{(1,2)}\right)$ *in Figure 1.3a (Above) is introspective but not epistemic and consists of the actual world and two disjoint fragments, each only accessible to one of the two agents: the upper line of worlds represents what a (mistakenly) thinks is their common view of the situation, while the lower line of worlds represents what b (mistakenly) takes to be their common view. To distinguish similarly labeled worlds, we call the worlds from the upper (lower) line a-worlds (b-worlds). To simplify notation without affecting the logical content of the puzzle, suppose a states b's number explicitly by saying "I know that you have number* 2*," which we represent by formula* $B_a 2_b$*. It is easy to see that* $\mathcal{N}, w \not\models B_a 2_b$ *for all b-worlds w; of the a-worlds, only the world* $(1,2)$ *satisfies* $B_a 2_b$*, and so does the actual world. Accordingly, the result of a public update with* $B_a 2_b$*, as depicted in Figure 1.3a (Below), contains only two worlds and* $\mathcal{N} \mid B_a 2_b, \underline{(1,2)} \vDash B_b \bot$*, prompting the "Wait, what?" comment in our informal rendition and triggering an a priori update by b.*

*To match the informal reasoning from Example 114, consider the nature of this public announcement: it was about a's beliefs rather than propositional facts. Hence, it is rational for b to return to the original model he considered before the public announcement: model* $\mathcal{M}^b$ *from Figure 5.3b is nothing but b-worlds from Figure 1.3a (Above) and* $U^b$ *is the b-cluster of worlds* $(1,2)$ *and* $(3,2)$ *that b considered possible pre-announcement. What b should modify is how a sees the situation. For that b correctly imagines the abbreviated natural numbers, resulting in* $\mathcal{M}^{\neg b}$ *being the same as a-worlds from Figure 1.3a (Above). Once again, the correspondence is determined by propositional truth of atomic propositions* $n_a$ *and* $m_b$*. It is easy to see that all coherency conditions are satisfied.*

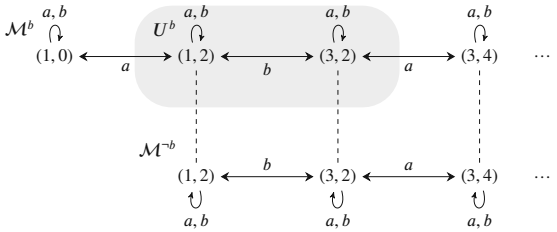
*The result of b applying a priori update* $U$ *to* $\left(\mathcal{N} \mid B_a 2_b, \underline{(1,2)}\right)$ *is shown in Figure 5.3c (Left). But this is not the model explaining why b now knows a's number. Indeed, this cannot be the final stage of b's reevaluation since b's part of this model does not reflect the public announcement a made. Agent b still needs to apply the standard public announcement update to the only part of the model it is aware of, which is comprised of worlds from* $U^b$ *and* $W^{\neg b}$*. This leaves b's part of the model with two worlds only,* $(1,2)$ *from* $U^b$ *and* $(1,2)$ *from* $W^{\neg b}$*, as shown in Figure 5.3c (Right). It can be easily seen that* $\left((\mathcal{N} \mid B_a 2_b) \circledcirc_b U\right)|_b B_a 2_b, \underline{(1,2)} \vDash B_b 1_a$*, explaining that b has now figured out a's number too.*

*Such a "private" public announcement update* $|_b$ *may not be standard, but it is warranted in this setting. Recall that b here assumes the role of the system designer and treats his part of the model as the whole model and the only model pre-announcement. It is natural then for b to update this partial model as if it is the whole model. This operation should not affect a's part of the model since b is not aware of it.*

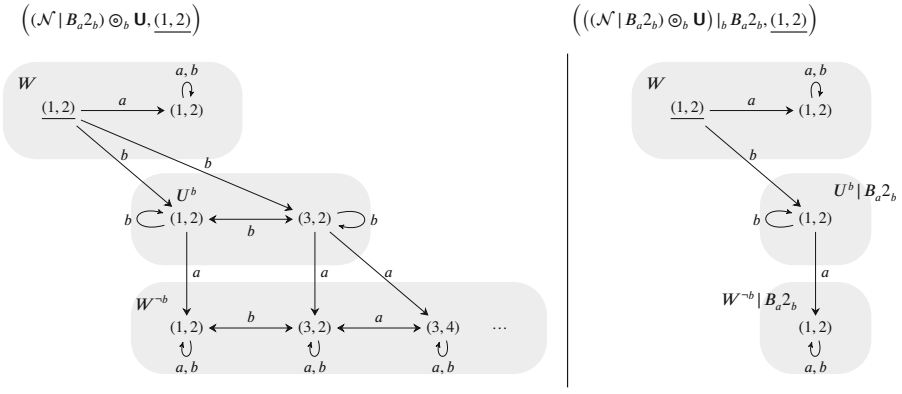
If, in the a priori updated model after the public announcement there is at least one world in $U^a$




**(a)** Introspective pointed Kripke model $\left(\mathcal{N}, \underline{(1,2)}\right)$ representing the initial state of both agents in the consecutive numbers puzzle from Example 128, where $b$ considers 0 to be a natural number while $a$ does not (Above). $\left(\mathcal{N} \mid B_a 2_b, \underline{(1,2)}\right)$ after $a$'s public announcement that $a$ knows $b$'s number is 2 (Below).




**(b)** Elements of the a priori belief update $\mathsf{U} = (\mathcal{M}^b, U^b, \mathcal{M}^{\neg b}, \mapsto)$ for $b$ in Example 128. The correspondence relation is represented by dashed lines. The grey rectangle is $b$'s equivalence class $U^b$.




**(c)** Result of $b$ applying a priori belief update in the inconsistent epistemic state of $\left(\mathcal{N} \mid B_a 2_b, \underline{(1,2)}\right)$ of Figure 1.3a (Left). Result of $b$ (re)applying $a$'s public announcement of $B_a 2_b$ to $b$'s part of the model (Right).

**Figure 5.3**

that makes the announced formula true, then the self-recovery operation is indeed successful, as stated by the following corollary:

**Corollary 129.** *Let $U = (\mathcal{M}^a, U^a, \mathcal{M}^{\neg a}, \mapsto)$ be an a priori belief update triggered by the public announcement of $\varphi$ and $(\mathcal{M}, v)$ be a pointed Kripke model with introspective model $\mathcal{M} = \langle W, R, V \rangle$. If $(\mathcal{M} \mid \varphi) \circledcirc_a U, u \vDash \varphi$ for some world $u \in U^a$, then a's beliefs are consistent after the a priori update, i.e., $\big((\mathcal{M} \mid \varphi) \circledcirc_a U\big) \mid_a \varphi, v \nvDash B_a \bot$.*

*Proof.* The statement follows from Theorem 125.4. □

Note that because of Moorean sentences (see [BR16]) of the form $\varphi \wedge B_i \neg \varphi$, there is generally no guarantee that $a$ believes in $\varphi$ after the public announcement.

### 5.4.2 Simultaneous a priori belief updates by several agents

So far we considered only scenarios where beliefs of exactly one agent become inconsistent, resulting in a priori belief update for this agent. It is, of course, entirely possible that several agents become inconsistent simultaneously. Given that a priori belief update is performed by an agent privately and has no affect on the rest of the model, it is straightforward to apply several such updates in parallel.

**Example 130** (Simultaneous a priori belief updates triggered by a public announcement)**.** *Let us continue the scenario from Example 127 where we left off, i.e., after a has restored her consistency, Father asked the children if they know whether they are muddy, and all three children stepped forward, which is equivalent to the public announcement of "all step forward"*

$$ASF = (B_a m_a \vee B_a \neg m_a) \wedge (B_b m_b \vee B_b \neg m_b) \wedge (B_c m_c \vee B_c \neg m_c). \tag{5.2}$$

*The resulting model $\mathcal{M}' := \big(\mathcal{M} \circledcirc_a U\big) \mid ASF$ is shown in Figure 5.4a and has only two worlds remaining from Figure 5.2b, the actual world $\underline{A}$ and world $A$ from $U^a$. Thus, beliefs of b and c have become inconsistent (there are no outgoing b- or c-arrows from the actual world $\underline{A}$) and a knows that this is the case (there are no outgoing b- or c-arrows from A, the only world a considers possible). At the same time, a still maintains consistency of beliefs and thinks of herself as muddy:*

$$\mathcal{M}', \underline{A} \vDash B_b \bot \wedge B_c \bot \wedge B_a B_b \bot \wedge B_a B_c \bot \wedge \neg B_a \bot \wedge B_a m_a.$$

*At this point b and c independently perform each their own a priori belief update triggered by the announcement of ASF. Although a can expect them to do so, one could argue that, due to the non-deterministic* ad hoc *nature of a priori belief updates, a does not think she can guess how b and c would choose to update their beliefs, nor is such a guess necessary for a to be able to respond to Father. Accordingly, a remains satisfied in her belief that she is muddy and that her companions are confused.*

*Suppose that each of b and c chooses to do the same thing that a has done earlier, i.e., using $\mathcal{M}^a$ from Figure 5.2a (Middle) as both $\mathcal{M}^b$ and $\mathcal{M}^c$ and using $\mathcal{M}^{\neg a}$ from Figure 5.2a (Right) as both*

*$\mathcal{M}^{\neg b}$ and $\mathcal{M}^{\neg c}$. Correspondence $\mapsto$ is also the same for both b and c. However, their local states $U^b = \{A, AB\}$ and $U^c = \{A, AC\}$ differ due to their differing points of view.*

*We present the state after b and c's simultaneous independent a priori update in Figure 5.4b. They then each apply ASF to their private parts of the model respectively. No worlds in $W^{\neg b}/W^{\neg c}$ satisfy ASF, so all are pruned. World AB of $U^b$ is rejected because there c does not know its state,*

$$\big((\mathcal{M} \circledcirc_a U) \mid ASF\big) \circledcirc_{b,c} U', AB \not\models B_c m_c \vee B_c \neg m_c;$$

*similarly AC of $U^c$ does not survive the announcement of ASF since b there is not sure of its state. This leaves only worlds A from $U^b$ and $U^c$ (and the whole a's part of the model, which is not affected by a priori thinking of b and c), resulting in Figure 5.4c. Note that now each agent believes that the beliefs of other two agents are inconsistent while themselves correctly believing that a is the only muddy child. In effect, all children solved the problem correctly but lost trust in each other's reasoning: using*

$$\mathcal{M}'' := \Big(\big((\mathcal{M} \circledcirc_a U) \mid ASF\big) \circledcirc_{b,c} U'\Big)|_{b,c} ASF$$

*for the model, $E\varphi := B_a\varphi \wedge B_b\varphi \wedge B_c\varphi$ for mutual knowledge, and $\hat{E}\varphi := \neg E\neg\varphi$ for its dual,*

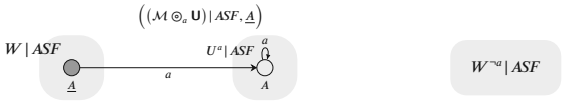
$$\mathcal{M}'', \underline{A} \vDash E(m_a \wedge \neg m_b \wedge \neg m_c) \wedge \neg\hat{E}\bot \wedge B_a(B_b\bot \wedge B_c\bot) \wedge B_b(B_a\bot \wedge B_c\bot) \wedge B_c(B_a\bot \wedge B_b\bot).$$

*Note that update $U'$ in Figure 5.4b represents two separate a priori belief updates, one by b (rectangles $U^b$ and $W^{\neg b}$) and another by c (rectangles $U^c$ and $W^{\neg c}$). Hence, here we take $U'$ to be a partial function from the set $\mathcal{A}$ of agents to a priori belief update tuples, so that $U'(b)$ is the a priori update performed by b while $U'(c)$ is the one for c. This notation is similar to using $R$ for accessibility relations of all agents. The domain of $U'$ is then listed in the subscript to $\circledcirc$ as in $\big((\mathcal{M} \circledcirc_a U) \mid ASF\big) \circledcirc_{b,c} U'$. This does not cause any formal problems because, being private, individual a priori belief updates do not interfere with each other.*

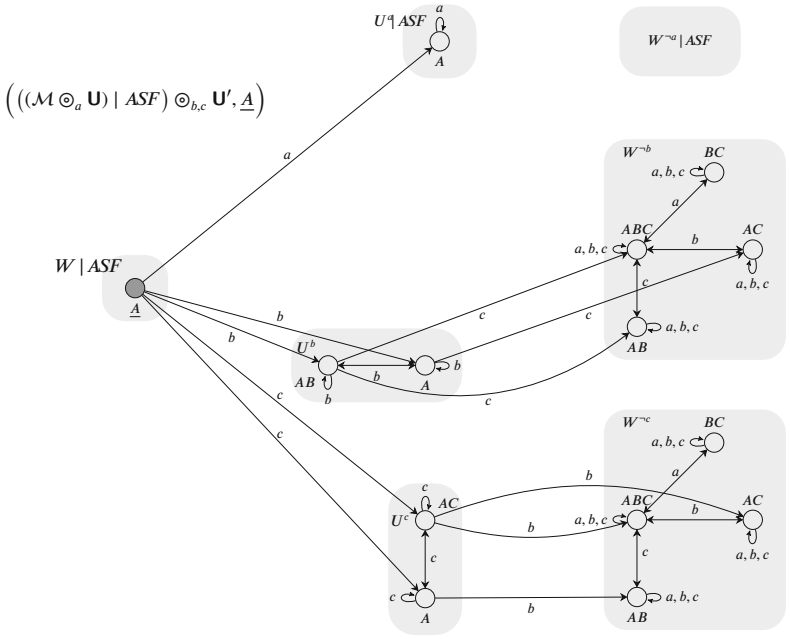
### 5.4.3 A priori belief updates need not yield (correct) solutions

In all examples considered so far, all agents performing the updates have succeeded in recovering a consistent epistemic state (though sometimes at the expense of expecting consistency from other agents). By no means do we claim that this is always the case. The *ad hoc* chosen a priori belief update U need not lead to the resolution of the agent's conundrum. It is rather a matter of luck whether it would. For instance, consider the setup of Example 128. Previously, we had agent *b* guess the exact model of the number line used by *a*. But *b*'s guess can also be wrong. For instance, *b* might think that *a* mistakenly considers all integers rather than non-negative integers only, making $\mathcal{M}^{\neg b}$ from Figure 5.3c extend infinitely in both directions, retaining world $(1, 0)$, and not resulting in a consistent state for *b* after $B_a 2_b$ is taken into account. If the agent is persistent, it should be expected that such an unsuccessful update is rejected and a new attempt is made to restore consistency using a different a priori belief update.

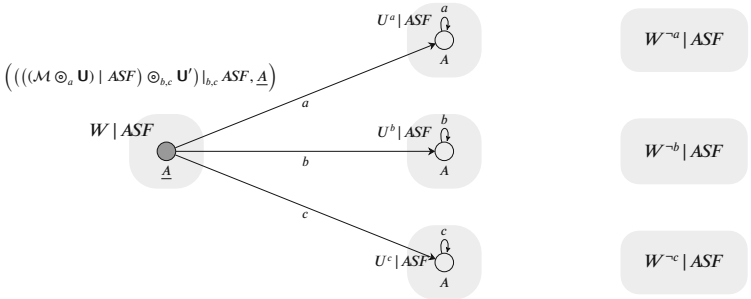
It is also possible that a wrong a priori guess does not result in an inconsistency but leads to wrong conclusions, undetected by any of the agents.




**(a)** Pointed Kripke model representing the result of the public announcement *ASF* to the a priori updated model $(\mathcal{M} \circledcirc_a \mathsf{U})$ of Example 127 described in Example 130.




**(b)** Pointed Kripke model showing the intermediate state of Example 130 after *b* and *c* independently performed a priori belief updates in parallel but have not yet (re)applied the public announcement *ASF*.




**(c)** Pointed Kripke model illustrating the final stage of Example 130 after a priori belief updates of *b* and *c* and their application of the public announcement of *ASF* to their newly created parts of the model.

**Figure 5.4**

**Example 131** (Simultaneous a priori belief updates creating consistent false beliefs)**.** *Consider a variant of Example 127 where all agents have the same initial explicit a priori common belief $APB_2$, but all are clean in actuality. In this case, all agents detect inconsistency from the start, and each privately and independently performs an a priori belief update. Suppose that, using the same reasoning as agent a in Example 127 and agents b and c in Example 130, agents use a simultaneous update $U''$ such that:*

$$U''_a := (\mathcal{M}^a, \{A\}, \mathcal{M}^{\neg a}, \mapsto), \qquad U''_b := (\mathcal{M}^b, \{B\}, \mathcal{M}^{\neg b}, \mapsto), \qquad U''_c := (\mathcal{M}^c, \{C\}, \mathcal{M}^{\neg c}, \mapsto),$$

*where the only difference to the previous case is in their local states $V^a = \{A\}$, $V^b = \{B\}$, and $V^c = \{C\}$ that are chosen based on what they observe. The resulting model is depicted in Figure 5.5a. It is easy to see that:*

$$\mathcal{M} \circledcirc_{a,b,c} U'', \underline{0} \vDash B_a(m_a \wedge B_b m_b \wedge B_c m_c) \wedge B_b(m_b \wedge B_a m_a \wedge B_c m_c) \wedge B_c(m_c \wedge B_a m_a \wedge B_b m_b).$$

*In other words, after this a priori belief update each child now erroneously believes that (I) it is the only muddy child and (II) other children "erroneously" believe to also be muddy.*[9] *Hence, all children step forward, a public update of ASF resulting in a model shown in Figure 5.5b. This behavior conforms to everyone's expectations, so all children preserve consistency of beliefs. However, each child thinks that this behavior should have puzzled the other two, so that after stepping forward they think each other to have inconsistent beliefs:*

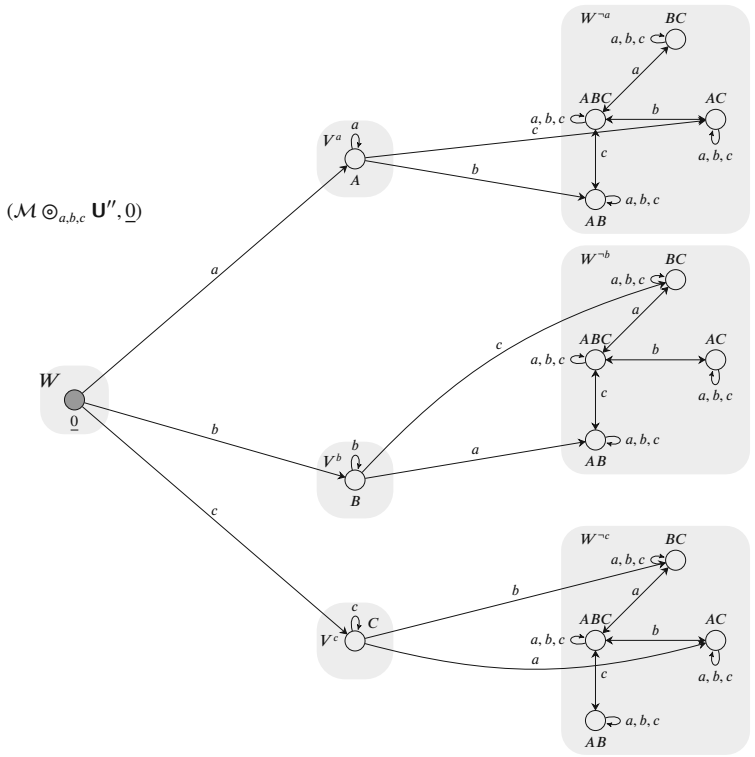
$$(\mathcal{M} \circledcirc_{a,b,c} U'') \,|\, ASF, \underline{0} \vDash B_a(B_b\bot \wedge B_c\bot) \wedge B_b(B_a\bot \wedge B_c\bot) \wedge B_c(B_a\bot \wedge B_b\bot).$$

## 5.5 Some heuristics for a priori belief updates

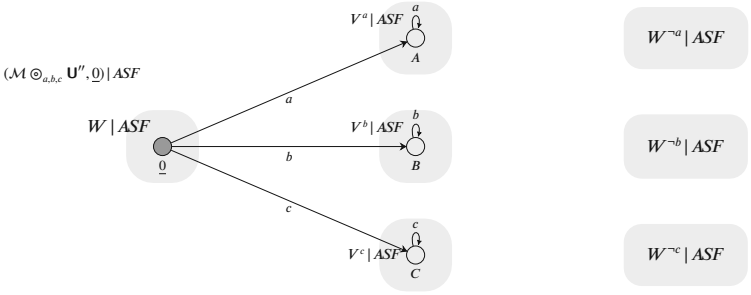
In developing a method for agents to "think outside the box" we did not want to put any boundaries or restrictions on the kinds of trial and backup models used in a priori belief updates, did not want to "box them in" as it were. At the same time, there exist rather regular methods of generating a priori belief updates, some of which we have already used. Let us outline some of these methods, which can easily be implemented in a form of an exhaustive trial-and-error search through finitely many possibilities.

**Heuristics 1: Predefined master models** If an agent is aware of several incompatible alternative models of reality but is not sure which one of them better suits its observations and/or is not sure which points of view other agents entertain, then a priori belief updates can be generated by assigning various combinations of these models to the agent itself and to other agents. This method was used in Example 128 for two different models of natural numbers.

[9] The word *erroneously* is here in air quotes because it has a flavor of a double Gettier example [Get63]. Firstly, *a* believes that *b* considers itself to be muddy, *b* does in fact consider itself to be muddy, but not for the reason *a* expects, as manifested by the difference between *b*'s part of the model representing *b*'s beliefs and $W^{\neg a}$ representing *a*'s rendition of *b*'s beliefs. Secondly, *a* thinks that *b* is wrong, i.e., that *b* is clean, *a* is in fact correct, but not for the reason *a* expects, as manifested by the difference between the real world $\underline{0}$ and world *A* of $V^a$.




**(a)** Pointed Kripke model representing the intermediate state of Example 131 after all agents independently performed a priori belief updates before any public announcement being made.




**(b)** Pointed Kripke model representing the final state of Example 131 after all agents independently performed a priori belief updates in parallel and have applied the public announcement $ASF$.

**Figure 5.5**

**Heuristics 2: Varying explicit a priori assumptions** Even when all agents agree on the same model, e.g., representing the general rules of the puzzle, they may differ in additional a priori assumptions each of them makes. Here one master model represents (common) implicit a priori assumptions while additional assumptions are private and explicit in that they are represented by formulas, imposed on the master model. For instance, $APB_2$ was such an explicit assumption in Example 127 imposed on the implicit assumptions modeled by $\mathcal{M}_0$ from Figure 1.2a. In many of our MCP-related examples, agent $a$ continues using $\mathcal{M}_0 \mid APB_2$ as $\mathcal{M}^{\neg a}$ to describe reasoning of others while itself switching to $\mathcal{M}_0 \mid APB_1$ as $\mathcal{M}^a$ for their own reasoning, where $APB_1 := m_a \vee m_b \vee m_c$ states that at least one child is muddy. If such an a priori belief update fails to restore their own consistency, they can then switch to the full $\mathcal{M}_0$ or consider a more complex a priori restriction.

**Heuristics 2a: Choosing explicit a priori assumptions based on prior failures** Using the master model, if the initial model or one of the previous unsuccessful attempts to reach a consistent state were based on some explicit a priori beliefs $APB$, then it is reasonable to attempt new explicit beliefs of the form $APB' \wedge \neg APB$ in constructing $\mathcal{M}^a$ because the possibility of $APB$ holding has already been rejected.

**Heuristics 3: Learning from inconsistencies arising from public announcements** If an agent $a$ reaches an inconsistency as a result of the factual public announcement of $\varphi$, then $\varphi$ is a valuable information for guiding the a priori reasoning process: in fact, to recover consistency, $\varphi$ should be true in at least one world of the equivalence class of the trial model $U^a$, so that when the public announcement is re-applied, the inconsistency won't arise again.

**Heuristics 4: Changing the underlying logic** It is also possible to relax the restrictions imposed on the trial and/or backup models. For instance, $a$ may try to explain the beliefs of $b$ and $c$ by the lack of positive and/or negative introspection on their part, which would necessitate the relaxation of the requirements of transitivity and euclideanity on the backup model.

## 5.6 Properties of a priori belief updates

This would be an appropriate place to list the axioms of a priori belief updates, perhaps, akin to axioms of PAL for public announcements from [Pla89]. Unfortunately, this does not seem to be any easier to do than to provide a finite syntactic description of common epistemic puzzles. Even restricting our attention to finite models only, the question amounts to asking whether any finite combination of maximal consistent sets for the logic can be represented by a formula (or finitely many formulas). The problem is that there are uncountably many maximal consistent sets (for an infinite set of atomic propositions), hence, uncountably many finite combinations thereof, while only countably many formulas (cf. [Art22]). It seems difficult if not hopeless to try describing syntactically the result of an update when no syntactic description of the update exists.

Thus, in this section we try to "model check" rather than axiomatize the results of a priori belief updates.

**Theorem 132.** *Let $U = (\mathcal{M}^a, U^a, \mathcal{M}^{\neg a}, \mapsto)$ be a single-agent a priori belief update with trial model $\mathcal{M}^a = \langle W^a, R^a, V^a \rangle$ and backup model $\mathcal{M}^{\neg a} = \langle W^{\neg a}, R^{\neg a}, V^{\neg a} \rangle$ and let $(\mathcal{M}, v)$ be a pointed Kripke model with $\mathcal{M} = \langle W, R, V \rangle$ and $R_a(v) = \varnothing$. The following properties hold after agent a's a priori belief update:*

1. *For any formula $\psi$ that does not involve any modalities $B_b$ for agents $b \neq a$, including for all purely propositional formulas, the following three statements are equivalent:*

   a) $\mathcal{M} \circledcirc_a U, v \vDash B_a\psi$;

   b) $\mathcal{M}^a, u \vDash \psi$ *for all* $u \in U^a$;

   c) $\mathcal{M}^a, u \vDash B_a\psi$ *for at least one* $u \in U^a$.

   *In other words, agent a's factual beliefs and a's beliefs about its own beliefs are fully determined by worlds from $U^a$ of $\mathcal{M}^a$.*

2. $\mathcal{M}^{\neg a}, w \vDash \varphi$ *iff* $\mathcal{M} \circledcirc_a U, w \vDash \varphi$ *for any* $w \in W^{\neg a}$ *and any formula* $\varphi$. *In other words, agent a's higher-order beliefs about other agents' beliefs are fully determined by $\mathcal{M}^{\neg a}$.*

*Proof.* 1. The first statement easily follows from the fact that the identity relation on $U^a$ is an $a$-bisimulation between $\mathcal{M}^a$ and $\mathcal{M} \circledcirc_a \mathsf{U}$ (where $a$-bisimulation means that forth and back conditions are restricted to $R_a$ transitions).

2. This follows from the fact that the identity relation on $W^{\neg a}$ is a full bisimulation between Kripke models $\mathcal{M}^{\neg a}$ and $\mathcal{M} \circledcirc_a \mathsf{U}$. □

**Remark 133.** *For an a priori update $U = (\mathcal{M}^a, U^a, \mathcal{M}^{\neg a}, \mapsto)$, one might think that global properties of models $\mathcal{M}^a$ and $\mathcal{M}^{\neg a}$ would be transferred to a's part of the model after the a priori update $U$, so that after the update a believes all global properties used in the update construction. This does hold for propositional validities, as follows from the preceding theorem. In fact, truth of $\psi$ in $U^a$ of $\mathcal{M}^a$ is already sufficient to ensure a's post-update belief $B_a\psi$ if $\psi$ is purely propositional. However, this transfer of global properties fails for epistemic formulas involving other agents. Consider, for instance, the a priori update $U$ from Figure 5.2a for Example 127. For formula ASF from* (5.2), *it is clear that $\mathcal{M}^a \vDash \neg ASF$ and $\mathcal{M}^{\neg a} \vDash \neg ASF$. After all, we know that, in the standard MCP, children do not step forward all at once unless all are muddy. However, in model $\mathcal{M} \circledcirc_a U$ from Figure 5.2b, we have $\mathcal{M} \circledcirc_a U, A \vDash ASF$, resulting in $\mathcal{M} \circledcirc_a U, \underline{A} \vDash B_a ASF$. Thus, global assumptions about other agents' beliefs need not survive in the face of the mismatch between two different points of view: the one a reserves for itself vs. the one a assigns to others. This negative result also extends to implicit assumptions such as factivity of beliefs since in the same example both $\mathcal{M}^a$ and $\mathcal{M}^{\neg a}$ represented agents with factive beliefs, but the model after the a priori update lacks reflexivity for b and c predictably resulting in their beliefs not being factive.*

**Theorem 134.** *When agent a's a priori update $U$ triggered by a public statement $\varphi$ of agent b results in agent a believing that agent b has inconsistent beliefs, (re)applying b's public statement after the a priori update does not affect a's factual beliefs, as well as a's beliefs about beliefs of*

*a and b. In other words, if* $\mathcal{M} \mid B_b\varphi, v \vDash B_a\bot$ *and* $(\mathcal{M} \mid B_b\varphi) \circledcirc_a U, v \vDash B_aB_b\bot$*, then for any formula* $\psi$ *that does not involve any modalities* $B_c$ *for agents* $c \notin \{a, b\}$*,*

$$(\mathcal{M} \mid B_b\varphi) \circledcirc_a U, v \vDash B_a\psi \qquad \Longleftrightarrow \qquad \big((\mathcal{M} \mid B_b\varphi) \circledcirc_a U\big) \mid_a B_b\varphi, v \vDash B_a\psi.$$

*Proof.* Indeed, $(\mathcal{M} \mid B_b\varphi) \circledcirc_a \mathsf{U}, v \vDash B_aB_b\bot$ means that there are no $b$-outgoing arrows from any world of $U^a$ in $(\mathcal{M} \mid B_b\varphi) \circledcirc_a \mathsf{U}$. While the public update may affect worlds in $W^{\neg a}$ and, thereby, what $a$ thinks of beliefs of other agents (not $a$ or $b$), the $U^a$ part of the model before the public announcement update is $ab$-bisimilar to itself after the public announcement. □

## 5.7 Conclusions and future Work

In this chapter we introduced a logical framework for modeling local a priori updates dynamically. Such updates are completely private and are initiated by SASO agents upon discovering some system error by reaching inconsistent conclusions.

[Auc07] introduces a difference between subjective and objective approach in epistemic logic: while the latter assumes a global and external perspective on the situation modeled (like in standard distributed systems) the former assumes the local perspective of agent. In the subjective approach, the modeler is involved in the situation. Our approach present an hybrid form of subjective and objective: on the one hand, a priori updates retain the objective model, meaning that they do not change the actual state of affairs. At the same time, an a priori update is conducted by an agent involved in the situation from its viewpoint, and the update extends the initial objective model with a subjective part for the updating agents.

From a dynamic perspective, the differences of the approach exposed in this chapter with the existing literature share a similar vein with the differences exposed in Chapter 4. Having a plausibility preference order on the worlds as in [BS16] (or even on the worlds of the trial model) would imply that these worlds are somehow accessible to the agents in advance. This is however against the spirit of local a priori updates, wherein an agent needs to construct a (trial) model that explains the situation at hand without using a preexisting one. Of course, bounding the search space is needed for applications, but here our goal is to expose the update mechanism in full generality. Similarly, we do not provide the set of possible updates to agents as in the epistemic planning framework, but an extension with a bounded set of elements from which these models can be constructed is for sure an interesting future work direction.

We conclude by drawing a comparison between local a priori updates and global a priori updates of Chapter 4: (i) local a priori updates are invoked autonomously by agents as a self-recovery method, while the global a priori updates have a broader scope depending on the goal of the system designer implementing them; (ii) both updates are modeled using Dynamic Epistemic Logic tools: global a priori updates are formalized through pointed action models that are synthesized from a goal formula, while local a priori updates are built *ad hoc* based on an heuristics that is (partially) inferred from the model that generated the inconsistency; (iii) Both updates stratify the model representing the private nature of a priori belief updates and they both aim at preserving/restoring consistency when possible; (iv) in both cases, the a priori reasoning process is conducted outside

of the logic, even though the trial and error nature of local a priori belief updates simulates part of the process.

We aim at extending the update mechanism to more general action models and developing logics for restricted types of a priori belief updates, which may find applications to belief update synthesis problem. We also plan to address the question of communicating new a priori beliefs (i.e., modeling a posteriori updates about a priori beliefs), which would enable other agents to guide the recovery (or dually, state corruption) of the agent in question.

CHAPTER 6

# Conclusions

In this thesis we have laid the foundations for the inquiry of the role of a priori beliefs in the epistemic analysis of distributed systems. This Chapter summarizes the main contributions and results from the preceding chapters.

Chapter 1 presents the foundational background for this thesis, primarily synthesizing existing work in the areas of distributed systems, epistemic logic, and philosophy. In particular, we:

- Provide a brief overview of distributed systems, which are the field of application of the present thesis. We highlight the differences between traditional distributed systems, heterogeneous distributed systems and self-adaptive and self-organizing (SASO) systems (Section 1.1).
- Introduce the formal framework of standard epistemic logic (Section 1.2), discussing group modalities such as common knowledge (and its role at the model level (Section 1.2.2)) and how epistemic logic is applied to distributed systems in the literature(Section 1.2.3). We then move to the dynamic extension of epistemic logic(Section 1.2.4 and Section 1.2.5), where we consider some logical puzzles and some extensions of dynamic epistemic logic.
- Enucleate the philosophical ideas exposed by Tahko about the relation between a priori and a posteriori(Section 1.3). The application of these ideas to distributed systems and their formalization are the main goal of this thesis.

In Chapter 2 we develop the philosophical framework of a priori beliefs in distributed systems, drawing from the recent philosophical account proposed by Tahko. In this chapter we:

- Adapt the philosophical distinction between a priori and a posteriori to distributed settings, and how they play a role in the design and implementation of distributed systems via a bootstrapping cycle involving both (Section 2.1.2 and Section 2.1.3).

- Introduce the key notions of a priori beliefs and a priori belief update (Section 2.1.2), distinguishing them from their a posteriori counterpart.
- Distinguish between three main kind of agencies (Section 2.1): system designer, processes (or agents) and designer agents, and their respective use of a priori beliefs and a priori reasoning in different distributed systems (traditional, adaptive and self-adaptive).
- Introduce two main kinds of a priori updates, a global version (Section 2.1.2), initiated by the system designer and a local version (Section 2.2.1), initiated by designer agents.
- Establish an informal connection between epistemic logic and the notions of a priori/ a posteriori beliefs (Section 2.1.4), that will be formalized in later chapters.
- Expose the main elements of the a priori reasoning process (Section 2.3), that we propose as a formalization of (some of) the capabilities of agents in SASO systems.

In Chapter 3 we relax the assumption about common a priori knowledge of the protocols, formalizing a logic suitable for the epistemic analysis of heterogeneous distributed systems. In particular we:

- Divide the set of agents into types, each corresponding to a different protocol being executed. Agents do not generally know the type of other agents (Section 3.1).
- Introduce an interpretation function that allows agents to interpret information from agents of a different types.
- Using agents type and interpretation functions, introduce the creed modality, which expresses knowledge of agents conditional to the type of the sender in communication scenarios. Uncertainty about the type of the sender is expressed as a conjunction of such creed formulas (Section 3.1). The creed modality generalizes the hope modality for byzantine agents [KPSF19c].
- Prove soundness and completeness of the resulting logic (Section 3.2). Surprisingly, despite the presence of the interpretation function the creed modality is an **S5** modality if the type of the sender is known (Section 3.3).
- Apply the creed modality to examples drawn from literature, logical puzzles and distributed systems (Section 3.4).

In Chapter 4 we introduce a mechanism to model global a priori belief updates. In particular, we:

- Simulate the effect of a global a priori belief update from a goal formula via a synthesis of a pointed action model that makes the formula true in the resulting model.
- Introduce the notion of pointed update, similar to the product update operation of standard epistemic logic, but with the additional feature of removing worlds that are not accessible from the actual world in the resulting model (Definition 81).

- The goal formula (from a restricted language) represents the target change wanted by the system designer during the update (Definition 84).
- Our mechanism construct an action model in a stratified way (Definition 87): given the actual event 0 (which will correspond to the actual world) and a sink event $-1$ (which will correspond to those worlds representing the status quo) we associate an additional event in between the events 0 and $-1$ for each modal operator present in the formula. In particular, nested operators correspond to stratified agent-clusters in the resulting action model.
- The stratification has a twofold motivation: on the one hand, it preserves beliefs of non-updated agents, by modifying the beliefs of agents specified in the goal formula; and it makes the actual world inaccessible, corresponding to the fallibilistic assumption of a priori updates.
- The update mechanism preserves consistency of agents whenever possible (Theorem 97): the goal formula will not be trivially made true by synthesizing a model where the target agent believes bottom.
- The update does not always increase the size of the model exponentially: in fact, once the model is completely stratified, iterated updates increase the model size only linearly (Theorem 99).
- We prove that our updates are successful w.r.t. to the goal formula (Theorem 92), and that have "minimal side effects" w.r.t. beliefs of agents not involved in the goal formula (Theorem 95), mirroring the effect of a global a priori update.

Finally, in Chapter 5 focuses on local a priori updates, which are initiated locally by agents in case of unexpected behavior. In particular we:

- Introduce the properties of local a priori belief updates, simulating the effect of designer agents in SASO systems that, upon discovering an unexpected error, autonomously perform some self-recovery operation.
- Formalize local a priori belief updates using standard tools of dynamic epistemic logic (Definition 122). In particular, the updating agent tries to reconstruct a consistent view of the system by substituting the current (empty) model with one similar to the previous one but with different a priori assumptions.
- Describe the effects of local a priori updates (Definition 123), formulating a mechanism to perform them (similarly to the product update operation in dynamic epistemic logic).
- Prove that the resulting model preserves the beliefs of the non-updated agents, and that it restores consistency of the updating agents (Theorem 125).
- Show the usefulness of the approach by providing variations of two famous logical puzzles wherein agents recover from erroneous system assumptions by means of a priori updates (Section 5.4).

- While the nature of these updates depends on the specific problem at hand, we provide some heuristics on how the process could be formally described once the initial arena of possibilities is delimited (Section 5.5).

Naturally, this thesis represents only an initial step in the exploration of the novel topic of a priori beliefs in distributed systems. There remains ample opportunity for further research in numerous directions:

- Each of the epistemic frameworks introduced in this thesis can be extended in multiple ways, as discussed at the end of each respective chapter. More generally, a priori updates (at least for restricted settings) could be formalized as update modalities and added directly to the logical language.
- A priori knowledge could be used to explicitly address the complexity of distributed algorithms: it is well known that the more assumptions are made, the simpler the resulting knowledge-based distributed algorithm is. Being able to quantify the weight of these assumptions could be beneficial in exploring the respective overall complexity of such algorithms.
- From a philosophical perspective, the bootstrapping cycle of a priori beliefs calls for questions about creativity of artificial agents, and how they learn and interact in an open world assumption. Further research on the a priori processing could help the development and understanding of some key characteristics of the interplay between rational capacities and creativity in problem solving.

# List of Figures

# List of Tables

# List of Algorithms

# Bibliography


[AGBPBTP20] Yackolley Amoussou-Guenou, Bruno Biais, Maria Potop-Butucaru, and Sara Tucci-Piergiovanni. Rational vs byzantine players in consensus-based blockchains. In *AAMAS'20: Proceedings of the 19th International Conference on Autonomous Agents and Multi Agent Systems*, pages 43–51. IFAAMAS, 2020.

[AGM85] Carlos E. Alchourrón, Peter Gärdenfors, and David Makinson. On the logic of theory change: Partial meet contraction and revision functions. *The Journal of Symbolic Logic*, 50(2):510–530, 1985.

[ALS06] Sameer Ajmani, Barbara Liskov, and Liuba Shrira. Modular software upgrades for distributed systems. In Dave Thomas, editor, *ECOOP 2006 — Object-Oriented Programming: 20th European Conference, Nantes, France, July 3–7, 2006, Proceedings*, pages 452–476. Springer, 2006.

[Art16] Sergei Artemov. Knowing the model. Eprint 1610.04955, arXiv, 2016.

[Art20] Sergei Artemov. Observable models. In Sergei Artemov and Anil Nerode, editors, *Logical Foundations of Computer Science: International Symposium, LFCS 2020, Deerfield Beach, FL, USA, January 4–7, 2020, Proceedings*, volume 11972 of *Lecture Notes in Computer Science*, pages 12–26. Springer, 2020.

[Art22] Sergei Artemov. Towards syntactic epistemic logic. *Fundameta Informaticae*, 186(1–4):45–62, 2022.

[AS13] Guillaume Aucher and François Schwarzentruber. On the complexity of dynamic epistemic logic. In Burkhard C. Schipper, editor, *TARK 2013: Theoretical Aspects of Rationality and Knowledge, Proceedings of the 14th Conference — Chennai, India*, pages 19–28. University of California, Davis, 2013.

[Auc07] Guillaume Aucher. A logical formalism for the subjective approach in a multi-agent setting. In Giacomo Bonanno, James Delgrande, Jérôme Lang, and Hans Rott, editors, *Formal Models of Belief Change in Rational Agents*, volume 7351 of *Dagstuhl Seminar Proceedings (DagSemProc)*, Dagstuhl, Germany, 2007. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[Ba11a] Thomas Bolander and Mikkel Birkegaard Andersen and. Epistemic planning for single- and multi-agent systems. *Journal of Applied Non-Classical Logics*, 21(1):9–34, 2011.

[BA11b] Thomas Bolander and Mikkel Andersen. Epistemic planning for singleand multi-agent systems. *Journal of Applied Non-classical Logics - JANCL*, 21:9–34, 01 2011.

[BBvD+08] Philippe Balbiani, Alexandru Baltag, Hans van Ditmarsch, Andreas Herzig, Tomohiro Hoshi, and Tiago de Lima. 'knowable' as 'known after an announcement'. *Review of Symbolic Logic*, 1(3):305–334, 2008.

[BCR21] Marta Bílková, Zoé Christoff, and Olivier Roy. Revisiting epistemic logic with names. In *Proceedings Eighteenth Conference on Theoretical Aspects of Rationality and Knowledge: Beijing, China, June 25–27, 2021*, volume 335 of *Electronic Proceedings in Theoretical Computer Science*, pages 39–54. Open Publishing Association, 2021.

[BdRV01] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*, volume 53 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2001.

[BG09] Andrew Berns and Sukumar Ghosh. Dissecting self-* properties. In *2009 Third IEEE International Conference on Self-Adaptive and Self-Organizing Systems*, pages 10–19, 2009.

[BHH21] Thomas Bolander, Lasse Dissing Hansen, and Nicolai Herrmann. Del-based epistemic planning for human-robot collaboration: Theory and implementation. In *Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning*, page 120–129. International Joint Conferences on Artificial Intelligence Organization, 2021.

[BM04] Alexandru Baltag and Lawrence S. Moss. Logics for epistemic programs. *Synthese*, 139(2):165–224, 2004.

[BMS98] Alexandru Baltag, Lawrence S. Moss, and Sławomir Solecki. The logic of public announcements, common knowledge, and private suspicions. In Itzhak Gilboa, editor, *Theoretical Aspects of Rationality and Knowledge: Proceedings of the Seventh Conference (TARK 1998)*, pages 43–56. Morgan Kaufmann, 1998.

[Bol17] Thomas Bolander. A gentle introduction to epistemic planning: The del approach. volume 243, pages 1–22. Open Publishing Association, 2017.

[Bon97] Laurence BonJour. *In Defense of Pure Reason: A Rationalist Account of a Priori Justification*. Cambridge University Press, 1997.

[BR16] Alexandru Baltag and Bryan Renne. Dynamic Epistemic Logic. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Winter 2016 edition, 2016.

[BS06] Alexandru Baltag and Sonja Smets. Dynamic belief revision over multi-agent plausibility models. In G. Bonanno, W. Van Der Hoek, and M. Wooldridge, editors, *Proceedings of the 7th Conference on Logic and the Foundations of Game and Decision (LOFT06)*, Proceedings of the 7th Conference on Logic and the Foundations of Game and Decision (LOFT06), pages 11–24. University of Liverpool, 2006.

[BS16] Alexandru Baltag and Sonja Smets. *A Qualitative Theory of Dynamic Interactive Belief Revision*, pages 813–858. Springer International Publishing, Cham, 2016.

[Bur93] Tyler Burge. Content preservation. *The Philosophical Review*, 102(4):457–488, 1993.

[Bur98] Tyler Burge. Computer proof, apriori knowledge, and other minds: the sixth philosophical perspectives lecture. *Philosophical Perspectives*, 32(12: Language, Mind, and Ontology):1–37, 1998.

[Bur10] Tyler Burge. *Origins of Objectivity*. Oxford University Press, 2010.

[Bur20] Tyler Burge. Entitlement: The Basis for Empirical Epistemic Warrant. In *Epistemic Entitlement*. Oxford University Press, 2020.

[BZM14] Ido Ben-Zvi and Yoram Moses. Beyond Lamport's *Happened-before*: On time bounds and the ordering of events in distributed systems. *Journal of the ACM*, 61(2:13), 2014.

[Cas03] Albert Casullo. *A Priori Justification*. Oxford University Press USA, 2003.

[Cas14] Albert Casullo. *Essays on a Priori Knowledge and Justification*. Oup Usa, 2014.

[CDKB11] George Coulouris, Jean Dollimore, Tim Kindberg, and Gordon Blair. *Distributed Systems: Concepts and Design*. Addison-Wesley Publishing Company, USA, 5th edition, 2011.

[CFG24] Giorgio Cignarale, Stephan Felber, and Hugo Rincon Galeana. A sufficient epistemic condition for solving stabilizing agreement. *arXiv*, 2024.

[CGM14] Armando Castañeda, Yannai A. Gonczarowski, and Yoram Moses. Unbeatable consensus. In Fabian Kuhn, editor, *DISC 2014: Distributed Computing*, volume 8784 of *LNCS*, pages 91–106. Springer, 2014.

[Cha20] David J. Chalmers. What is conceptual engineering and what should it be? *Inquiry*, 0(0):1–18, 2020.

[CK24] Giorgio Cignarale and Roman Kuznets. A priori belief updates as a method for agent self-recovery. *Review of Analytic Philosophy*, 4(1):1, 2024.

[CKRS23] Giorgio Cignarale, Roman Kuznets, Hugo Rincon Galeana, and Ulrich Schmid. Logic of communication interpretation: How to not get lost in translation. In Uli Sattler and Martin Suda, editors, *Frontiers of Combining Systems: 14th International Symposium, FroCoS 2023, Prague, Czech Republic, September 20–22, 2023, Proceedings*, volume 14279 of *Lecture Notes in Artificial Intelligence*, pages 119–136. Springer, 2023.

[CnGM22] Armando Castañeda, Yannai A. Gonczarowski, and Yoram Moses. Unbeatable consensus. *Distributed Computing*, 35(2):123–143, April 2022.

[CSTK23] Giorgio Cignarale, Ulrich Schmid, Tuomas Tahko, and Roman Kuznets. The role of a priori belief in the design and analysis of fault-tolerant distributed systems. *Minds and Machines*, 33(2):293–319, June 2023.

[DDvdHKK20] Hans van Ditmarsch, Wiebe van der Hoek, Barteld Kooi, and Louwe B. Kuijer. Arrow update synthesis. *Information and Computation*, 275:104544, 2020.

[Dou25] Igor Douven. Abduction. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Summer 2025 edition, 2025.

[FH87] Ronald Fagin and Joseph Y. Halpern. Belief, awareness, and limited reasoning. *Artificial Intelligence*, 34(1):39–76, 1987.

[FHMV95] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning About Knowledge*. MIT Press, 1995.

[FKS21] Krisztina Fruzsa, Roman Kuznets, and Ulrich Schmid. Fire! In Joseph Halpern and Andrés Perea, editors, *Proceedings of the Eighteenth Conference on Theoretical Aspects of Rationality and Knowledge, Beijing, China, June 25–27, 2021*, volume 335 of *Electronic Proceedings in Theoretical Computer Science*, pages 139–153. Open Publishing Association, 2021.

[FLP85] Michael J. Fischer, Nancy A. Lynch, and M. S. Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the ACM*, 32(2):374–382, April 1985.

[FvdHIK13] Tim French, Wiebe van der Hoek, Petar Iliev, and Barteld Kooi. On the succinctness of some modal logics. *Artificial Intelligence*, 197(C):56–85, 2013.

[Get63] Edmund L. Gettier. Is justified true belief knowledge? *Analysis*, 23(6):121–123, June 1963.

[GG97] Jelle Gerbrandy and Willem Groeneveld. Reasoning about information change. *Journal of Logic, Language, and Information*, 6(2):147–169, April 1997.

[GH93] Adam J. Grove and Joseph Y. Halpern. Naming and identity in epistemic logics part i: The propositional case. *Journal of Logic and Computation*, 3(4):345–378, 08 1993.

[GKTZ12] Adam Groce, Jonathan Katz, Aishwarya Thiruvengadam, and Vassilis Zikas. Byzantine agreement with a rational adversary. In Artur Czumaj, Kurt Mehlhorn, Andrew Pitts, and Roger Wattenhofer, editors, *Automata, Languages, and Programming: 39th International Colloquium, ICALP 2012, Warwick, UK, July 2012, Proceedings, Part II*, volume 7392 of *Lecture Notes in Computer Science*, pages 561–572. Springer, 2012.

[GM18] Guy Goren and Yoram Moses. Silence. In *PODC 2018: Principles of Distributed Computing*, pages 285–294, 2018.

[GR21] Valentin Goranko and Antje Rumberg. Temporal Logic. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Fall 2021 edition, 2021.

[Hal03] Joseph Halpern. Using reasoning about knowledge to analyze distributed systems. *Annual Review of Computer Science*, 2:37–68, 11 2003.

[Hal13] James Hales. Arbitrary action model logic and action model synthesis. In *2013 28th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 253–262, 2013.

[Han22] Sven Ove Hansson. Logic of belief revision. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Spring 2022 edition, 2022.

[Hin62] Jaakko Hintikka. *Knowledge and Belief: An Introduction to the Logic of the Two Notions*. Cornell University Press, 1962.

[HM90] Joseph Y. Halpern and Yoram Moses. Knowledge and common knowledge in a distributed environment. *Journal of the ACM*, 37:549–587, 1990.

[Jai09] Sanjay Jain. Hypothesis spaces for learning. In Adrian Horia Dediu, Armand Mihai Ionescu, and Carlos Martín-Vide, editors, *Language and Automata Theory and Applications*, pages 43–58, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[Kit80] Philip Kitcher. A priori knowledge. *Philosophical Review*, 89(1):3–23, 1980.

[KK07] Igor Kononenko and Matjaz Kukar. *Machine Learning and Data Mining: Introduction to Principles and Algorithms*. Horwood Publishing Limited, 2007.

[KPS+19] Roman Kuznets, Laurent Prosperi, Ulrich Schmid, Krisztina Fruzsa, and Lucas Gréaux. Knowledge in Byzantine message-passing systems I: Framework and the causal cone. Technical Report TUW-260549, TU Wien, 2019.

[KPSF19a] Roman Kuznets, Laurent Prosperi, Ulrich Schmid, and Krisztina Fruzsa. Causality and epistemic reasoning in byzantine multi-agent systems. In Lawrence S. Moss, editor, *TARK 2019: Theoretical Aspects of Rationality and Knowledge*, volume 297 of *EPTCS*, pages 293–312. Open Publishing Association, 2019.

[KPSF19b] Roman Kuznets, Laurent Prosperi, Ulrich Schmid, and Krisztina Fruzsa. Epistemic reasoning with byzantine-faulty agents. In Andreas Herzig and Andrei Popescu, editors, *FroCoS 2019: Frontiers of Combining Systems*, volume 11715 of *LNCS*, pages 259–276. Springer, 2019.

[KPSF19c] Roman Kuznets, Laurent Prosperi, Ulrich Schmid, and Krisztina Fruzsa. Epistemic reasoning with byzantine-faulty agents. In Andreas Herzig and Andrei Popescu, editors, *Frontiers of Combining Systems: 12th International Symposium, FroCoS 2019, London, UK, September 4–6, 2019, Proceedings*, volume 11715 of *Lecture Notes in Artificial Intelligence*, pages 259–276. Springer, 2019.

[KR11a] Barteld Kooi and Bryan Renne. Generalized arrow update logic. In Krzysztof R. Apt, editor, *TARK XIII: Theoretical Aspects of Rationality and Knowledge*, pages 205–211, 2011.

[KR11b] B.P. Kooi and B. Renne. Arrow update logic. *Review of Symbolic Logic*, 4(4):536–559, December 2011.

[Kri72] Saul A. Kripke. Naming and necessity. In *Semantics of Natural Language*, pages 253–355. D. Reidel Publishing Company, 1972.

[Lew69] David K. Lewis. *Convention: A Philosophical Study*. Harvard University Press, 1969.

[LPW11] Benedikt Löwe, Eric Pacuit, and Andreas Witzel. Del planning and some tractable cases. In Hans van Ditmarsch, Jérôme Lang, and Shier Ju, editors, *Logic, Rationality, and Interaction*, pages 179–192, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[LSP82] Leslie Lamport, Robert Shostak, and Marshall Pease. The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, 1982.

[LW13] Fenrong Liu and Yanjing Wang. Reasoning about agent types and the hardest logic puzzle ever. *Minds and Machines*, 23(1):123–161, March 2013.

[Lyn96] Nancy A. Lynch. *Distributed Algorithms*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1996.

[Mos15] Yoram Moses. Relating knowledge and coordinated action: The knowledge of preconditions principle. In R. Ramanujam, editor, *TARK 2015*, pages 231–245, 2015.

[MR02] Yoram Moses and Sergio Rajsbaum. A layered analysis of consensus. *SIAM Journal on Computing*, 31(4):989–1021, January 2002.

[MS93] Yoram Moses and Yoav Shoham. Belief as defeasible knowledge. *Artificial Intelligence*, 64(2):299–321, December 1993.

[MT88] Yoram Moses and Mark R. Tuttle. Programming simultaneous actions using common knowledge. *Algorithmica*, 3:121–169, 1988.

[Pla89] J. A. Plaza. Logics of public communications. In M. L. Emrich, M. S. Pfeifer, M. Hadzikadic, and Z.W. Ras, editors, *Proceedings of the Fourth International Symposium on Methodologies for Intelligent Systems: Poster Session Program*, pages 201–216. Oak Ridge National Laboratory, 1989.

[PS10] Riccardo Pucella and Mehrnoosh Sadrzadeh. A runs-and-systems semantics for logics of announcements. In Giacomo Bonanno, Benedikt Löwe, and Wiebe van der Hoek, editors, *Logic and the Foundations of Game and Decision Theory – LOFT 8*, pages 112–134, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[Put78] Hilary Putnam. There is at least one a priori truth. *Erkenntnis*, 13:153–170, 1978.

[Qui51] W. V. Quine. Two dogmas of empiricism. *Philosophical Review*, 60:20–43, 1951.

[SCF+16] Karla Saur, Joseph Collard, Nate Foster, Arjun Guha, Laurent Vanbever, and Michael Hicks. Safe and flexible controller upgrades for SDNs. In *Symposium on Software Defined Networking (SDN) Research (SOSR'16): March 14–15, 2016, in Santa Clara, CA*. ACM, 2016.

[SKC24] Thomas Schlögl, Roman Kuznets, and Giorgio Cignarale. Consistent update synthesis via privatized beliefs. *arXiv*, 2024.

[Smu78] Raymond M. Smullyan. *What Is the Name of This Book? The Riddle of Dracula and Other Logical Puzzles*. Prentice-Hall, 1978.

[Tah08] Tuomas E. Tahko. A new definition of a priori knowledge: In search of a modal basis. *Metaphysica*, 9:57–68, 2008.

[Tah11] Tuomas E. Tahko. A priori and a posteriori: A bootstrapping relationship. *Metaphysica*, 12:151–164, 2011.

[Tah15] Tuomas E. Tahko. *An Introduction to Metametaphysics*. Cambridge University Press, United Kingdom, 2015.

[Tah20] Tuomas Tahko. A priori or a posteriori? pages 353–363. Abingdon, New York: Routledge, 2020.

[THvM+14] Sven Tomforde, Jörg Hähner, Sebastian von Mammen, Christian Gruhl, Bernhard Sick, and Kurt Geihs. "know thyself" - computational self-reflection in intelligent technical systems. In *2014 IEEE Eighth International Conference on Self-Adaptive and Self-Organizing Systems Workshops*, pages 150–159, 2014.

[vBS15] J. van Benthem and S. Smets. Dynamic logics of belief change. In H. van Ditmarsch, J.Y. Halpern, W. van der Hoek, and B. Kooi, editors, *Handbook of epistemic logic*, pages 313–394. College Publications, 2015.

[vDFK22] Hans van Ditmarsch, Krisztina Fruzsa, and Roman Kuznets. A new hope. In David Fernández-Duque, Alessandra Palmigiano, and Sophie Pinchinat, editors, *Advances in Modal Logic*, volume 14, pages 349–369. College Publications, 2022.

[vDHvdHK15] H. van Ditmarsch, J.Y. Halpern, W. van der Hoek, and B.P. Kooi. *Handbook of Epistemic Logic*. College Publications, 2015.

[vDK15] Hans van Ditmarsch and Barteld Kooi. *One Hundred Prisoners and a Light Bulb*. Springer, 2015.

[vDvdHK07] Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi. *Dynamic Epistemic Logic*. Springer, 2007.

[vvK17] Hans van Ditmarsch, Wiebe van der Hoek, and Louwe B. Kuijer. The undecidability of arbitrary arrow update logic. *Theoretical Computer Science*, 693:1–12, 2017.

[vvKK17] Hans van Ditmarsch, Wiebe van der Hoek, Barteld Kooi, and Louwe B. Kuijer. Arbitrary arrow update logic. *Artificial Intelligence*, 242:80–106, 2017.

[Wil13] Timothy Williamson. How Deep is the Distinction between A Priori and A Posteriori Knowledge? In *The A Priori in Philosophy*. Oxford University Press, 2013.

[WYL13] Ximing Wen, Quan Yu, and Yongmei Liu. Multi-agent epistemic explanatory diagnosis via reasoning about actions. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI)*, pages 27–33, 2013.