FAKULTÄT FÜR !NFORMATIK

Faculty of Informatics

Logics for Coalitional Games of Imperfect Information

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Computational Intelligence

eingereicht von

Georg Seitz
Matrikelnummer 0426015

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung: Ao.Prof. Dipl.-Ing. Dr.techn. Christian Georg Fermüller

Wien, September 19, 2011 ____________________ ____________________

(Unterschrift Verfasser) (Unterschrift Betreuung)

FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Logics for Coalitional Games of Imperfect Information

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Computational Intelligence

by

Georg Seitz
Registration Number 0426015

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Ao.Prof. Dipl.-Ing. Dr.techn. Christian Georg Fermüller

Vienna, September 19, 2011 ______________________ ______________________
(Signature of Author) (Signature of Advisor)

Erklärung zur Verfassung der Arbeit

Georg Seitz
Meiselstraße 8/2/14, 1150 Wien

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

(Ort, Datum) (Unterschrift Verfasser)

Acknowledgements

Now that I finish my studies in Computational Intelligence, I have to admit that I never planned to start them in the first place - it rather happened by accident. For my PhD studies in mathematics I had to take some courses which relate to the topic of my thesis, and I chose some courses on algorithms and complexity theory. In addition, I took “Higher Order Logic” (held by Chris Fermüller), not because I needed it, but because the topic seemed to be nice. It was in this semester that I realized how interesting theoretical computer science (and especially logic) is, and so I attended some more courses. Before I knew it, I had gained half of the ECTS credits necessary for the studies in Computational Intelligence (admittedly, with the help of some credits which I had earned during my undergraduate studies in mathematics), and so I decided to complete them, “just for fun”.

Having explained this, I would first of all like to thank Chris Fermüller, who played an important role in awakening my interest in logic. Next, I want to thank Alois Panholzer, my PhD advisor, for letting me enough freedom to pursue a second course of study. I also want to mention my colleagues (in alphabetical order) Marie-Louise “Mimi” Bruner, Veronika “Veri” Kraus, Benoît “Benni” Loridant, and Johannes “Hannes” Morgenbesser, who never let me run out of sweets and assured that my caffeine level was always high enough. Moreover, I want to thank my parents who financed my studies in mathematics and thus made it possible for me to finance my second studies myself. Last but not least, I’d like to put forward my girlfriend Sabine Palatin, first of all for the proofreading (even though she skipped every paragraph which contains more than two mathematical symbols), but most notably because she always smiles and nods when I try to tell her about another interesting mathematical fact :-).

Abstract

Coalitional games serve as a model for multi-agent systems in which the agents have the possibility to form coalitions in order to achieve certain goals. Alternating-Time Temporal Logic (ATL) is a well established logic for the formalization of such games in the case that the players always have perfect information about the actual state of the game. Regarding games of imperfect information no such "standard" logic seems to exist, but several approaches can be found in recent literature.

In this thesis we discuss and compare some of these approaches with respect to expressivity, complexity, and problems. As it turns out, in the case of memoryless agents some very expressive logics exist which allow one to describe various notions of strategic abilities in coalitional games. Quite on the contrary, the assumption of perfect recall soon leads to the problem that reasonably meaningful logics for such games are undecidable. As our own contribution to the topic, we present an attempt to defuse this problem by approximating such an undecidable logic by decidable ones.

Kurzfassung

Koalitions-Spiele dienen als Modell für Multi-Agenten-Systeme, in denen die Agenten die Möglichkeit haben, Koalitionen zu bilden um bestimmte Ziele zu erreichen. Alternating-Time Temporal Logic (ATL) ist eine etablierte Logik zur Formalisierung solcher Spiele für den Fall dass die Spieler durchgehend perfekte Information über den tatsächlichen Zustand des Spiels besitzen. Für Spiele mit imperfekter Information scheint keine derartige „Standard"-Logik zu existieren, aber in aktueller Literatur finden sich verschiedenste Ansätze.

In dieser Arbeit diskutieren und vergleichen wir einige dieser Ansätze im Hinblick auf Ausdruckskraft, Komplexität und etwaige Probleme. Es stellt sich heraus, dass im Fall von gedächtnislosen Agenten sehr ausdrucksstarke Logiken existieren, die es erlauben, verschiedene Begriffe strategischer Fähigkeiten in Koalitions-Spielen zu beschreiben. Ganz im Gegenteil dazu führt die Annahme von perfektem Gedächtnis schnell zu dem Problem, dass einigermaßen aussagekräftige Logiken für solche Spiele unentscheidbar sind. Als unseren eigenen Beitrag zu der Thematik präsentieren wir einen Ansatz zur Entschärfung dieses Problems durch Approximation solch einer unentscheidbaren Logik durch entscheidbare.

Contents

1 Introduction **1**

2 Game models for coalitional games **7**
2.1 Alternating transition systems 7
2.2 Concurrent game structures 11
2.3 Remarks on the game models 13
2.4 Computations 15
2.5 Strategies 16
2.6 Outcomes 18
2.7 Group knowledge 18

3 Alternating-time temporal logic (ATL) **20**
3.1 Syntax 20
3.2 Semantics 21
3.3 Discussion 23
3.4 Alternative semantics 25
3.5 ATL* 26

4 Logics for coalitional games of imperfect information and perfect recall **27**
4.1 ATEL 27
4.2 ATL$_{iR}$ 31
4.3 ATL$^{D}_{iR}$ 35

5 Logics for coalitional games of imperfect information and imperfect recall **42**
5.1 ATL$_{ir}$ 43
5.2 ATOL 46
5.3 CSL 50

6 Approximating ATL$_{iR}$ **57**
6.1 ATL$_{i\bar{R}}$ 58
6.2 ATL$_{iR^C}$ 64

7 Conclusion **71**

A Complexity classes **73**

B CTL **74**

Notation **76**

List of Figures **78**

Bibliography **79**

CHAPTER 1

Introduction

In this thesis we consider logics which can be used to specify and formalize coalitional games of imperfect information. Rather than jumping right into the details of what these games and logics are, we start with an informal introduction into the basic concepts and ideas we are interested in.

Surely everyone knows what a *game* is, but there are very different forms in which one can formalize them. In game theory one often models games using game trees, which nicely visualize the possible actions of each player in every situation of the game, and whose meaning is often clear at first sight. For example, suppose that one is presented the game tree in Figure 1.1. One easily understands that in this game player 1 first has to pick "rock", "paper" or "scissors", followed by player 2 making the same decision. Moreover, one grasps that the game ends in a tie if both players pick the same item, and that otherwise "rock" beats "scissors", "scissors" beats "paper", and "paper"

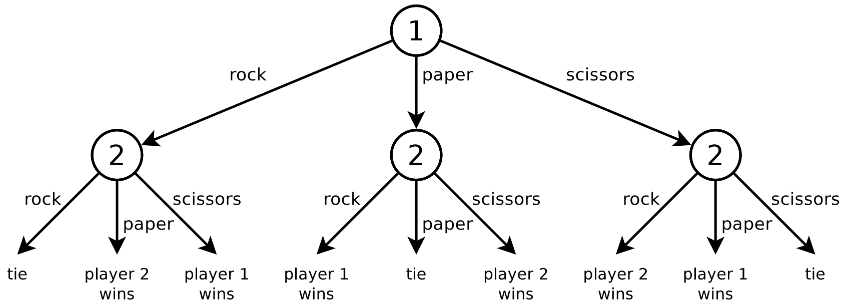


Figure 1.1: A game tree.

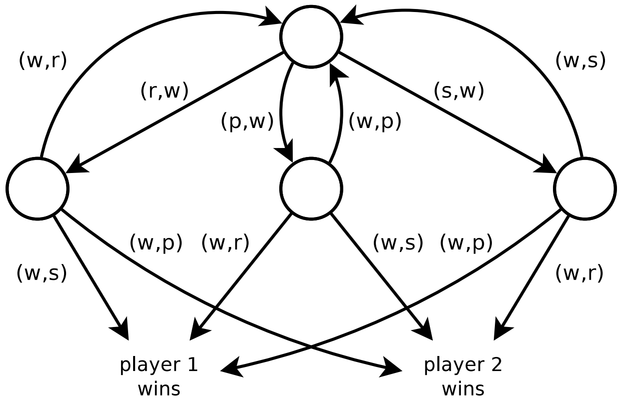


Figure 1.2: An automaton-like game structure.

beats “rock”. It is therefore not hard to realize that this game tree describes a very unfair variant of the well-known game rock-paper-scissors.

However, in this thesis we will describe games not by game trees but rather by automata-like structures. Such a structure for an unfair rock-paper-scissors game similar to the one above could roughly look like Figure 1.2 (rigorous definitions of the used game structures will be given in Chapter 2).

This picture probably needs some more explanation than the tree above. The game starts in the topmost state, in which player 1 has to choose (r)ock, (p)aper or (s)cissors, while player 2 has to (w)ait. After player 1 has made his decision, player 2 is to decide. If he picks the same item, the game moves back into the topmost state, and the game continues. If he picks a different item, either he or player 1 wins the game, depending on his decision.

This example illustrates two nice features of the game structures we will use: Firstly, the players in our game models choose their next moves in each situation simultaneously (albeit in the above example always one of the two players has no move other than “wait” available), which allows us to model a more general class of games. Secondly, possibly never-ending games can often be described by using a game structure with only a finite number of states. Note that this is not possible using a game tree: If we allow the players in the unfair rock-paper-scissors game to continue playing whenever a tie occurs (which is of course usual practice in standard rock-paper-scissors), then we need an infinite tree in order to describe this, in spite of the very simple rules of the game.

We now turn to the question what a *coalitional game* is. Basically every game can be viewed as a coalitional game, by just assuming that the players are allowed to cooperate (i.e., form coalitions) in order to achieve certain goals. Of course, in this context it is

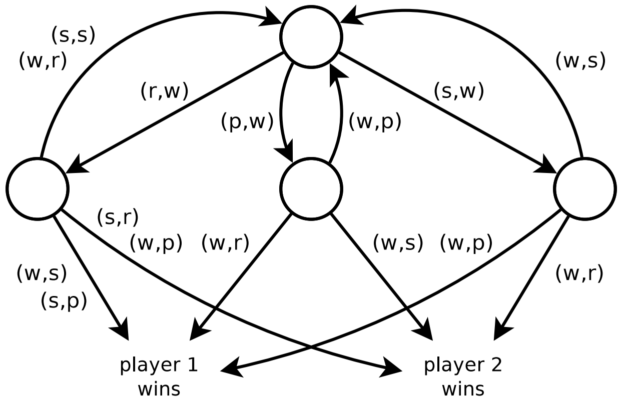


Figure 1.3: Another automaton-like game structure.

always assumed that they are not only allowed to work together but also have some mechanism at hand which enables them to agree on a strategy.

Our main emphasis in this thesis then lies on the question what coalitions of players in a given game can achieve (as opposed to the question what they *want* to achieve). For example, in the unfair rock-paper-scissors game in Figure 1.2 on the preceding page, player 2 has complete control over the outcome of the game. He can win the game if he wants to, or he can let player 1 win. Apart from that, if he likes the game so much that he wants it to run forever, he can simply achieve this by always making the same choice as player 1. However, suppose that we change the rules of this game a bit, such that after having chosen rock, player 1 is allowed to change his mind and switch to scissors simultaneously with player 2 making his decision (this is depicted in Figure 1.3). In this game, none of the two players alone can ensure that he will win, nor that he will lose, nor that the game runs forever. Hence, in order to achieve one of these goals, the players will have to form a coalition.

Having explained what a coalitional game is, we can now clarify what *logics for coalitional games* are. By such logics we mean languages (equipped with appropriate semantics) which can be used to express certain properties of coalitional games, like, e.g., “player 2 on his own cannot ensure that the game runs forever, but together with player 1 he can”. One well-established logic of this kind is “Alternating-Time Temporal Logic” (ATL), which we will formally introduce in Chapter 3. This logic allows one to speak about the powers of coalitions by the use of *cooperation modalities* $\langle\langle A\rangle\rangle$, which have the intuitive meaning of “coalition A has a strategy which ensures ...”. Moreover, ATL contains *temporal operators* $\bigcirc$, $\Box$, and $\mathcal{U}$, which allow one to express the temporal notions “in the next state of the game”, “always”, and “until”, respectively. For example, the above-mentioned example "player 2 on his own cannot ensure that the game runs forever, but together with player 1 he can" could be expressed in ATL by a formula of the form

$$\neg\langle\langle\{2\}\rangle\rangle \, \Box \, game\_runs \wedge \langle\langle\{1,2\}\rangle\rangle \, \Box \, game\_runs.$$

The semantics of ATL rests on an assumption which we have also implicitly made in the above examples, namely that the players have *almost*[1] *perfect information* about the game. That is, we have assumed that

1. all players know exactly how the game structure looks (i.e., they know the "rules" of the game, the possible moves in each situation and their results, the preferences/utilities/winning conditions of the other players, etc.), and, moreover,

2. all players know at each point during the game the exact state of the game (i.e., they know "where they are" in the game structure).

While we will in this thesis always take the first assumption for granted, we will later omit the second one, which leads to *coalitional games of imperfect information*. Note that this is not be confused with games of *incomplete information*, in which the first condition is violated (cf., e.g., [Mye82]).

Imperfect information shows up very naturally in various kinds of games. For example, consider a card game like *bridge*[2]: Of course, we may assume that every bridge player knows the rules of the game, but during the game he certainly does not always know the exact state of the game since he cannot see the other players' cards.

We can modify our rock-paper-scissors example from Figure 1.2 on page 2 a bit by adding imperfect information like it is done in Figure 1.4 on the next page. Here, the dashed line is meant to enclose the states which player 2 cannot distinguish. In this game player 1 has to choose first, but he is allowed to do so behind his back, so that player 2 knows nothing about his decision. Hence, this is an "asynchronous" variant of standard rock-paper-scissors.

When constructing logics which formalize games of imperfect information, the main difficulties are of conceptual nature. One problem that one has to deal with is the question what it should actually mean that a coalition can enforce something. For perfect information games, this is rather simple. Under the assumption of perfect information, a coalition A has a strategy to reach a certain goal iff the players in A know that they

[1] We will clarify later what this "almost" means.

[2] Knowledge of this game is neither presumed nor required before reading on. However, if you are interested, check out `http://en.wikipedia.org/wiki/Contract_bridge`.

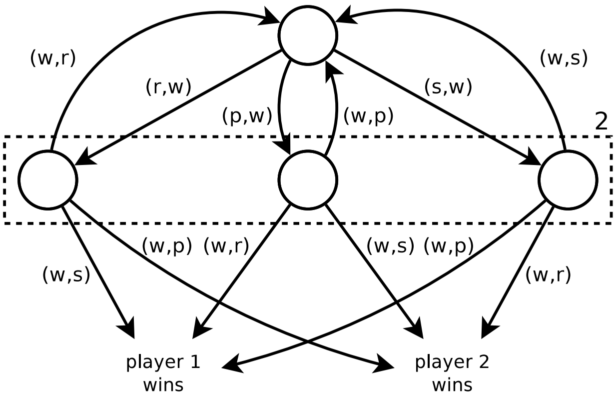


Figure 1.4: An automaton-like game structure with imperfect information. The dashed line encloses the states which player 2 cannot distinguish.

have such a strategy, which is also equivalent to knowing the strategy itself. In contrast, it was pointed out in [Jam04a] and [JÅ07] that when dealing with games of imperfect information, things are more subtle. There it was argued that (at least) four different types of strategic ability can be considered:

(A1) The players in coalition A have the possibility to behave in such a way that the other players cannot avoid that A's goal is achieved. However, this behaviour may have no executable specification due to the lack of information of the players in A, i.e., the players have to "guess" the right moves in each situation.

(A2) The players in A have an executable strategy to reach their goal, but they do not necessarily know that such a strategy exists.

(A3) The players in A know that they have an executable strategy in order to achieve their goal. However, they may not know what this strategy looks like (this is sometimes referred to as a strategy "*de dicto*" [JvdH04, JÅ07]).

(A4) The players in A have an executable strategy to reach their goal, and moreover they know (i.e., can identify) this strategy (this is often called a strategy "*de re*" [JvdH04, JÅ07]).

Moreover, in the latter two types of strategic ability, there are different ways in which the notion "the players in A know" can be understood. It can (at least) mean three different things (we will formalize this in Section 2.7):

- If they communicate and exchange their knowledge, then they know ("distributed knowledge").

- Each of the players in A knows individually ("everyone's knowledge").
- Everyone in A knows, and moreover everyone in A knows that everyone in A knows, and moreover everyone in A knows that everyone in A knows that everyone in A knows, etc. ("common knowledge").

Another question which arises when dealing with games of imperfect information is whether or not the players are assumed to remember the complete history of the game. For ATL this question is inessential, since the resulting semantics for both scenarios coincide (see Section 3.2). But if one generalizes the ATL semantics to imperfect information games in the most direct way, then the memory of the players makes a significant difference. Hence, we will in Chapters 4 and 5 collect and discuss logics for coalitional games of imperfect information both under the assumption of perfect recall and under the assumption of imperfect recall.

As we will see, the assumption of perfect recall soon leads to the problem that reasonably meaningful logics for such games are undecidable. Our own contribution to the topic will thus be an attempt to defuse this problem by at least *approximating* such a logic by decidable ones. This will be done in Chapter 6, where it will also become more clear what we mean by one logic approximating another.

CHAPTER 2

Game models for coalitional games

In this section we introduce the game models which are most widely used in recent research papers on logics for coalitional games (cf., e.g., [AHK98, Gor01, AHK02, vdHW03a, Sch04, Ågo06, JÅ07]) and will thus frequently occur in the subsequent chapters, namely *alternating (epistemic) transition systems* and *concurrent (epistemic) game structures*.

In order to avoid unnecessary case distinctions, we choose our definitions such that the two game models only differ in the way in which the possible moves of the players and the resulting transitions are represented. We want to remark that our definitions thus slightly differ from the ones in recent literature here and there, but this concerns only minor variations which lead to equivalent models (e.g., some authors let the set of players be an arbitrary finite set while others identify the players with natural numbers, a discrepancy which is inessential for our purpose).

2.1 Alternating transition systems

Definition 2.1 (Alternating transition system). *An* alternating transition system *(ATS, for short) is a* 5*-tuple* $\langle \Pi, \Sigma, Q, \pi, \delta \rangle$ *with the following components:*

- Π *is a finite, non-empty set of* propositions.
- Σ *is a finite, non-empty set of* players *(or* agents*). We may always assume that* $\Sigma = \{1, \ldots, k\}$*, where* k *is the number of players. Each set* $A \subseteq \Sigma$ *is called a* coalition.

- *Q is a finite, non-empty set of* states.
- *$\pi : Q \to 2^{\Pi}$ is a* labelling function *which maps each state to the set of propositions that are true in the state.*
- *$\delta : \Sigma \times Q \to 2^{2^Q}$ maps a player and a state to a non-empty set of* choices*, where a choice is simply a set of states. We say that the choices in $\delta(a, q)$ are* enabled for player a in q.

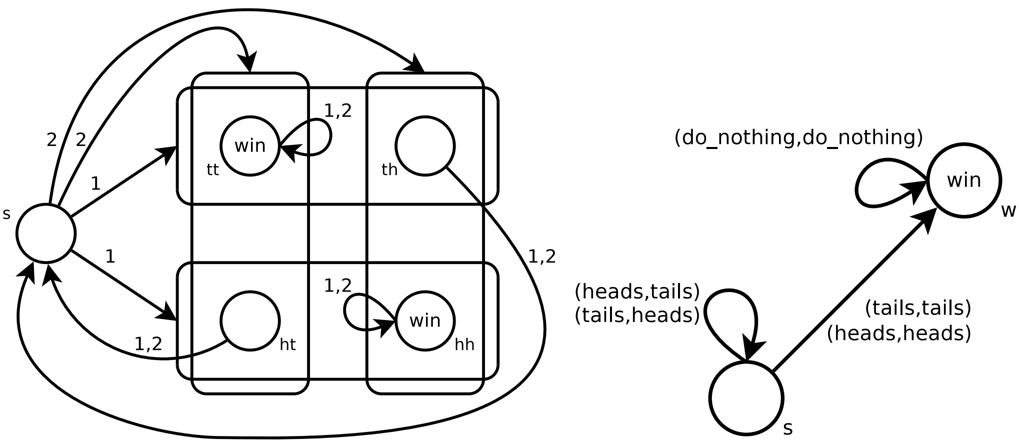
Alternating transition systems are game models which are inspired by the notion of *(α-)effectivity* in game theory: A player a is said to be effective for a set S of game states iff a can pick a move which ensures that the next state of the game will lie in S [Bez98]. This is exactly the meaning of the function δ in the definition above: If $S \in \delta(a, q)$, then a is effective for S in state q.

This should make clear how an ATS models a game: Whenever the system is in state q, each player $a \in \Sigma$ chooses a set $\alpha_a \in \delta(q, a)$. The next state of the game then has to lie in the intersection of all choices. Since we are only interested in *deterministic* games, we will at all times assume that the choices of the players always determine a unique state of the game, i.e., that for every state $q \in Q$ and every set $\{\alpha_a \mid a \in \Sigma\}$ of choices $\alpha_a \in \delta(q, a)$, the intersection $\bigcap_{a\in\Sigma} \alpha_a$ is a singleton, $\bigcap_{a\in\Sigma} \alpha_a = \{q'\}$ (note that this particularly implies that each player has always at least one choice available). We note that this induces in a natural way a deterministic transition function o which defines for each state $q \in Q$ and each tuple $(\alpha_a)_{a\in\Sigma}$ of choices which are enabled in q the next state $q' = o(q, (\alpha_a)_{a\in\Sigma})$ of the system. We will refer to this transition function instead of δ whenever it seems useful.

Alternating transition systems model games sometimes called *almost perfect information games* [vBBW05, Chapter 20]: The players are assumed to have at all times perfect information about the current state of the game, the only uncertainty about the progress of the game arising from the fact that all players choose their next moves simultaneously without knowing what the other players will do.

Example 2.2. Consider the following simple 2-player game: Each player has a coin and simultaneously each of them turns his coin to heads or tails. If their choices coincide then they win the game, otherwise they play another round.

This game can be modelled using an ATS in the following way: The set of players is $\Sigma = \{1, 2\}$ and the set of propositions is $\Pi = \{win\}$. The states of the game are given by $Q = \{ht, hh, th, tt, s\}$, where ht is the state in which player 1 has chosen heads and player 2 has chosen tails, hh is the state in which both players have chosen



(a) ATS (b) CGS

Figure 2.1: ATS and CGS representation of the coin game described in Examples 2.2 and 2.6.

heads, and so on. s is the initial state of the game in which the players have not yet chosen anything. In state s the possible choices for the players are given by $\delta(1, s) = \{\{ht, hh\}, \{tt, th\}\}$ and $\delta(2, s) = \{\{hh, th\}, \{ht, tt\}\}$, respectively. In the states ht and th the players have lost a round of the game, and then both only have the choice of playing again, i.e., $\delta(1, ht) = \delta(1, th) = \delta(2, ht) = \delta(2, th) = \{\{s\}\}$. On the other hand, in states hh and tt the players have won the game, i.e., $\pi(hh) = \pi(tt) = \{win\}$, and in this case the game is over (which is modelled by $\delta(1, hh) = \delta(2, hh) = \{\{hh\}\}$ and $\delta(1, tt) = \delta(2, tt) = \{\{tt\}\}$, i.e., the game loops forever in the respective winning state). A graphical representation of this ATS is given in Figure 2.1a.

Of course, since we are eventually interested in games of imperfect information, we will need structures which allow to model games in which the players do not always know "where they are", i.e., in which they do not always have perfect information about the current state of the game. This can be done by extending alternating transition systems to *alternating epistemic transition systems*:

Definition 2.3 (Alternating epistemic transition system). *An* alternating epistemic transition system *(AETS, for short) is a* 6*-tuple* $\langle \Pi, \Sigma, Q, (\sim_a)_{a\in\Sigma}, \pi, \delta \rangle$ *with the following components:*

- Π, Σ, Q, π, *and* δ *are as in Definition 2.1.*
- *For each* $a \in \Sigma$, $\sim_a \subseteq Q \times Q$ *is an equivalence relation on* Q*. We require that* $\delta(a, q) = \delta(a, q')$ *whenever* $q \sim_a q'$.

The relations $\sim_a$ are called *epistemic accessibility relations* which model the uncertainty of the players about the actual state of the game, i.e., it is understood that $q_1 \sim_a q_2$ iff a cannot distinguish q_1 from q_2 (which also explains why each $\sim_a$ should be an *equivalence* relation).

Note that the requirement that $\delta(a, q) = \delta(a, q')$ whenever $q \sim_a q'$, which was first imposed in [vdHW03b], simply means that players have the same choices in indistinguishable states, which is a reasonable restriction since players should not be able to distinguish principally indistinguishable states by their available choices.

Example 2.4. Think of a 4x4 chessboard on which a single king is positioned. There are two players, the first one of which has control over the vertical position (1-4) of the king, while the second one controls the horizontal position (a-d). At every point of the game, each player is allowed to change the king's position by exactly one row/column or leave it untouched. We assume that the players cannot see the chessboard, and hence player 1 does not know in which of the columns the king is positioned, while player 2 does not know anything about the vertical position.

This can be modelled using an AETS in the following way: The set of states, which represent the position of the king, is given by $Q = \{a1, a2, a3, a4, b1, b2, \ldots, d4\}$. To player 1 all pairs of states are indistinguishable which differ only in the horizontal position, i.e., $xy \sim_1 x'y'$ iff $y = y'$ (for $x, x' \in \{a, b, c, d\}$ and $y, y' \in \{1, 2, 3, 4\}$). Similarly, $xy \sim_2 x'y'$ iff $x = x'$. We let the model include the set of propositions $\Pi = \{p_1, \ldots, p_4, p_a, \ldots, p_d\}$, which encode the information in which row/column each state is: E.g., $\pi(a1) = \{p_a, p_1\}$, $\pi(a2) = \{p_a, p_2\}$, and so on.

The choices of player 1 at each state mirror the fact that he has the control over the vertical position of the king: At each state $x1$ (for $x \in \{a, b, c, d\}$), his possible choices are $\{a1, b1, c1, d1\}$ (leave the king untouched) and $\{a2, b2, c2, d2\}$ (move the king one row up). In row 2 (i.e., in each of the states $x2$) player 1 has three different choices, namely $\{a1, b1, c1, d1\}$ (move the king one row down), $\{a2, b2, c2, d2\}$ (leave the king untouched), and $\{a3, b3, c3, d3\}$ (move the king one row up). Similarly, the choices for player 1 at rows 3 and 4 are defined, and analogously the possible choices for player 2 at each column. It is clear that in this way the choices of both players at each state uniquely determine the next state of the game.

This ATS is depicted in Figure 2.2 on the next page. We omitted the labels of the transitions in this picture, but from the description above it should be clear which pair of choices leads to which transition. ⚁

Assuming a fixed ATS, we will use the following notation: Given a player $a \in \Sigma$ and a state $q \in Q$, we denote the equivalence class of q with respect to $\sim_a$ by $[q]_a$, i.e.,

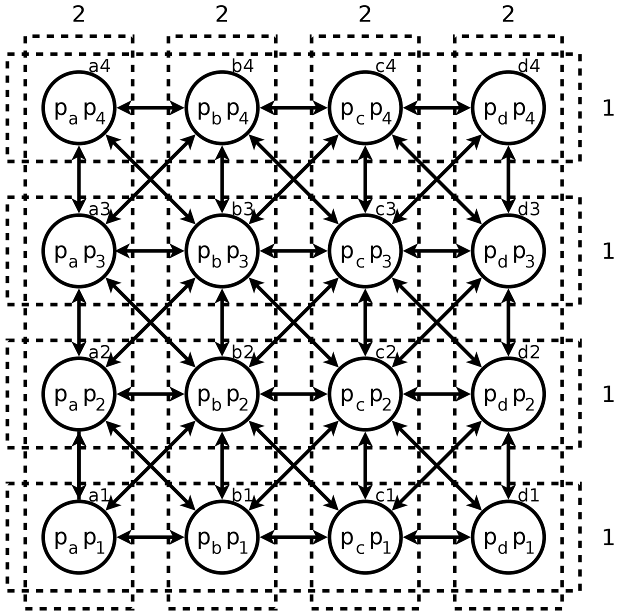


Figure 2.2: AETS for the chess game. Dashed lines enclose indistinguishable states. The possible choices of each player which lead to the given transitions are described in Example 2.4.

$[q]_a := \{q' \in Q \mid q \sim_a q'\}$. We will sometimes call $[q]_a$ an a-*view*, since it is the set of states in which the game could possibly be from a's point of view if the actual state of the game is q. Furthermore, we let Q_a be the set of all a-views, i.e., $Q_a := \{[q]_a \mid q \in Q\}$. Clearly, this is always a partition of Q.

2.2 Concurrent game structures

Definition 2.5 (Concurrent game structure). *A* concurrent game structure *(CGS, for short) is a* 7*-tuple* $\langle \Pi, \Sigma, Q, \pi, Act, d, o \rangle$ *with the following components:*

- Π *is a finite, non-empty set of* propositions.
- Σ *is a finite, non-empty set of* players *(or* agents*). We may always assume that* $\Sigma = \{1, \ldots, k\}$*, where* k *is the number of players. Each set* $A \subseteq \Sigma$ *is called a* coalition.
- Q *is a finite, non-empty set of* states.

- $\pi : Q \to 2^{\Pi}$ *is a* labelling function *which maps each state to the set of propositions that are true in the state.*
- Act *is a finite, non-empty set of* actions *(sometimes also called* choices*).*
- $d : \Sigma \times Q \to 2^{Act} \setminus \{\emptyset\}$ *defines non-empty sets of actions available to the players at each state. We say that the choices in* $d(a, q)$ *are* enabled for player a in state q.
- o *is a deterministic* transition function *which defines for each state* $q \in Q$ *and each tuple* $(\alpha_a)_{a\in\Sigma}$ *of actions which are enabled in* q *(i.e.,* $\alpha_a \in d(a, q)$ *for all* a*), the next state* $q' = o(q, (\alpha_a)_{a\in\Sigma})$ *of the system.*

CGSs model games in a very similar way as ATSs, but in an even more direct manner: Whenever the system is in state q, each player $a \in \Sigma$ chooses an enabled action $\alpha_a \in d(q, a)$. These choices then lead the game into the next state $q' = o(q, (\alpha_a)_{a\in\Sigma})$. Like ATSs, CGSs model games of *almost perfect information*.

Example 2.6. The coin game from Example 2.2 can be represented using a CGS in the following very simple way: As in the ATS representation, we have $\Sigma = \{1, 2\}$ and $\Pi = \{win\}$, but the set of states is now simply given by $Q = \{s, w\}$, where s is the initial state of the game in which the players have not yet made their decision, and w represents the state in which the players have won the game (hence we set $\pi(w) = \{win\}$). The set of possible actions is $Act = \{\text{heads}, \text{tails}, \text{do\_nothing}\}$. In state s the actions heads and tails are enabled for both players, i.e., $d(1, s) = d(2, s) = \{\text{heads}, \text{tails}\}$, while they do not have any choice in the winning state, i.e., $d(1, w) = d(2, w) = \{\text{do\_nothing}\}$. The transition function is given by $o(s, (\text{heads}, \text{heads})) = o(s, (\text{tails}, \text{tails})) = w$, i.e., coinciding choices lead the players into the winning state, and $o(s, (\text{heads}, \text{tails})) = o(s, (\text{tails}, \text{heads})) = s$, i.e., differing choices force the players to play again. If they reach the winning state then the game is over, which is modelled by the fact that in this state the only possible transition is $o(w, (\text{do\_nothing}, \text{do\_nothing})) = w$ (i.e., the game loops in w forever). A graphical representation of this CGS is given in Figure 2.1b on page 9. ⚀

In order to also have a more complex example at hand for future reference, we can also turn the rock-paper-scissors variant from Figure 1.3 (in which player 1 is allowed to change his mind after having chosen rock) into a CGS. We omit the formal definitions in this case, but a graphical representation of this CGS is given in Figure 2.3 on the next page.

Again, we will eventually be interested in modelling imperfect information games. An extension of CGSs for this purpose is the following:

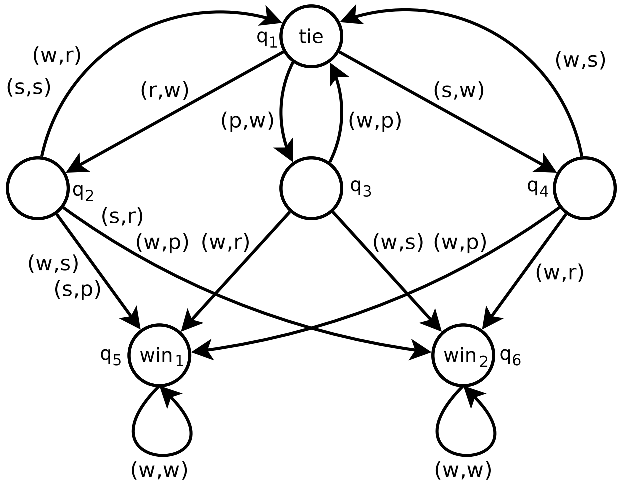


Figure 2.3: CGS for a variant of rock-paper-scissors.

Definition 2.7 (Concurrent epistemic game structure). *A* concurrent epistemic game structure *(CEGS, for short) is a* 8*-tuple* $\langle \Pi, \Sigma, Q, (\sim_a)_{a\in\Sigma}, \pi, Act, d, o\rangle$ *with the following components:*

- Π*,* Σ*,* Q*,* π*,* Act*,* d*, and* o *are as in Definition 2.5.*
- *For each* $a \in \Sigma$*,* $\sim_a \subseteq Q \times Q$ *is an equivalence relation on* Q*. We require that* $d(a, q) = d(a, q')$ *whenever* $q \sim_a q'$*.*

Like in the case of alternating transition systems the relation $\sim_a$ is understood to contain the pairs of states which player a cannot distinguish. For each $q \in Q$ and $a \in \Sigma$ we define $[q]_a$ and Q_a as in Section 2.1. Again, the requirement that $d(a, q) = d(a, q')$ whenever $q \sim_a q'$ ensures that players cannot distinguish states solely by their available choices.

Example 2.8. Figure 2.4 on the following page depicts a CEGS for the asynchronous variant of rock-paper-scissors which we informally described in the introduction (compare Figure 1.4). The dotted lines enclose the states which player 2 cannot distinguish. ⚁

2.3 Remarks on the game models

Note that both game models do not explicitly contain any terminal states: In each state, all players have at least one choice available, and these choices lead the game into

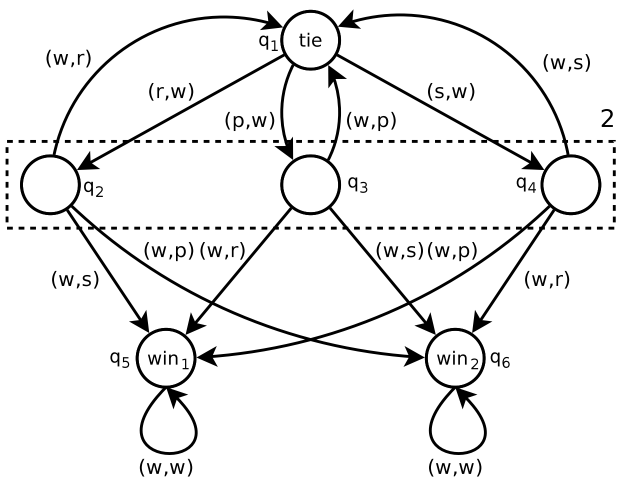


Figure 2.4: CEGS for the asynchronous rock-paper-scissors variant.

a new state, i.e., each game runs forever. This convention is used throughout in the literature on logics for coalitional games, cf. the references in the beginning of this chapter. The deeper reason for this (probably) lies in the fact that otherwise arbitrarily ridiculous statements like “player 1 can ensure that in all subsequent states player 2 will have a whale in his mouth” could become true, namely in states for which there are no subsequent states (which, of course, makes perfect sense from the logical point of view, but is still somewhat irritating).

Of course, as we have already seen in the above examples, it is nevertheless possible to represent the fact that a game is over by introducing states t in which each player has only one choice available (e.g., “do_nothing“), and where the only possible transition leads again to t (i.e., the game “loops” in t forever).

Apart from the lack of terminal states in our game structures, it might seem surprising that there is no built-in notion of “winning” and “payoff/utility” in A(E)TSs and C(E)GSs. This is mainly due to the fact that (as already mentioned in the introduction) we are primarily interested in formalizing what coalitions of players in such games *can* achieve and completely neglect what they *want* to achieve.

But of course, information about winning conditions as well as payoffs/utilities can easily be encoded in the propositions in Π. In [vdHJW05] for example the authors suggested to introduce propositions $u_i \geq v$ (“the utility for player i is at least v”) for all possible utilities v and all players i, which are then assigned to the corresponding states via π. Using this simple idea, they were even able to construct a logic which allows to express game-theoretic concepts such as *Nash Equilibria* and *Pareto Optimality* in

CGS-like game structures. However, since our emphasis lies on completely different questions, we will not go into detail here.

2.3.1 CGSs vs. ATSs

We briefly discuss the relationship between concurrent game structures and alternating transition systems. We concentrate on the perfect information game structures, but the same relationships hold analogously for the epistemic variants.

While ATSs have the nice property that effectivities of players are explicitly represented in the structure, CGSs are often more natural to read due to the fact that the possible actions can bear meaningful names. Also, CGSs are smaller in most cases (compare e.g. the coin game example, Figures 2.1a and 2.1b). This can be made more precise: Clearly, for each ATS $\langle \Pi, \Sigma, Q, \pi, \delta \rangle$ there exists an isomorphic CGS $\langle \Pi, \Sigma, Q, \pi, Act, d, o \rangle$ which can simply be constructed by setting $Act := 2^Q$ and $d := \delta$, and letting o be the transition function which is induced by δ (as described in Section 2.1). However, the inverse does not hold, i.e., there are ATSs for which there is no isomorphic CGS [Jam04b, Section 2.7.4]. Furthermore, given an informal specification of a game, a CGS which satisfies this specification is usually easier to construct than an ATS. This is due to the fact that the set of actions Act and the transition function o in a CGS can basically be chosen arbitrarily, while δ in an ATS has to satisfy the constraint that the choices of all players always determine a unique state.

2.4 Computations

After having defined our game structures (which basically only represent the "rules" of a game), we can now turn to *computations* in those structures (which correspond to games which are being played). In the following we do not need to make any distinction between A(E)TSs and C(E)GSs, since we have defined the transition function o also for A(E)TSs (compare Section 2.1).

Definition 2.9 (Computation). *A* computation *in an A(E)TS or C(E)GS $\mathcal{G}$ is an infinite sequence $\lambda = q_0 q_1 q_2 \dots$ of states of $\mathcal{G}$ with the property that for all $i \geq 0$ there is a transition from q_i to q_{i+1}, i.e., for all $i \geq 0$ there is a tuple $(\alpha_a)_{a \in \Sigma}$ of choices, each choice α_a being enabled for player a in q_i, such that $q_{i+1} = o(q_i, (\alpha_a)_{a \in \Sigma})$. A* finite computation *(or* history*) is a finite prefix of a computation.*

If $\lambda = q_0 q_1 q_2 \dots$ is a computation, we use the following notation to refer to states and prefixes of λ: $\lambda[i] := q_i$ (note that numbering always starts with 0), and $\lambda[0..i] :=$

$q_0 \dots q_i$.

Furthermore, if $\lambda = q_0 \dots q_n$ is a finite computation, we say that λ is a computation of length n, i.e., the length of each finite computation is given by the number of its transitions.

2.5 Strategies

A *strategy* is a conditional plan for a player which uses his information about the history of the game in order to decide which action he should take in each situation. Of course, the amount of information a player a can use in such a strategy depends on

- whether or not a has perfect information about the game, and
- whether or not a is able to remember (his view of) the history of the game.

Following [Sch04] we will thus define four different types of strategies, one for each combination of those options.

In order to define those strategies simultaneously for alternating (epistemic) transition systems and concurrent (epistemic) game structures, we let C be the set of principally possible choices in the respective game structure, i.e., $C = 2^Q \setminus \{\emptyset\}$ in an A(E)TS, and $C = Act$ in a C(E)GS.

Definition 2.10 (IR strategy). *A* perfect information and perfect recall strategy *(for short, an IR strategy) for a player $a \in \Sigma$ in an A(E)TS or C(E)GS is a function $f_a : Q^+ \to C$ with the property that each history $\lambda = q_0 \dots q_n \in Q^+$ is mapped to a choice which is enabled for a in q_n.*

Hence, an IR strategy is a strategy for a player who has perfect information about the game and can remember the complete game history: If player a has decided to follow the strategy f_a, then he will, given that the game went through history λ, choose the action $f_a(\lambda)$.

Remark 2.11. Actually, in game theoretic terms IR strategies do *not* model perfect recall. In game theory, perfect recall usually means that the players do not only remember what they knew in past states of the game, but also the actions they chose in the past [Mye97]. However, in the logics for coalitional games which we will consider in this thesis, this deviation is inessential. ⌟

Definition 2.12 (Ir strategy). *A* perfect information and imperfect recall strategy *(for short, an Ir strategy) for a player $a \in \Sigma$ in an A(E)TS or C(E)GS is a function*

$f_a : Q \to C$ with the property that for all states $q \in Q$ the choice $f_a(q)$ is enabled for player a in q.

Ir strategies allow a player to base his next choice at each point only on the current state of the game, but not on the rest of the game history. That is, if the game went through history $\lambda = q_0 \dots q_n$, then an Ir strategy f_a dictates player a to choose action $f_a(q_n)$. Hence, this is a strategy for a player without memory but with perfect information about the game.

Remark 2.13. It might seem odd that we define IR and Ir strategies also in AETSs and CEGSs, even if in these game structures players are not assumed to have perfect information about the game. It is clear that in AETSs and CEGSs such strategies might not actually be *executable* to the players: It might happen that they assign different choices to two states (or histories) which are indistinguishable to the player, in which case he does not know which choice is the correct one in the current situation.

Nevertheless, it makes sense to consider perfect information strategies also in games of imperfect information: If a player a has a perfect information strategy to achieve a certain goal, then this means that he can at least "guess" the right behaviour in each situation. In particular, this implies that the other players have no strategy to avoid a's goal. ⌟

Definition 2.14 (iR strategy). *An* imperfect information and perfect recall strategy *(for short, an iR strategy) for a player $a \in \Sigma$ in an AETS or CEGS is a function $f_a : Q_a^+ \to C$ with the property that each sequence $[q_0]_a \dots [q_n]_a \in Q_a^+$ is mapped to a choice which is enabled for a in the states of $[q_n]_a$.*

Given the history $\lambda = q_0 \dots q_n$, a player a who has decided to follow the iR strategy f_a will choose the action $f_a([q_0]_a \dots [q_n]_a)$. Hence, this type of strategy allows a to choose his next move depending on the whole history of the game, but taking into account a's uncertainty, i.e., the strategy assigns the same choice to all histories which are indistinguishable to a. This is also referred to as a *uniform* perfect recall strategy [JvdH04, Jam06], and such a strategy is (in contrast to IR strategies) always executable to a player with perfect recall.

Definition 2.15 (ir strategy). *An* imperfect information and imperfect recall strategy *(for short, an ir strategy) for a player $a \in \Sigma$ in an AETS or CEGS is a function $f_a : Q_a \to C$ with the property that for all a-views $[q]_a \in Q_a$ the choice $f_a([q]_a)$ is enabled for player a in the states of $[q]_a$.*

An ir strategy is a uniform strategy for a player without memory, i.e., given a history $\lambda = q_0 \dots q_n$, strategy f_a dictates player a to choose the action $f_a([q_n]_a)$. Due to this uniformity, such a strategy is always executable to a.

Now that we know what a strategy for an individual player is, we define strategies also for coalitions:

Definition 2.16 (Strategy for a coalition). *An XY strategy (where $X \in \{i, I\}$ and $Y \in \{r, R\}$) f_A for a coalition $A \subseteq \Sigma$ is a tuple of XY strategies $(f_a)_{a \in A}$, one for each player.*

2.6 Outcomes

In this section, we will define what an *outcome* of a strategy is. In order to avoid case distinctions we will for this purpose view all four types of strategies as functions $f_a : Q^+ \to C$. In particular, if $\lambda = q_0 q_1 \dots q_n$ is a sequence of states, we set

- if f_a is an Ir strategy: $f_a(\lambda) := f_a(q_n)$,
- if f_a is an iR strategy: $f_a(\lambda) := f_a([q_0]_a[q_1]_a \dots [q_n]_a)$,
- if f_a is an ir strategy: $f_a(\lambda) := f_a([q_n]_a)$.

Definition 2.17 (Outcome). *A computation λ is an* outcome *of a strategy $f_A = (f_a)_{a \in A}$* from state *q if it is a possible computation starting in q under the restriction that the members of A stick to their strategies, i.e., if $\lambda[0] = q$ and for each $i \geq 0$ there exist choices α_a for the players $a \in \Sigma \setminus A$, each α_a being enabled for a in $\lambda[i]$, such that $\lambda[i+1] = o(\lambda[i], (\alpha_a)_{a \in \Sigma})$, where $\alpha_a = f_a(\lambda[0..i])$ for all $a \in A$. We denote the set of outcomes from state q of strategy f_A by $out(q, f_A)$, and if S is a set of states we write $out(S, f_A)$ as a shorthand notation for $\bigcup_{q \in S} out(q, f_A)$.*

Note that for each strategy $f_\Sigma = (f_a)_{a \in \Sigma}$ the set of outcomes $out(q, f_\Sigma)$ is a singleton, i.e., if all players have chosen a strategy then the outcome of the game is uniquely determined.

2.7 Group knowledge

As described above, the knowledge of individual players about the current state of the game in AETSs and CEGSs is modelled by the epistemic accessibility relations $\sim_a$. But of course, when considering such games, one will also be interested in what groups of players know. As already mentioned in the introduction, one can distinguish (at least) three different types of group knowledge. This knowledge will also be represented by epistemic relations, and these can directly be obtained from the individual relations $\sim_a$:

Definition 2.18 (Epistemic relations for group knowledge). *Given a fixed AETS or CEGS, the following relations are derived from the epistemic relations $(\sim_a)_{a\in\Sigma}$:*

- $\sim_{D_A}$ *is, for $A \subseteq \Sigma$, defined as the intersection of the relations $\sim_a$ belonging to the players in A, i.e., $\sim_{D_A} := \bigcap_{a\in A} \sim_a$. This relation describes* distributed knowledge *among the players of A. Note that also $A = \emptyset$ is possible, in which case this definition has to be understood as $\sim_{D_\emptyset} := Q \times Q$.*
- $\sim_{E_A}$ *is, for $A \subseteq \Sigma$, defined as the union of the relations $\sim_a$ belonging to the players in A, i.e., $\sim_{E_A} := \bigcup_{a\in A} \sim_a$. This relation describes* everyone's knowledge *among the players of A. Note that in this case we interpret the empty union as the diagonal of $Q \times Q$, i.e., $\sim_{E_\emptyset} := \{(q, q) \mid q \in Q\}$.*
- $\sim_{C_A}$ *is, for $A \subseteq \Sigma$, defined as the transitive closure of $\sim_{E_A}$. This relation describes* common knowledge *among the members of A.*

We shortly explain what these different forms of group knowledge express: Distributed knowledge among the players of A is the knowledge the players in A have if they communicate and exchange their individual knowledge. That is, if one of the players in A can distinguish q from q', then the group can. Everyone's knowledge among a coalition A is of course the knowledge that every player in A has, i.e., $q_1 \not\sim_{E_A} q_2$ iff all players in A can distinguish q_1 from q_2. Finally, common knowledge among a coalition A describes the notion of "everyone in A knows, and everyone in A knows that everyone in A knows, and everyone in A knows that everyone in A knows that everyone in A knows, etc.".

Note that these different types of group knowledge are not specific to coalitional games, they are used in various kinds of epistemic logics (cf., e.g., [vBBW05, Chapters 18 and 20]).

In the rest of this thesis, we will use the following notation when referring to group knowledge: If $q \in Q$ and $\mathcal{K}_A = C_A$, $\mathcal{K}_A = E_A$, or $\mathcal{K}_A = D_A$, we write $[q]_{\mathcal{K}_A}$ for the set of states which are "$\mathcal{K}_A$-indistinguishable" from q, i.e., $[q]_{\mathcal{K}_A} := \{q' \in Q \mid q' \sim_{\mathcal{K}_A} q\}$. Moreover, if S is a set of states, we write $[S]_{\mathcal{K}_A}$ as a shorthand notation for $\bigcup_{q\in S}[q]_{\mathcal{K}_A}$, i.e., $[S]_{\mathcal{K}_A}$ contains the states of Q which are $\mathcal{K}_A$-indistinguishable from at least one state of S.

CHAPTER 3

Alternating-time temporal logic (ATL)

Since basically all of the logics for imperfect information games which we will consider in this thesis are based on (or at least inspired from) *Alternating-Time Temporal Logic* (ATL), we will in this chapter shortly introduce this logic. As already mentioned in the introduction, ATL is a logic which formalizes coalitional games of almost perfect information, i.e., games where the players are assumed to know in each situation exactly the actual state of the game but choose their next moves simultaneously, which leads to a certain lack of information about the possible progress of the game.

ATL was introduced by Rajeev Alur, Thomas Henzinger and Orna Kupferman in [AHK97, AHK98] where their proposed semantics where based on alternating transitions systems. In later work (e.g. in [AHK02]) they used concurrent game structures instead, probably because the original semantics had some undesirable properties when generalizing them to imperfect information games (as will be discussed in Section 4.1.3). In this chapter we will present the ATL syntax and both semantics and discuss some properties of this logic.

3.1 Syntax

The logic ATL is defined with respect to a finite set Π of propositions and a finite set $\Sigma = \{1, \ldots, k\}$ of players. An ATL formula is one of the following:

- p, where $p \in \Pi$ is a proposition.
- $\neg\varphi$, where φ is an ATL formula.

- $\varphi_1 \vee \varphi_2$, where φ_1 and φ_2 are ATL formulas.
- $\langle\langle A \rangle\rangle \bigcirc \varphi$, where $A \subseteq \Sigma$ is a set of players and φ is an ATL formula.
- $\langle\langle A \rangle\rangle \Box \varphi$, where $A \subseteq \Sigma$ is a set of players and φ is an ATL formula.
- $\langle\langle A \rangle\rangle \varphi_1 \mathcal{U} \varphi_2$, where $A \subseteq \Sigma$ is a set of players and φ_1 and φ_2 are ATL formulas.

The additional Boolean connectives $\wedge$ and $\rightarrow$ as well as the truth constants $\top$ and $\bot$ can of course be defined from $\neg$ and $\vee$ in the usual manner. Note that we usually write $\langle\langle a_1, \dots, a_i \rangle\rangle$ instead of $\langle\langle \{a_1, \dots, a_i\} \rangle\rangle$.

The operator $\langle\langle A \rangle\rangle$ is a *cooperation modality* which acts as a path quantifier: it selects sets of outcomes which result from a strategy of A, i.e., $\langle\langle A \rangle\rangle \varphi$ has the intended meaning "there is a strategy for coalition A which ensures φ". The operators $\bigcirc$ ("next"), $\Box$ ("always") and $\mathcal{U}$ ("until") are *temporal operators*.

Remark 3.1. Note that in ATL formulas each cooperation modality is always directly followed by a temporal operator, and each temporal operator is always directly preceded by a cooperation modality. If one omits this restriction, one obtains the logic ATL* (see Section 3.5). ⌟

3.2 Semantics

We will first present the "new" ATL semantics as defined in [AHK02]: ATL formulas, defined with respect to the set Π of propositions and the set $\Sigma = \{1, \dots, k\}$ of players, are interpreted over the states of a CGS $\mathcal{G} = \langle \Pi, \Sigma, Q, \pi, Act, d, o \rangle$. We write $\mathcal{G}, q \models \varphi$ to indicate that the state q in $\mathcal{G}$ satisfies the formula φ (however, if $\mathcal{G}$ is clear from the context, we just write $q \models \varphi$).

The satisfaction relation $\models$ is defined, for all states q of $\mathcal{G}$, inductively as follows:

(S_p) $q \models p$, for a proposition $p \in \Pi$, iff $p \in \pi(q)$.

($S_\neg$) $q \models \neg\varphi$ iff $q \not\models \varphi$.

($S_\vee$) $q \models \varphi_1 \vee \varphi_2$ iff $q \models \varphi_1$ or $q \models \varphi_2$.

($S_\bigcirc$) $q \models \langle\langle A \rangle\rangle \bigcirc \varphi$ iff there exists an IR strategy $f_A = (f_a)_{a \in A}$ for the players in A, such that all outcomes $\lambda \in out(q, f_A)$ satisfy $\lambda[1] \models \varphi$.

($S_\Box$) $q \models \langle\langle A \rangle\rangle \Box \varphi$ iff there exists an IR strategy $f_A = (f_a)_{a \in A}$ for the players in A, such that for all outcomes $\lambda \in out(q, f_A)$ and all positions $i \geq 0$ we have $\lambda[i] \models \varphi$.

($S_\mathcal{U}$) $q \models \langle\langle A \rangle\rangle \varphi_1 \mathcal{U} \varphi_2$ iff there exists an IR strategy $f_A = (f_a)_{a \in A}$ for the players in A, such that for all outcomes $\lambda \in out(q, f_A)$ there exists a position $i \geq 0$ such that $\lambda[i] \models \varphi_2$ and for all positions $0 \leq j < i$ we have $\lambda[j] \models \varphi_1$.

Remark 3.2. Note that in the case $A = \emptyset$ the cooperation modality acts as a universal quantifier over all possible computations, i.e., e.g., $q \models \langle\langle \emptyset \rangle\rangle \Box \varphi$ iff $\lambda[i] \models \varphi$ for all computations λ with $\lambda[0] = q$ and all positions $i \geq 0$. ⌟

Given a concurrent game structure $\mathcal{G} = \langle \Pi, \Sigma, Q, \pi, Act, d, o \rangle$, a set q in Q, and an ATL formula φ, the *model checking problem for ATL* asks whether $\mathcal{G}, q \models \varphi$. We say that φ is valid in $\mathcal{G}$ iff φ is satisfied in all states of Q. Moreover, we say that φ is valid iff φ is valid in every CGS.

Remark 3.3. In the above semantics it is actually sufficient to consider memoryless strategies, i.e., if one replaces the term "IR strategy" in ($S_\bigcirc$), ($S_\Box$), and ($S_\mathcal{U}$) by "Ir strategy", then this still yields the same satisfaction relation $\models$ [AHK02]. This is an interesting observation, since it means that all properties of coalitional games (or states in such games) which can be specified by ATL formulas do not depend on whether or not the players are assumed to remember the history of the game. ⌟

Example 3.4. As a first simple example, let $\mathcal{G}$ be the concurrent game structure for the coin game depicted in Figure 2.1b on page 9. In this game, we have $s \not\models \langle\langle 1 \rangle\rangle \top \mathcal{U}\, win$ and $s \not\models \langle\langle 2 \rangle\rangle \top \mathcal{U}\, win$, i.e., none of the players alone can ensure that they will eventually win. On the other hand, the players can of course cooperate in order to win the game, i.e., it holds that $s \models \langle\langle 1, 2 \rangle\rangle \bigcirc win$, which is justified by the strategy (f_1, f_2) with $f_1(s) = f_2(s) =$ tails.

In state w, the players have won and the game stays in this state forever. Hence, e.g., it holds that $w \models \langle\langle \emptyset \rangle\rangle \Box win$.

Moreover, the formula $\langle\langle 1, 2 \rangle\rangle \bigcirc \langle\langle \emptyset \rangle\rangle \Box win$ is valid in $\mathcal{G}$, since it is satisfied in both s and w. ⚁

Example 3.5. As a more complex example, let $\mathcal{G}$ be the concurrent game structure for the rock-paper-scissors variant in Figure 2.3 on page 13. In this game, one has e.g. $q_1 \not\models \langle\langle 2 \rangle\rangle \top \mathcal{U}\, win_2$, i.e., player 2 has no winning strategy. However, if the game is in state q_4 (i.e., if player 1 has chosen scissors in the first step), then player 2 has complete control over what happens next: It holds that $q_4 \models \langle\langle 2 \rangle\rangle \bigcirc win_1$, $q_4 \models \langle\langle 2 \rangle\rangle \bigcirc win_2$, and $q_4 \models \langle\langle 2 \rangle\rangle \bigcirc tie$.

If the players cooperate then they can decide to play forever without one of them winning, i.e., the formula $\langle\langle 1, 2 \rangle\rangle \Box (\neg win_1 \wedge \neg win_2)$ is satisfied in all states except q_5 and q_6.

As a last example, consider the formula $(\neg win_1 \wedge \langle\langle 1, 2 \rangle\rangle \bigcirc win_1) \rightarrow \langle\langle 1, 2 \rangle\rangle \bigcirc win_2$, which expresses that if player 1 has not yet won but can win in one step if the players cooperate, then the players can also decide to let player 2 win instead. This formula is valid in $\mathcal{G}$, since it is satisfied in all states of $\mathcal{G}$. ⚁

We will later also evaluate ATL formulas in CEGSs, by just "ignoring" the epistemic relations $(\sim_a)_{a \in \Sigma}$. In order to emphasize the fact that ATL formulas are evaluated using IR strategies (rather than imperfect information strategies, as one might expect in a CEGS), we then mark the cooperation modalities with the letters "IR", i.e., we write $\langle\langle A \rangle\rangle_{IR} \bigcirc \varphi$, $\langle\langle A \rangle\rangle_{IR} \Box \varphi$, and $\langle\langle A \rangle\rangle_{IR} \varphi_1 \mathcal{U} \varphi_2$ instead of $\langle\langle A \rangle\rangle \bigcirc \varphi$, $\langle\langle A \rangle\rangle \Box \varphi$, and $\langle\langle A \rangle\rangle \varphi_1 \mathcal{U} \varphi_2$, respectively. We will then also refer to ATL as ATL$_{IR}$. This convention has been suggested by Schobbens in [Sch04].

3.3 Discussion

As the authors of [AHK02] already mention, ATL is actually a "multi-agent generalization" of CTL (cf. Appendix B for a short overview of this logic). That is, ATL defined with respect to a single player, i.e., $\Sigma = \{1\}$, can be viewed as CTL in the following way: The Kripke structures which occur in the semantics of CTL can just be interpreted as concurrent game structures for a single player. The existential path quantifier E of CTL has then the same meaning as $\langle\langle 1 \rangle\rangle$, and the universal path quantifier A of CTL matches the modality $\langle\langle \emptyset \rangle\rangle$ of ATL. Moreover, the temporal operators $\bigcirc$, $\Box$, and $\mathcal{U}$ of ATL correspond to the operators X, G, and U of CTL.

Due to this relationship it is not surprising that, like CTL (compare Theorem B.1 in Appendix B), ATL can be completely axiomatized:

Theorem 3.6 (Axiomatizability of ATL [GvD06]). *ATL is completely axiomatizable. The axiomatic system consisting of the following axioms and rules of inference is sound and complete with respect to the semantics given in Section 3.2 (i.e., an ATL formula φ is derivable in this system iff φ is valid):*

Axioms ($A, A_1, A_2 \subseteq \Sigma$):

(TAUT) *All (or "enough") propositional tautologies.*

($\bot$) $\neg\langle\langle A\rangle\rangle \bigcirc \bot$.

($\top$) $\langle\langle A\rangle\rangle \bigcirc \top$.

(Σ) $\neg\langle\langle \emptyset\rangle\rangle \bigcirc \neg\varphi \to \langle\langle \Sigma\rangle\rangle \bigcirc \varphi$.

(S) $\langle\langle A_1\rangle\rangle \bigcirc \varphi_1 \wedge \langle\langle A_2\rangle\rangle \bigcirc \varphi_2 \to \langle\langle A_1 \cup A_2\rangle\rangle \bigcirc (\varphi_1 \wedge \varphi_2)$ *for disjoint* A_1 *and* A_2.

(FP$_\Box$) $\langle\langle A\rangle\rangle \Box \varphi \leftrightarrow \varphi \wedge \langle\langle A\rangle\rangle \bigcirc \langle\langle A\rangle\rangle \Box \varphi$.

(GFP$_\Box$) $\langle\langle \emptyset\rangle\rangle \Box (\theta \to (\varphi \wedge \langle\langle A\rangle\rangle \bigcirc \theta)) \to \langle\langle \emptyset\rangle\rangle \Box (\theta \to \langle\langle A\rangle\rangle \Box \varphi)$.

(FP$_\mathcal{U}$) $\langle\langle A\rangle\rangle \varphi_1 \mathcal{U} \varphi_2 \leftrightarrow \varphi_2 \vee (\varphi_1 \wedge \langle\langle A\rangle\rangle \bigcirc \langle\langle A\rangle\rangle \varphi_1 \mathcal{U} \varphi_2)$.

(LFP$_\mathcal{U}$) $\langle\langle \emptyset\rangle\rangle \Box ((\varphi_2 \vee (\varphi_1 \wedge \langle\langle A\rangle\rangle \bigcirc \theta)) \to \theta) \to \langle\langle \emptyset\rangle\rangle \Box (\langle\langle A\rangle\rangle \varphi_1 \mathcal{U} \varphi_2 \to \theta)$.

Rules of inference:

(Modus Ponens) $\frac{\varphi_1 \quad \varphi_1 \to \varphi_2}{\varphi_2}$.

($\langle\langle A\rangle\rangle\bigcirc$-Monotonicity) $\frac{\varphi_1 \to \varphi_2}{\langle\langle A\rangle\rangle \bigcirc \varphi_1 \to \langle\langle A\rangle\rangle \bigcirc \varphi_2}$.

($\langle\langle \emptyset\rangle\rangle \Box$-Necessitation) $\frac{\varphi}{\langle\langle \emptyset\rangle\rangle \Box \varphi}$.

We will not even sketch the very involved proof[1] of the completeness part of this statement. However, the soundness of the given axiomatic system is easy to check: Each of the axioms is valid in every CGS, and the given rules of inference clearly preserve validity.

We shortly comment on some of the axioms: ($\bot$) simply expresses that there is always a next state, i.e., each game runs forever. (Σ) describes that everything which is reachable at all (in one step) can be reached by the grand coalition of all players. Axiom (S) expresses the fact that disjoint coalitions with distinct goals can combine their strategies in order to achieve both goals at the same time. Note that this is clearly not true for coalitions which are not disjoint, since in this case the strategies for different goals might conflict with each other. Axiom (FP$_\Box$) states that $\langle\langle A\rangle\rangle \Box \varphi$ is a fixpoint of the operator $F(X) = \varphi \wedge \langle\langle A\rangle\rangle \bigcirc X$, while (GFP$_\Box$) expresses that it is the greatest fixpoint of $F(X)$. Similarly, (FP$_\mathcal{U}$) means that $\langle\langle A\rangle\rangle \varphi_1 \mathcal{U} \varphi_2$ is a fixpoint of the operator $F(X) = \varphi_2 \vee (\varphi_1 \wedge \langle\langle A\rangle\rangle \bigcirc X)$, while (LFP$_\mathcal{U}$) states that it is the least fixpoint.

A nice feature of ATL which makes it interesting for practical applications is the fact that model checking can be done in polynomial time:

[1] the main part of the 25 pages of [GvD06]

Theorem 3.7 (ATL model checking complexity [AHK02]). *The ATL model checking problem is P-complete, and can be solved in time $O(m \cdot \ell)$ for a game structure with m transitions and an ATL formula of length ℓ.*

Note, however, that this result does not necessarily imply that model checking ATL formulas in CGSs is computationally easy. In fact, it has a well-known catch: In practice, multi-agent models are seldom explicitly given in terms of their states and transitions, instead one uses specification languages which permit very succinct specifications of models. As an example, in [vdHLW06] a language called Simple Reactive Modules Language (SRML) was considered, which can be used in combination with the model checker MOCHA [AHM$^+$98] in order to check ATL formulas in CGSs. This language allows one to represent CGSs by defining sets of propositions which are under the control of the players, and in addition a set of rules which describe the circumstances under which the players are allowed to change the values of single propositions. The states of the system are then implicitly given by all possible combinations of truth values for the propositions, and the possible choices and transitions at each state are described by the rules. Given such a succinct representation of a CGS, it turns out that the model checking problem is EXPTIME-complete in terms of the size of the model description [vdHLW06]. However, in the rest of this thesis all complexity results will be given in terms of the size of an explicitly given model.

As a final remark in this section, we want to mention that independently of [AHK98] and [AHK02], a second logic for coalitional games of almost perfect information (simply called "Coalition Logic") was developed in [Pau02]. But since this logic can simply be viewed as the fragment of ATL in which the only temporal operator is $\bigcirc$, we will not go into detail here.

3.4 Alternative semantics

We now shortly present the original semantics of ATL from [AHK98]: In these semantics, ATL formulas, defined with respect to the set Π of propositions and the set $\Sigma = \{1, \ldots, k\}$ of players, are interpreted over the states of an ATS $\mathcal{G} = \langle \Pi, \Sigma, Q, \pi, \delta \rangle$. Again, if $\mathcal{G}$ is fixed, we write $q \models \varphi$ in order to indicate that state q in $\mathcal{G}$ satisfies the formula φ.

The satisfaction relation $\models$ is defined, for all states q of $\mathcal{G}$, inductively by the same rules as for the "new" semantics, i.e., by rules (S_p)-$(S_{\mathcal{U}})$ from Section 3.2. Note, though, that in this case an IR strategy for a coalition A is of a different type as in the "new" semantics, i.e., it is a tuple $f_A = (f_a)_{a \in A}$, where $f_a : Q^+ \to 2^Q$ (compare Section 2.5) rather than $f_a : Q^+ \to Act$ as for CGSs. However, this distinction does not affect the rules (S_p)-$(S_{\mathcal{U}})$ since we defined strategies and outcomes for ATSs and CGSs in a compatible way (see Definitions 2.10 and 2.17).

Example 3.8. Let $\mathcal{G}$ be the ATS for the coin game from Example 2.2 which is depicted in Figure 2.1a on page 9. Like in the corresponding CGS semantics, we have $s \not\models \langle\langle 1\rangle\rangle \top \,\mathcal{U}\, win$ and $s \not\models \langle\langle 2\rangle\rangle \top \,\mathcal{U}\, win$, i.e., none of the players alone can ensure that they will eventually win. But of course, the players can cooperate in order to win the game, i.e., it holds that $s \models \langle\langle 1, 2\rangle\rangle \bigcirc win$, which is for example justified by any strategy (f_1, f_2) with $f_1(s) = \{hh, ht\}$ and $f_2(s) = \{hh, th\}$.

In contrast to Example 3.4, where the CGS based semantics is used, the formula $\langle\langle 1, 2\rangle\rangle \bigcirc \langle\langle \emptyset\rangle\rangle \Box\, win$ is not valid in $\mathcal{G}$ in this case: The formula is not satisfied in the states th and ht, since the players need two time steps to win from each of these states. ⚁

3.5 ATL*

In [AHK02] a more expressive logic called ATL* was introduced, in which not every occurrence of a temporal operator has to be directly preceded by a cooperation modality. Examples of such formulas are, e.g., $\langle\langle A\rangle\rangle \bigcirc \bigcirc \varphi$, expressing that A can enforce that after two computation steps φ is satisfied, and $\langle\langle A\rangle\rangle \neg(\varphi_1 \,\mathcal{U}\, \varphi_2)$, expressing that A can avoid that φ_1 holds until φ_2 becomes true. This leads to a higher expressiveness, but has two disadvantages. Firstly, one has to make a distinction between "state formulas", which can be evaluated in a single state, and "path formulas", which have to be evaluated on computation paths. Secondly, for this logic the memory of players is important, i.e., one cannot simply modify the semantics of ATL* to use Ir instead of IR strategies without changing the satisfaction of formulas, other than for ATL (compare Remark 3.3). This makes the model checking problem for ATL* computationally harder:

Theorem 3.9 (ATL* model checking complexity [BDJ10]). *Model checking ATL* is 2-EXPTIME-complete in the number of transitions in the model and the length of the formula.*

However, we will only refer to ATL* in some informal remarks, hence we will not go into detail here.

CHAPTER 4

Logics for coalitional games of imperfect information and perfect recall

Since the ATL semantics are in their original formulation based on IR strategies (even though one could also use Ir strategies without changing the satisfaction relation at all, compare Remark 3.3), i.e., it is assumed that the players can remember the whole history of the game, it comes as no surprise that this assumption was also made in the first logics which were introduced as attempts to generalize ATL to imperfect information games.

In the following we will collect and discuss three of these logics.

4.1 ATEL

Alternating-Time Temporal Epistemic Logic (ATEL) was introduced by Wiebe van der Hoek and Michael Wooldridge in [vdHW03a]. This logic extends ATL to games with imperfect information and includes *knowledge modalities* which allow to express what players or groups of players know in each state of the game.

4.1.1 Syntax

Like ATL, ATEL is defined with respect to a finite set Π of propositions and a finite set $\Sigma = \{1, \ldots, k\}$ of players. An ATEL formula is one of the following:

- p, where $p \in \Pi$ is a proposition.

- $\neg\varphi$, where φ is an ATEL formula.
- $\varphi_1 \vee \varphi_2$, where φ_1 and φ_2 are ATEL formulas.
- $\langle\langle A\rangle\rangle \bigcirc \varphi$, $\langle\langle A\rangle\rangle \Box \varphi$, or $\langle\langle A\rangle\rangle \varphi_1 \mathcal{U} \varphi_2$, where $A \subseteq \Sigma$ is a set of players and $\varphi, \varphi_1, \varphi_2$ are ATEL formulas.
- $K_a\varphi$, where $a \in \Sigma$ is a player and φ is an ATEL formula.
- $E_A\varphi$ or $C_A\varphi$, where $A \subseteq \Sigma$ is a set of players and φ is an ATEL formula.

The additional Boolean connectives $\wedge$ and $\rightarrow$ and the truth constants $\top$ and $\bot$ can be defined from $\neg$ and $\vee$ as usual.

The intended meaning of $K_a\varphi$ is "player a knows that φ", whereas the modalities E_A and C_A express everyone's knowledge and common knowledge among the players of A, respectively.

4.1.2 Semantics

ATEL formulas, defined with respect to a set Π of propositions and a set $\Sigma = \{1, \ldots, k\}$ of players, are evaluated in the states of an AETS $\mathcal{G} = \langle \Pi, \Sigma, Q, (\sim_a)_{a\in\Sigma}, \pi, \delta\rangle$. As always, we write $\mathcal{G}, q \models \varphi$ (or just $q \models \varphi$, if $\mathcal{G}$ is clear from the context) to indicate that the state q in $\mathcal{G}$ satisfies the formula φ.

The satisfaction of formulas of the form p ($p \in \Pi$), $\neg\varphi$, $\varphi_1 \vee \varphi_2$, $\langle\langle A\rangle\rangle \bigcirc \varphi$, $\langle\langle A\rangle\rangle \Box \varphi$, and $\langle\langle A\rangle\rangle \varphi_1 \mathcal{U} \varphi_2$ is defined exactly like in the "old" ATL semantics (see Section 3.4). For the remaining cases the satisfaction relation $\models$ is defined as follows (where $[q]_a$, $[q]_{E_A}$, and $[q]_{C_A}$ denote the sets of $\sim_a$/$\sim_{E_A}$/$\sim_{C_A}$-indistinguishable states, cf. Sections 2.1 and 2.7):

(S_{K_a}) $q \models K_a\varphi$ iff all $q' \in [q]_a$ satisfy $q' \models \varphi$.

(S_{E_A}) $q \models E_A\varphi$ iff all $q' \in [q]_{E_A}$ satisfy $q' \models \varphi$.

(S_{C_A}) $q \models C_A\varphi$ iff all $q' \in [q]_{C_A}$ satisfy $q' \models \varphi$.

Note that the modality K_a is actually redundant, since it can be represented both as $E_{\{a\}}$ and as $C_{\{a\}}$.

Example 4.1. Consider the asynchronous rock-paper-scissors variant in Figure 2.4 (note that this Figure actually represents a CEGS, but it can easily be reinterpreted as an AETS since at each state one of the two players has only one action available). In this game, one has of course $q_1 \not\models \langle\langle 1\rangle\rangle \top\, \mathcal{U}\, win_1$, i.e., player 1 has no strategy which ensures that he will eventually win. On the other hand, we have $q_4 \models \langle\langle 2\rangle\rangle \bigcirc win_2$, i.e., in state q_4 player 2 has a winning strategy (namely picking "rock"). It should be noted that even though player 2 even *knows* in q_4 that he has a winning strategy (i.e., it holds that $q_4 \models K_2\langle\langle 2\rangle\rangle \bigcirc win_2$), he can not identify this strategy (because the strategy which achieves win_2 in one step is different for each of the states q_2, q_3, and q_4). ⚀

We give another example in order to illustrate the use of the knowledge modalities for group knowledge:

Example 4.2. Consider the chess game from Example 2.4 which is depicted in Figure 2.2 on page 11. In this game, it holds that $b2 \models K_1 p_2$ and $b2 \models K_2 p_b$. Hence, it also holds that $b2 \models E_{\{1,2\}}(p_b \vee p_2)$, i.e., both players know that the king is either in column b or in row 2. However, this is not common knowledge (i.e., $b2 \not\models C_{\{1,2\}}(p_b \vee p_2)$). In particular, it does not hold that $b2 \models E_{\{1,2\}}E_{\{1,2\}}(p_b \vee p_2)$, because $b2 \sim_{E_{\{1,2\}}} c2$ and $c2 \sim_{E_{\{1,2\}}} c3$ and $c3 \not\models p_b \vee p_2$.

As an example which combines group knowledge and strategic abilities, consider the formula $C_{\{1,2\}}(\neg p_a \wedge \neg p_1 \rightarrow \langle\langle 1,2\rangle\rangle \bigcirc (p_c \wedge p_3))$. This formula expresses that it is common knowledge among the players that, as long as the king is neither in row 1 nor in column a, they can reach position $c3$ in one step if they work together. This formula is valid in this game, since the formula $\neg p_a \wedge \neg p_1 \rightarrow \langle\langle 1,2\rangle\rangle \bigcirc (p_c \wedge p_3)$ is. ⚀

4.1.3 Discussion

At first sight, ATEL seems to be a very promising generalization of ATL to imperfect information games, since it allows one to formalize strategic abilities of coalitions in imperfect information games and in addition the modalities K_a, E_A, and C_A can be used to describe different notions of knowledge of the players in the game. Moreover, the model checking problem for ATEL is not harder than the one for ATL:

Theorem 4.3 (ATEL model checking complexity [vdHW03a]). *The model checking problem for ATEL is P-complete.*

However, as Wojciech Jamroga noted in [Jam04a], ATEL suffers a serious problem: The logic does not formalize the strategic abilities of agents under incomplete information which were probably intended, because IR strategies are used in the semantics. Hence, the players are assumed to be able to make their choices for every history independently, completely neglecting the question whether the players can distinguish these histories or not. For example, in the rock-paper-scissors game in Figure 2.4 on page 14, it holds that $q_1 \models \langle\langle 2\rangle\rangle \top \,\mathcal{U}\, win_2$, which expresses that player 2 has a winning strategy in q_1 and is justified by any strategy f_2 with $f_2(q_1q_2) = p$, $f_2(q_1q_3) = s$, and $f_2(q_1q_4) = r$. But of course, player 2 does *not* have an *executable* winning strategy in this game, because he cannot distinguish the states q_2, q_3, and q_4, on which the above strategy depends.

Hence, it turns out that ATEL formalizes only the strategic ability (A1) (compare the overview of strategic abilities on page 5), i.e., $\langle\langle A\rangle\rangle\varphi$ is true iff the players in A *may happen* to enforce φ, i.e., if they have the possibility to behave in such a way that the other players cannot avoid φ.

The above problem could of course be overcome by using iR instead of IR strategies. But, as Jamroga mentions in [Jam04a], then another problem of ATEL shows up, which derives from the fact that its semantics are based on AETSs: Of course, if one uses iR strategies, one has to impose the restriction that in each AETS the condition $\delta(a, q) = \delta(a, q')$ whenever $q \sim_a q'$ holds, i.e., the same actions must be available in all states which are indistinguishable to a player. However, this is problematic, which we illustrate in the following example:

Example 4.4. Cleopatra's taster is called to the palace in order to check a cake for poison. Of course, he has no other choice than trying a piece. If the cake is poisoned, this will get him two days at the hospital, while otherwise nothing bad will happen.

An AETS for this scenario certainly has to include two states $cake\_poisoned$ and $cake\_ok$, which are indistinguishable to the taster. Furthermore, one will need two states $taster\_ill$ and $taster\_ok$, which represent the outcomes of the tasting. Due to the description of the game and the fact that there is only one player, the choice $\{taster\_ill\}$ certainly has to be enabled in state $cake\_poisoned$, and $\{taster\_ok\}$ has to be available in state $cake\_ok$. If we require the same actions to be available in indistinguishable states, this would mean that also $\{taster\_ill\}$ has to be available in state $cake\_ok$ and $\{taster\_ok\}$ in $cake\_poisoned$, which does not fit the game description.

This does not mean that the given scenario cannot be described by an AETS. Instead of the choices above, one could enable the choice $\{taster\_ok, taster\_ill\}$ both in $cake\_ok$ and in $cake\_poisoned$, and introduce a second player ("fate") which chooses the correct outcome in each of the states. But of course, this is a bit artificial. ⚀

The problem in this example is the fact that in AETSs actions are just given as sets of outcomes, while in many games the same action started in two different states leads to different outcomes.

Jamroga thus concludes that some relation of "subjective unrecognizability" is necessary over the players' choices to tell which decisions will be considered the same in which states. This suggests that CEGSs are more suitable as game models for games of imperfect information, since in these game structures every choice simply bears a "label", and choices with the same label represent the same action from each player's point of view.

4.2 ATL$_{iR}$

Alternating-Time Temporal Logic with Perfect Recall and Imperfect Information, denoted by ATL$_{iR}$, was defined by Schobbens in [Sch04]. It is a very basic generalization of ATL to imperfect information games, which uses exactly the same syntax as ATL. However, in order to mark explicitly the distinction from ATL, the cooperation modalities are marked with the letters "iR" as a subscript, i.e., ATL$_{iR}$ formulas starting with a cooperation modality are of the form $\langle\langle A\rangle\rangle_{iR} \bigcirc \varphi$, $\langle\langle A\rangle\rangle_{iR} \Box \varphi$, and $\langle\langle A\rangle\rangle_{iR} \varphi_1 \mathcal{U} \varphi_2$.

4.2.1 Semantics

ATL$_{iR}$ formulas, defined with respect to the set Π of propositions and the set $\Sigma = \{1, \ldots, k\}$ of players, are interpreted in the states of a concurrent epistemic game structure $\mathcal{G} = \langle \Pi, \Sigma, Q, (\sim_a)_{a\in\Sigma}, \pi, Act, d, o\rangle$. As usual, we write $\mathcal{G}, q \models \varphi$ (or just $q \models \varphi$) to indicate that the state q in $\mathcal{G}$ satisfies the formula φ.

The satisfaction of formulas of the form p ($p \in \Pi$), $\neg\varphi$, and $\varphi_1 \vee \varphi_2$ is defined exactly like in ATL (see (S_p), ($S_\neg$), and ($S_\vee$) in Section 3.2). For the remaining cases the satisfaction relation $\models$ is defined as follows:

($S'_\bigcirc$) $q \models \langle\langle A\rangle\rangle_{iR} \bigcirc \varphi$ iff there exists an iR strategy $f_A = (f_a)_{a\in A}$ for the players in A, such that all outcomes $\lambda \in out([q]_{E_A}, f_A)$ satisfy $\lambda[1] \models \varphi$.

($S'_\Box$) $q \models \langle\langle A\rangle\rangle_{iR} \Box \varphi$ iff there exists an iR strategy $f_A = (f_a)_{a\in A}$ for the players in A, such that for all outcomes $\lambda \in out([q]_{E_A}, f_A)$ and all positions $i \geq 0$ we have $\lambda[i] \models \varphi$.

($S'_\mathcal{U}$) $q \models \langle\langle A\rangle\rangle_{iR} \varphi_1 \mathcal{U} \varphi_2$ iff there exists an iR strategy $f_A = (f_a)_{a\in A}$ for the players in A, such that for all outcomes $\lambda \in out([q]_{E_A}, f_A)$ there exists a position $i \geq 0$ such that $\lambda[i] \models \varphi_2$ and for all positions $0 \leq j < i$ we have $\lambda[j] \models \varphi_1$.

Remark 4.5. Actually, Schobbens defines the semantics of ATL$_{iR}$ as a relation between formulas and *computations* rather than single states. This is due to the fact that in [Sch04] he bases his considerations not on the language of ATL but on the more general ATL* (compare Section 3.5). But since in ATL$_{iR}$ each of the temporal operators $\bigcirc$, $\Box$, and $\mathcal{U}$ is always directly preceded by a cooperation modality $\langle\langle A\rangle\rangle_{iR}$, it suffices to evaluate these formulas in single states. In particular, it holds that $\lambda \models \varphi$ for a computation λ (using Schobben's original definition) iff $\lambda[0] \models \varphi$ (using the above definition) for every ATL$_{iR}$ formula φ. This can easily be shown by induction over the structure of φ. ⌟

We want to point out a mistake in the original definition of the ATL$_{iR}$ semantics from [Sch04]: Instead of using the phrase "... such that for all outcomes $\lambda \in out([q]_{E_A}, f_A)$ ..." as in ($S'_{\bigcirc}$), ($S'_{\Box}$), and ($S'_{\mathcal{U}}$) above, Schobbens employs the formulation "... such that $\forall a \in A$, $\forall q' \sim_a q$, $\forall \lambda \in out(q', f_A)$ ...", which means that the respective satisfaction conditions for $\bigcirc$, $\Box$, and $\mathcal{U}$ always trivially hold if A is empty. This has the counter-intuitive effect that the empty coalition can achieve anything, e.g., $\langle\langle\emptyset\rangle\rangle_{iR} \bot \, \mathcal{U} \, \bot$, which was certainly not intended. This is also the reason why we defined $\sim_{E_\emptyset}$ to be the diagonal of $Q \times Q$ instead of the empty set (see Definition 2.18).

Example 4.6. Consider the asynchronous rock-paper-scissors variant in Figure 2.4. It holds that $q_1 \not\models \langle\langle 2\rangle\rangle_{iR} \top \, \mathcal{U} \, win_2$, which expresses that player 2 has no winning strategy, as it should be in this game. Also player 1 cannot ensure that player 2 will win, i.e., $q_1 \not\models \langle\langle 1\rangle\rangle_{iR} \top \, \mathcal{U} \, win_2$. However, the players can cooperate in order to achieve this, i.e., it holds that $q_1 \models \langle\langle 1, 2\rangle\rangle_{iR} \top \, \mathcal{U} \, win_2$, which is justified e.g. by any iR strategy (f_1, f_2) with $f_1([q_1]_1) = \text{p}$ and $f_2([q_1]_2[q_3]_2) = \text{s}$. ⚁

In the above example we did not actually need the assumption that the players can use the whole game history in order to make their decisions. We now give an example where memory is important:

Example 4.7. Consider the following 1-player game: Along a circular corridor there are 1000 doors, one of which is the entry to a treasure chamber, while all other doors lead to rooms full of deadly traps. Indiana Jones is placed somewhere in the corridor, but he does not know where. He is only allowed to walk in clockwise direction and at some point open one door in order to grab the treasure. The only additional information he gets is the fact that one door is white while all others are black, and that the treasure is behind the 50th door in counter-clockwise direction from the white door. The CEGS for this game is depicted in Figure 4.1 on the following page.

Indiana Jones certainly has an iR strategy which ensures that he will find the treasure (i.e., we have $d_j \models \langle\langle 1\rangle\rangle_{iR} \top \, \mathcal{U} \, rich$, for $0 \leq j \leq 999$): He just has to walk clockwise until he finds the white door, and then he continues and opens the 950th black door he

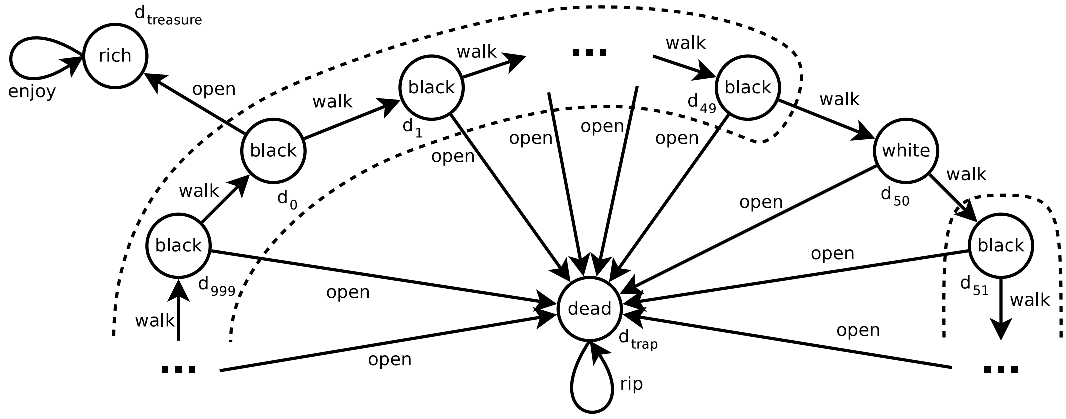


Figure 4.1: CEGS for Indiana Jones' treasure hunt.

encounters, which is achieved by the strategy f_1 defined by

$$f_1(\lambda) = \begin{cases} \text{open}, & \text{if } \lambda = \underbrace{[d_0]_1 \ldots [d_0]_1}_{\text{arbitrary many}}[d_{50}]_1 \underbrace{[d_0]_1 \ldots [d_0]_1}_{\text{950 times}}, \\ \text{enjoy}, & \text{if } \lambda = \lambda' d_{treasure} \text{ for some } \lambda', \\ \text{rip}, & \text{if } \lambda = \lambda' d_{trap} \text{ for some } \lambda', \\ \text{walk}, & \text{else.} \end{cases}$$

Also, no matter what happens, Indiana can always decide to stop walking, i.e., the formula $\langle\langle\emptyset\rangle\rangle_{iR} \Box \langle\langle 1 \rangle\rangle_{iR} \bigcirc (rich \vee dead)$ is valid in this game. However, in most situations he cannot control which of the events ("rich" or "dead") will occur, i.e., the formula $\langle\langle 1 \rangle\rangle_{iR} \bigcirc rich \vee \langle\langle 1 \rangle\rangle_{iR} \bigcirc dead$ is not valid in this game (actually it is only satisfied in the states d_{trap}, $d_{treasure}$, and d_{50}).

4.2.2 Discussion

ATL$_{iR}$ is probably the simplest generalization of ATL to games of imperfect information. The main emphasis lies on expressing the abilities of coalitions under imperfect information and perfect recall. More refined notions like the knowledge of players cannot be expressed.

Note that the definitions of the semantics of $\langle\langle A \rangle\rangle_{iR}$ (see ($S'_\bigcirc$), ($S'_\Box$), and ($S'_\mathcal{U}$) above) use the epistemic relation $\sim_{E_A}$ to specify the set of states from which a strategy has to be successful, i.e., $\langle\langle A \rangle\rangle_{iR}\varphi$ holds in a state iff there exists a uniform perfect recall strategy of which all players in A know that it will ensure φ. Hence, these semantics describe the strategic ability (A4), where “the players in A know” is understood as everyone’s knowledge. Note in particular, that thus an “everyone’s knowledge” operator E_A like in ATEL could also be defined in ATL$_{iR}$ by $E_A\varphi := \langle\langle A\rangle\rangle_{iR}\, \varphi\, \mathcal{U}\, \varphi$.

Even though ATL$_{iR}$ is so simple, it seems that it is actually not simple enough to work with in practice. Since the introduction of ATL it has been conjectured that the model checking problem of ATL$_{iR}$ is undecidable [AHK97, Sch04, JÅ07]. This fact has very recently been proved by Cătălin Dima and Ferucio Laurenţiu Ţiplea:

Theorem 4.8 ([DŢ11]). *The model checking problem for ATL$_{iR}$ is undecidable.*

They prove their result by a rather involved reduction from the non-halting problem, i.e., given a Turing machine M, they construct a 3-player CEGS $G(M)$ containing a state q with the property that the formula $\langle\langle 1, 2\rangle\rangle_{iR}\, \Box\, p$ is satisfied in q iff M does not halt. Hence, they actually prove that the model checking problem is undecidable even for the fragment of ATL$_{iR}$ with 3 players which only consists of formulas of the form $\langle\langle A\rangle\rangle_{iR}\, \Box\, p$.

Remark 4.9. Actually, they prove their result for a slightly modified version of ATL$_{iR}$: When evaluating formulas of the form $\langle\langle A\rangle\rangle_{iR} \bigcirc \varphi$, $\langle\langle A\rangle\rangle_{iR}\, \Box\, \varphi$, and $\langle\langle A\rangle\rangle_{iR}\, \varphi_1\, \mathcal{U}\, \varphi_2$ in a state q, they do not use the relation $\sim_{E_A}$ but only require that an iR strategy $f_A = (f_a)_{a\in A}$ for the players in A exists, such that the respective conditions for $\bigcirc$, $\Box$, and $\mathcal{U}$ are satisfied for all outcomes $\lambda \in out(q, f_A)$. Hence their variant of ATL$_{iR}$ actually describes the strategic ability (A2), i.e., only the existence of an iR strategy is important, regardless of whether the players in A know that it exists.

But of course, model checking an ATL$_{iR}$ formula φ under these modified semantics can easily be reduced to model checking φ under the ATL$_{iR}$ semantics given in Section 4.2.1 by a suitable modification of the considered CEGS. Hence their undecidability result is clearly transferable to the usual definition of ATL$_{iR}$. ⌟

Apart from the fact that ATL$_{iR}$ is undecidable, there is also a little flaw in the semantics: In fact, even though ATL$_{iR}$ allows players to use iR strategies in order to achieve their goals, it actually does not capture all of the abilities of players with perfect recall in imperfect information games. As an example, consider again the rock-paper-scissors game from Figure 2.4. As we have seen in Example 4.6, it holds that $q_1 \models \langle\langle 1, 2\rangle\rangle_{iR}\, \top\, \mathcal{U}\, win_2$, which is justified e.g. by any iR strategy (f_1, f_2) with $f_1([q_1]_1) =$ p and $f_2([q_1]_2[q_3]_2) =$ s. If the players use this strategy, the game will pass through state q_3 and then reach the winning state, and it would thus be logical that also

$q_1 \models \langle\langle 1, 2\rangle\rangle_{iR} \bigcirc \langle\langle 1, 2\rangle\rangle_{iR} \bigcirc win_2$. But, as one easily checks, this does not hold! It thus seems that players in ATL_{iR} may happen to forget their strategies during the game.

The reason for this problem has already been recognized e.g. in [JvdH04]: Unlike states in game trees (cf. the introduction), which contain the whole information about the history of the game, states in CEGSs do not have this property. Thus, evaluating a formula in a single state of a CEGS corresponds to the assumption that the game *starts* in that state, which of course does not fit the idea that the agents are able to use the whole game history in their strategy.

Thus, the authors in [JvdH04] concluded that for games of perfect recall with imperfect information one should evaluate formulas in computations rather than in single states, in contrast to ATL where this is not necessary.

However, since ATL_{iR} in its present form is already undecidable, it probably makes no sense to make its semantics even more complex. Also, at least for the fragment of ATL_{iR} which does not contain any nested cooperation modalities, the above-mentioned problem does not occur.

4.3 $\mathbf{ATL}_{iR}^{D}$

Alternating Time Logic with Knowledge and Communicating Coalitions has been introduced by Dima, Enea, and Guelev in [DEG10]. They use the acronym ATL_{iR}^{D} for this logic in order to emphasize the fact that it uses distributed knowledge, imperfect information, and perfect recall.

ATL_{iR}^{D} has two nice features: Firstly, this logic uses perfect recall not only in the semantics of the cooperation modalities but for the evaluation of all subformulas (i.e., formulas are evaluated in histories rather than in single states), and hence it really reflects the powers of players under the assumption of perfect recall (in contrast to ATL_{iR}, compare Section 4.2.2). Secondly, it uses strategies which allow the players to base their choices on the distributed knowledge among the coalition members, which is to our knowledge a novel idea.

4.3.1 Game arenas

Other than all other logics considered in this thesis, formulas of ATL_{iR}^{D} are interpreted in so-called *game arenas*:

Definition 4.10 (Game arena). *A* game arena *is a game structure which is given by a 7-tuple* $\langle \Sigma, Q, (Act_a)_{a\in\Sigma}, \delta, Q_0, (\Pi_a)_{a\in\Sigma}, \pi\rangle$ *with the following components:*

- Σ *is a finite, non-empty set of agents (or players). We may always assume that* $\Sigma = \{1, \ldots, k\}$*, where* k *is the number of players. Each set* $A \subseteq \Sigma$ *is called a coalition.*
- Q *is a finite, non-empty set of states.*
- *For each* $a \in \Sigma$*,* Act_a *is a set of actions available to player* a*. We set* $Act_A := \prod_{a \in A} Act_a$ *and write* Act *for* Act_Σ*.*
- $d : Q \times Act \to (2^Q \setminus \{\emptyset\})$ *is the* transition relation. *We write* $q \xrightarrow{c} r$ *for transitions* $(q, c, r) \in d$*.*
- $Q_0 \subseteq Q$ *is the set of* initial states.
- *For each* $a \in \Sigma$*,* Π_a *is a set of* atomic propositions *which are* observable to a. *Given* $A \subseteq \Sigma$*, we set* $\Pi_A := \bigcup_{a \in A} \Pi_a$*, and we write* Π *for* Π_Σ*.*
- $\pi : Q \to 2^\Pi$ *is the* state-labelling function. *Moreover, we define* $\pi_A : Q \to 2^{\Pi_A}$*, for* $A \subseteq \Sigma$*, by putting* $\pi_A(q) := \pi(q) \cap \Pi_A$*.*

Remark 4.11. We remark that the above definition deviates from the original one from [DEG10] in terms of the names of the game arena components. Our naming convention is supposed to informally reflect the relation to A(E)TSs and C(E)GSs. ⌟

We want to point out the main differences between AETSs/CEGSs and game arenas: First of all, game arenas do not contain any epistemic relations. The knowledge of the players is implicitly defined by the observability of variables, as we will see later. Moreover, there is no notion of enabled actions in game arenas: Every action of Act_a is available to player a at each point of the game. Finally, game arenas do not necessarily describe *deterministic* games. This is due to the fact that d can be an arbitrary relation rather than a function.

Runs, strategies, and outcome

We now collect some definitions which will be needed in the semantics of ATL_{iR}^{D}:

An element $c \in Act$ is called an *action tuple*. If $c = (c_a)_{a \in \Sigma}$ and $A \subseteq \Sigma$, we denote by $c|_A$ the restriction of c to the players in A, i.e., $c|_A := (c_a)_{a \in A}$.

A *run* is a sequence of transitions $q'_i \xrightarrow{c_i} q''_i$ with the property that $q'_{i+1} = q''_i$. We write $\lambda = (q_{i-1} \xrightarrow{c_i} q_i)_{1 \le i \le n}$ and $\lambda = (q_{i-1} \xrightarrow{c_i} q_i)_{i \ge 1}$ for finite and infinite runs, respectively. The length $|\lambda|$ of a finite run λ is the number of its transitions. A run is *initialized* if $q_0 \in Q_0$.

Remark 4.12. Note that, other than computations in A(E)TS and C(E)GS, runs in game arenas "store" the actions of all users during the game. This definition makes it possible to use the agents' memory about their past actions when evaluating formulas in game arenas. ⌟

Given a run $\lambda = q_0 \xrightarrow{c_1} q_1 \xrightarrow{c_2} \ldots$, we define $\lambda[i] := q_i$, $act(\lambda, i) := c_{i+1}$, and $\lambda[0..i] := q_0 \xrightarrow{c_1} q_1 \xrightarrow{c_2} \ldots \xrightarrow{c_i} q_i$.

As already mentioned, game arenas contain no epistemic relations. The indistinguishability of states (and thus also of runs) is implicitly given by the observability of propositions to the players:

Two finite runs λ and λ' are *indistinguishable to coalition A*, denoted by $\lambda \sim_A \lambda'$, iff

- the lengths of λ and λ' coincide, i.e., $|\lambda| = |\lambda'|$,
- the actions of A in λ and λ' coincide, i.e., $act(\lambda, i)|_A = act(\lambda', i)|_A$ for all $i < |\lambda|$, and
- the observations of A in λ and λ' coincide, i.e., $\pi_A(\lambda[i]) = \pi_A(\lambda'[i])$ for all $i \leq |\lambda|$.

We now introduce the new type of strategy which Dima, Enea, and Guelev use in their logic:

Definition 4.13 (iR^D strategy). *A strategy for a coalition A in a game arena is a function $f_A : (2^{\Pi_A})^+ \to Act_A$. We say that such a strategy is an iR^D strategy for A, since it is an imperfect information and perfect recall strategy which uses the distributed knowledge of the players in A.*

Remark 4.14. In [DEG10], strategies for a coalition A where actually defined to be of the type $f_A : (2^{\Pi_A})^* \to Act_A$. However, requiring a strategy to assign an action to the "empty" history corresponds to the players having a plan for what to do before the game starts, which is certainly not necessary. ⌟

Maybe it is not clear at first sight how iR^D strategies should be interpreted in terms of "real-life" games. We refer to Section 4.3.4 for a short discussion.

Definition 4.15 (Outcome). *An initialized infinite run λ is an outcome of a strategy f_A for coalition A iff λ is* compatible *with f_A, i.e., iff $f_A(\pi_A(\lambda[0]) \ldots \pi_A(\lambda[i])) = act(\lambda, i)|_A$ for all i. We write $out(f_A)$ for the set of all outcomes of f_A.*

4.3.2 Syntax

Like all logics considered so far, ATL_{iR}^{D} is defined with respect to a set Σ of players and a set Π of propositions. In the following, we mark all cooperation modalities with the letters iR as a subscript and D as a superscript in order to make the distinction to other logics clear, even though this was not done in [DEG10]. An ATL_{iR}^{D} formula is one of the following:

- p, where $p \in \Pi$.
- $\neg\varphi$ or $\varphi_1 \wedge \varphi_2$, where φ, φ_1, φ_2 are ATL_{iR}^{D} formulas.
- $\langle\langle A\rangle\rangle_{iR}^{D} \bigcirc \varphi$, $\langle\langle A\rangle\rangle_{iR}^{D} \varphi_1 \,\mathcal{U}\, \varphi_2$, or $\langle\langle A\rangle\rangle_{iR}^{D} \varphi_1 \,\mathcal{W}\, \varphi_2$, where φ, φ_1, φ_2 are ATL_{iR}^{D} formulas and $A \subseteq \Sigma$ is a set of players.
- $K_A\varphi$, where φ is an ATL_{iR}^{D} formula.

Here $\mathcal{W}$ denotes the so-called *weak until* operator (sometimes also called *unless*), which is used in various temporal logics (cf., e.g., [Eme90]). The knowledge modality K_A (which should not be confused with K_a from ATEL) refers to distributed knowledge among the players of A under the additional assumption that the players' knowledge enhances during the game depending on the history. This will become more clear in the next section.

4.3.3 Semantics

ATL_{iR}^{D} formulas, defined with respect to a set Σ of players and a set Π of propositions, are evaluated in a game arena $\mathcal{G} = \langle \Sigma, Q, (Act_a)_{a\in\Sigma}, \delta, Q_0, (\Pi_a)_{a\in\Sigma}, \pi\rangle$ with $\bigcup_{a\in A} \Pi_a = \Pi$. However, formulas are evaluated in infinite runs rather than in single states, in contrast to ATEL and ATL_{iR}.

We write $(\mathcal{G}, \lambda, i) \models \varphi$ in order to indicate that at the i-th position of run λ in the game arena $\mathcal{G}$ the formula φ is satisfied. As usual, if the game arena $\mathcal{G}$ is clear from the context, we do not explicitly refer to it in the notation, i.e., we just write $(\lambda, i) \models \varphi$ in that case.

The satisfaction relation $\models$ for a fixed game arena $\mathcal{G}$ is defined in the following way:

- $(\lambda, i) \models p$ iff $p \in \pi(\lambda[i])$.
- $(\lambda, i) \models \varphi_1 \wedge \varphi_2$ iff $(\lambda, i) \models \varphi_1$ and $(\lambda, i) \models \varphi_2$.
- $(\lambda, i) \models \neg\varphi$ iff $(\lambda, i) \not\models \varphi$.

- $(\lambda, i) \models \langle\langle A \rangle\rangle^D_{iR} \bigcirc \varphi$ iff there exists an iR^D strategy f_A for coalition A such that $(\lambda', i+1) \models \varphi$ for every outcome $\lambda' \in out(f_A)$ which satisfies $\lambda[0..i] \sim_A \lambda'[0..i]$.

- $(\lambda, i) \models \langle\langle A \rangle\rangle^D_{iR} \varphi_1 \mathcal{U} \varphi_2$ iff there exists an iR^D strategy f_A for coalition A such that for every $\lambda' \in out(f_A)$ which satisfies $\lambda[0..i] \sim_A \lambda'[0..i]$ there exists a $j \geq i$ such that $(\lambda', j) \models \varphi_2$ and for all positions $i \leq k < j$ we have $(\lambda', k) \models \varphi_1$.

- $(\lambda, i) \models \langle\langle A \rangle\rangle^D_{iR} \varphi_1 \mathcal{W} \varphi_2$ iff there exists an iR^D strategy f_A for coalition A such that for every $\lambda' \in out(f_A)$ which satisfies $\lambda[0..i] \sim_A \lambda'[0..i]$ one of the following two situations occurs:

    1. there exists a $j \geq i$ such that $(\lambda', j) \models \varphi_2$ and for all positions $i \leq k < j$ we have $(\lambda', k) \models \varphi_1$, or
    2. $(\lambda', k) \models \varphi_1$ for all $k \geq i$.

- $(\lambda, i) \models K_A\varphi$ iff $(\lambda', i) \models \varphi$ for all initialized infinite runs λ' which satisfy $\lambda[0..i] \sim_A \lambda'[0..i]$.

We say that a formula φ is valid in a game arena $\mathcal{G}$ iff $(\mathcal{G}, \lambda, 0) \models \varphi$ for all initialized infinite runs in $\mathcal{G}$.

Given an ATL^D_{iR} formula φ and a game arena $\mathcal{G}$, the model checking problem for ATL^D_{iR} asks whether φ is valid in $\mathcal{G}$.

Remark 4.16. From the above semantics it is clear that the temporal operator $\Box$ ("always") can also be defined in ATL^D_{iR}, via $\langle\langle A \rangle\rangle^D_{iR} \Box \varphi := \langle\langle A \rangle\rangle^D_{iR} \varphi_1 \mathcal{W} \bot$. ⌟

Example 4.17. We can view the chess game from Example 2.4 which is depicted in Figure 2.2 on page 11 as a game arena. In order to do this, we just define the sets Π_1 and Π_2 of variables which are observable to player 1 and 2, respectively, as $\Pi_1 := \{p_1, p_2, p_3, p_4\}$ and $\Pi_2 := \{p_a, p_b, p_c, p_d\}$. The sets of actions for the players are $Act_1 := \{\text{down}, \text{stay}, \text{up}\}$ and $Act_2 := \{\text{left}, \text{stay}, \text{right}\}$, and the transition relation d is defined in correspondence with the game description in Example 2.4 in the obvious way. As the set Q_0 of initial states we can take (e.g.) the set of all states.

If λ is any run in this game which starts in state $a1$ (i.e., $\lambda[0] = a1$), then one has for example $(\lambda, 0) \models K_{\{1\}} \neg \langle\langle 1, 2 \rangle\rangle^D_{iR} \neg p_2 \mathcal{U} p_3$, i.e., player 1 knows that it is not possible to reach the third row without crossing the second one. On the other hand, player 2 does not know this, i.e., $(\lambda, 0) \not\models K_{\{2\}} \neg \langle\langle 1, 2 \rangle\rangle^D_{iR} \neg p_2 \mathcal{U} p_3$, because in his view $\lambda[0]$ could also be $a3$ or $a4$.

As another example, consider $\langle\langle 1,2\rangle\rangle_{iR}^{D} \bigcirc (\langle\langle 1,2\rangle\rangle_{iR}^{D} \Box\, p_1 \vee \langle\langle 1,2\rangle\rangle_{iR}^{D} \bigcirc (p_c \wedge p_4))$. This formula expresses that the two players can cooperate in order to either reach the first row in one step and then stay there forever, or reach state $c4$ in two steps. This formula is valid in this game, since it is satisfied in every run. ⚁

4.3.4 Discussion

Due to the fact that ATL_{iR}^{D} formulas are evaluated in runs rather than in single states, ATL_{iR}^{D} really captures strategic abilities of players with perfect recall in imperfect information games: A formula of the form $\langle\langle A\rangle\rangle_{iR}^{D}\varphi$ is satisfied (at some point of a run), iff there is a uniform strategy of which the players in A know (at this point of the run) that it will ensure φ. This is (almost) the strategic ability (A4), where the term "the players in A know" is understood as distributed knowledge, with the only exception that also the strategy of the players may depend on their distributed knowledge.

We now briefly discuss this new type of strategy which is used in ATL_{iR}^{D}. Like the authors in [DEG10] note, the use of strategies which depend on distributed knowledge requires some care: Pursuing such a strategy requires the coalition members to communicate during the game, which of course enhances the knowledge of the individual players, and it is not entirely clear what the information of each player should look like as soon as the coalition stops to cooperate. In fact, the "correct" behaviour depends on the considered game. We give an informal example:

Example 4.18. Consider a turn-based card game, where in each round every player has to pick a card from his hand which he wants to play. The uncertainty of each player about the actual game state derives from the fact that he can only see his own cards. Then two different scenarios are imaginable:

- First, think of a game in which all cards are distributed among the players before the game starts. If a coalition A in such a game decides to cooperate using a strategy which depends on their distributed knowledge, they have to exchange their knowledge (i.e., tell each other which cards they have). When the coalition later decides to break off their cooperation, then each of the players in A still knows the cards of the other players and can use this information for the rest of the game. In this case, the individual knowledge of each player of A should be replaced by their distributed knowledge, i.e., the players can use the coalition's knowledge even after the coalition split up.
- Now, think of a game in which only a part of the cards is distributed in the beginning, and after each round every player has to pick up a new card from the deck.

If a coalition A in such a game decides to cooperate, they have to keep each other up-to-date on which cards they have as long as they pursue their strategy. In this case, when the coalition splits up and they stop to communicate, the players in A no longer know all cards of their earlier companions. Hence, the knowledge of each player after the breakup will be different from the coalition's distributed knowledge. ⚁

This example shows that there are different ways in which communication can affect the knowledge of players in a game, and such subtle differences can just not be encoded in the game structures we consider.

In [DEG10], the authors circumvent this problem by providing a mechanism which allows a coalition to use a strategy depending on their distributed knowledge without actually having to communicate: Their idea is to equip each coalition with a *virtual supervisor* who collects the members' information and returns the appropriate actions to achieve their goal.

While this interpretation may seem a bit artificial, ATL_{iR}^{D} has one important advantage over ATL_{iR}:

Theorem 4.19 (ATL_{iR}^{D} model checking [DEG10]). *The model checking problem for ATL_{iR}^{D} is decidable.*

Dima, Enea, and Guelev prove this result by presenting an algorithm which constructs a sequence of game arenas and uses certain tree automata in order to check subformulas and replace them with new propositions. Since this algorithm is very involved, we will not go into detail at this point. However, we will later use Theorem 4.19 in order to prove the decidability of the logic ATL_{iR}^{C} which we will construct in order to approximate ATL_{iR} (see Section 6.2).

CHAPTER 5

Logics for coalitional games of imperfect information and imperfect recall

As we have seen in the previous chapter, it seems to be difficult to come up with a logic for games of imperfect information and perfect recall which has both a meaningful interpretation and a decidable model checking problem. Hence, it is natural to consider instead games in which it is assumed that the players have no memory at all. One might argue that this is an unrealistic assumption for many applications, but still it is not so far-fetched to consider such logics:

First of all, in practice (e.g., when modelling multi-agent systems) the players mostly will have bounded memory. As noted in [JvdH04], in cases where a bound on the memory is known, one can actually model the whole system as a memoryless game, by encoding the memory of the players directly in the states of the system.

As Schobbens remarks in [Sch04], another field of application of imperfect recall is the *approximation* of imperfect information games with perfect recall: Clearly, if a coalition A has a memoryless strategy to achieve a certain goal, then A can also achieve this goal under the assumption of perfect recall. Hence, considering the powers of coalitions without memory leads to a lower bound of what coalitions with perfect recall can achieve (we will pursue this idea a bit further in Chapter 6).

Due to the above arguments it certainly makes sense to consider logics for coalitional games with memoryless players. In the following sections, we thus collect and discuss three of such logics.

5.1 ATL$_{ir}$

In [Sch04], Schobbens defines *Alternating-Time Temporal Logic with Imperfect Recall and Imperfect Information* (ATL$_{ir}$). Like the perfect recall variant ATL$_{iR}$ (see Section 4.2), this logic uses exactly the same syntax as ATL. However, in order to avoid confusion, the cooperation modalities $\langle\langle A\rangle\rangle$ are marked with the letters "ir" as a subscript, i.e., ATL$_{ir}$ formulas starting with a cooperation modality are of the form $\langle\langle A\rangle\rangle_{ir} \bigcirc \varphi$, $\langle\langle A\rangle\rangle_{ir} \Box \varphi$, and $\langle\langle A\rangle\rangle_{ir} \varphi_1 \mathcal{U} \varphi_2$.

5.1.1 Semantics

ATL$_{ir}$ formulas, defined with respect to the set Π of propositions and the set $\Sigma = \{1, \ldots, k\}$ of players, are interpreted in the states of a concurrent epistemic game structure $\mathcal{G} = \langle \Pi, \Sigma, Q, (\sim_a)_{a\in\Sigma}, \pi, Act, d, o\rangle$. We write $\mathcal{G}, q \models \varphi$ to indicate that the state q in $\mathcal{G}$ satisfies the formula φ, or just $q \models \varphi$ if $\mathcal{G}$ is clear from the context.

The satisfaction of formulas of the form p ($p \in \Pi$), $\neg\varphi$, and $\varphi_1 \vee \varphi_2$ is defined exactly like in ATL (see (S_p), ($S_\neg$), and ($S_\vee$) in Section 3.2). For the remaining cases the satisfaction relation $\models$ is defined as follows:

($S'_\bigcirc$) $q \models \langle\langle A\rangle\rangle_{ir} \bigcirc \varphi$ iff there exists an ir strategy $f_A = (f_a)_{a\in A}$ for the players in A, such that all outcomes $\lambda \in out([q]_{E_A}, f_A)$ satisfy $\lambda[1] \models \varphi$.

($S'_\Box$) $q \models \langle\langle A\rangle\rangle_{ir} \Box \varphi$ iff there exists an ir strategy $f_A = (f_a)_{a\in A}$ for the players in A, such that for all outcomes $\lambda \in out([q]_{E_A}, f_A)$ and all positions $i \geq 0$ we have $\lambda[i] \models \varphi$.

($S'_\mathcal{U}$) $q \models \langle\langle A\rangle\rangle_{ir} \varphi_1 \mathcal{U} \varphi_2$ iff there exists an ir strategy $f_A = (f_a)_{a\in A}$ for the players in A, such that for all outcomes $\lambda \in out([q]_{E_A}, f_A)$ there exists a position $i \geq 0$ such that $\lambda[i] \models \varphi_2$ and for all positions $0 \leq j < i$ we have $\lambda[j] \models \varphi_1$.

Remark 5.1. Like in the case of ATL$_{iR}$, Schobbens actually defines the satisfaction of ATL$_{iR}$ formulas for computations rather than for single states, since his work is motivated by the more general ATL*. But since in ATL$_{ir}$ each of the temporal operators is always directly preceded by a cooperation modality, it is sufficient to evaluate these formulas in the state $\lambda[0]$ of any computation λ (compare Remark 4.5). ⌟

Example 5.2. Consider the asynchronous rock-paper-scissors variant depicted in Figure 2.4 on page 14. It holds that $q_1 \not\models \langle\langle 2\rangle\rangle_{ir} \top \mathcal{U}\, win_2$, which expresses that player 2 has no winning strategy, as it should be in this game. Also player 1 cannot ensure that player 2 will win, i.e., $q_1 \not\models \langle\langle 1\rangle\rangle_{ir} \top \mathcal{U}\, win_2$. However, the players can cooperate in order to achieve this, i.e., it holds that $q_1 \models \langle\langle 1, 2\rangle\rangle_{ir} \top \mathcal{U}\, win_2$, which is justified e.g. by any ir strategy (f_1, f_2) with $f_1([q_1]_1) =$ p and $f_2([q_3]_2) =$ s. ⚁

By the way, the formulas above are exactly the same as in Example 4.6 in the context of ATL$_{iR}$. As expected, the results do not depend on whether or not the players are supposed to remember the history of the game. This is of course different for Indiana Jones' treasure hunt (see Figure 4.1 on page 33):

Example 5.3. Indiana Jones has no ir strategy to find the treasure from any of the states d_j, i.e., it holds that $d_j \models \neg\langle\langle 1\rangle\rangle_{ir} \top \mathcal{U}\, rich$ for $0 \leq j \leq 999$. But of course, Indiana's lack of memory does not prevent him from stopping his treasure hunt whenever he likes, i.e., the formula $\langle\langle \emptyset\rangle\rangle_{ir} \Box \langle\langle 1\rangle\rangle_{ir} \bigcirc (rich \vee dead)$ is valid in this game. However, in most situations he cannot control which of the two events ("rich" or "dead") will occur, i.e., the formula $\langle\langle 1\rangle\rangle_{ir} \bigcirc rich \vee \langle\langle 1\rangle\rangle_{ir} \bigcirc dead$ is not valid in this game. ⚁

5.1.2 Discussion

ATL$_{ir}$ is probably the simplest generalization of ATL to games of imperfect information under the assumption of imperfect recall. Like in ATL$_{iR}$, the main emphasis lies on expressing the abilities of coalitions under imperfect information and imperfect recall. It contains no modalities which allow one to express what players know in certain states, the only notion of knowledge implicitly appears in the semantics of the cooperation modalities (and hence E_A could be defined by $E_A\varphi := \langle\langle A\rangle\rangle_{ir}\, \varphi \mathcal{U}\, \varphi$, like in ATL$_{iR}$).

Due to the use of the relation $\sim_{E_A}$ in the semantics of $\langle\langle A\rangle\rangle_{ir}$, a formula $\langle\langle A\rangle\rangle_{ir}\varphi$ is satisfied iff a uniform (i.e., executable) strategy exists of which all players in A know that it will enforce φ. Hence, ATL$_{iR}$ expresses the strategic ability (A4) of players without memory in imperfect information games, where "the players in A know" is understood as everyone's knowledge.

Schobbens' motivation for introducing ATL$_{ir}$ actually was its simplicity. In [Sch04] he argues that incomplete information is usually more complex than complete information, but imperfect recall, although restricting the capabilities of agents even more, usually reduces complexity.

In the same paper he also proves that ATL$_{ir}$ is Δ_2^P-easy by presenting an algorithm which polynomially often guesses a strategy, trims the model according to this strategy, and then calls a CTL model checking procedure (which runs in deterministic polynomial time (cf. Theorem B.2 in Appendix B). Schobbens also conjectured that the model checking problem for ATL$_{ir}$ is Δ_2^P-hard. This fact was later proven by Wojciech Jamroga and Jürgen Dix:

Theorem 5.4 (ATL$_{ir}$ model checking complexity [JD08]). *Model checking ATL$_{ir}$ is Δ_2^P-complete in the number of transitions in the model and the length of the formula.*

They prove this result by a reduction from the so-called $SNSAT$ problem. However, we will not go into detail here.

The above arguments show that ATL$_{ir}$ both describes a meaningful type of strategic ability and is not too hard in terms of model checking, which makes it usable for practical applications in contrast to its perfect recall counterpart ATL$_{iR}$.

However, we want to mention a fact about ATL$_{ir}$ which is maybe not clear at first sight: Even if a formula $\langle\langle A\rangle\rangle_{ir}\varphi$ is satisfied in a state q, this does not necessarily imply that coalition A will be able to ensure φ without some form of communication. This is shown in the following example.

Example 5.5. Consider the following variant of the well-known *Two Generals Paradox* [Gra78]: Two armies, each led by a general, are preparing to attack a city. The armies are camping on opposite sides of the city and have no possibility to communicate. Each of the generals sends a spy to the city in order to find a weakness in the enemy's defense. If one of the spies gathers enough information, then the armies will be able to conquer the city if they attack together the next morning. If only one army attacks, it will lose the fight regardless of its information. This scenario is depicted in Figure 5.1 on the following page: ss is the state in which none of the spies gathers enough information, Ss is the state in which the first spy is successful while the other is not, and so on. Since the generals cannot communicate, each of them only knows whether his own spy has gathered enough information, and hence, e.g., the first general cannot distinguish the states Ss and SS.

Now note that in state SS the ATL$_{ir}$ formula $\langle\langle 1,2\rangle\rangle_{ir} \bigcirc win$ is satisfied. Still, the generals cannot be sure that their attack in SS will succeed: As far as the first general knows, he might be in state Ss, in which case the second general would deem state ss possible, from which an attack would lead to a certain loss. Hence the first general cannot be sure that the second one will attack, and thus he himself will not attack in order to prevent his army from damage. The same holds analogously for the second general. ⚁

The problem in this example is of course the fact that, while there is a strategy of which both coalition members know that it ensures φ, each of them doesn't know

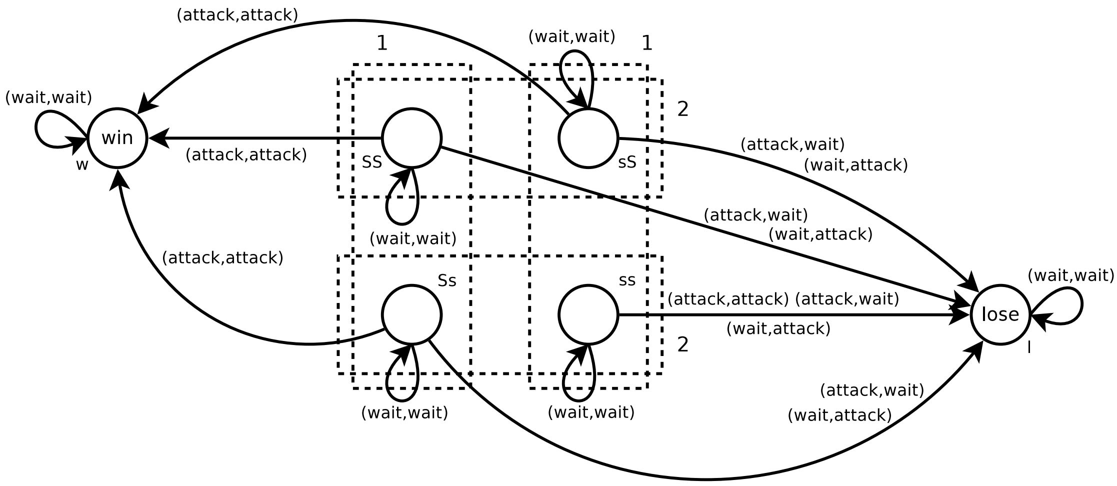


Figure 5.1: CEGS representation of the generals problem described in Example 5.5.

whether the other player knows that it does. The players would need to have common knowledge of the fact that the strategy will be successful in order to execute it without further communication, but this cannot be expressed in ATL_{ir}. In the following section we will consider a logic where this is possible.

5.2 ATOL

Alternating-Time Temporal Observational Logic (ATOL) has been introduced by Wojciech Jamroga and Wiebe van der Hoek in [JvdH04]. It allows to describe strategic abilities of coalitions under imperfect information together with different notions of *observability* of these abilities.

5.2.1 Syntax

ATOL is defined with respect to a set Π of propositions and a number k of players. As usual, we let $\Sigma = \{1, \ldots, k\}$ be the set of players. An ATOL formula is then one of the following:

- p, where $p \in \Pi$.
- $\neg\varphi$ or $\varphi_1 \vee \varphi_2$, where φ, φ_1, φ_2 are ATOL formulas.
- $Obs_a\varphi$, where $a \in \Sigma$ and φ is an ATOL formula.

- $CO_A\varphi$, $EO_A\varphi$, or $DO_A\varphi$, where A is a set of players and φ is an ATOL formula.
- $\langle\langle A\rangle\rangle^{\bullet}_{Obs(\gamma)} \bigcirc \varphi$, $\langle\langle A\rangle\rangle^{\bullet}_{Obs(\gamma)} \Box \varphi$, or $\langle\langle A\rangle\rangle^{\bullet}_{Obs(\gamma)} \varphi_1 \mathcal{U} \varphi_2$, where φ, φ_1, φ_2 are ATOL formulas, A is a set of player, and γ is an player (which is not necessarily a member of A).
- $\langle\langle A\rangle\rangle^{\bullet}_{\Theta(\Gamma)} \bigcirc \varphi$, $\langle\langle A\rangle\rangle^{\bullet}_{\Theta(\Gamma)} \Box \varphi$, or $\langle\langle A\rangle\rangle^{\bullet}_{\Theta(\Gamma)} \varphi_1 \mathcal{U} \varphi_2$, where φ, φ_1, φ_2 are ATOL formulas, A and Γ are sets of players, and $\Theta \in \{CO, DO, EO\}$.

The intended meaning of $Obs_a\varphi$ is "player a observes that φ" (similar to $K_a\varphi$ in ATEL). The modalities $CO_A\varphi$ and $EO_A\varphi$ refer to "common observation" and "everybody observes" (like $C_A\varphi$ and $E_A\varphi$ in ATEL) and $DO_A\varphi$ describes "distributed observation" (similar to $K_A\varphi$ in ATL^D_{iR}).

The intuitive meaning of $\langle\langle A\rangle\rangle^{\bullet}_{Obs(\gamma)}\varphi$ is "there is a uniform strategy for coalition A of which player γ knows that it ensures φ". Analogously, $\langle\langle A\rangle\rangle^{\bullet}_{CO(\Gamma)}\varphi$, $\langle\langle A\rangle\rangle^{\bullet}_{EO(\Gamma)}\varphi$, and $\langle\langle A\rangle\rangle^{\bullet}_{DO(\Gamma)}\varphi$, respectively, express that the players in A have a uniform strategy of which coalition Γ knows that it will achieve φ (where "Γ knows" is understood as common knowledge, everyone's knowledge, and distributed knowledge, respectively).

The modalities Obs_a and $\langle\langle A\rangle\rangle^{\bullet}_{Obs(\gamma)}$ are actually redundant: They can be replaced by $CO_{\{a\}}$ and $\langle\langle A\rangle\rangle^{\bullet}_{CO(\{\gamma\})}$, respectively, as can be seen from the semantics in the following section. Still, the authors of [JvdH04] decided to include them in ATOL, probably in order to be able to emphasize what a single player can observe.

5.2.2 Semantics

In [JvdH04] the semantics of ATOL were defined with respect to so-called *concurrent observational game structures*. These game structures differ from CEGSs mainly by the fact that instead of containing a set of actions Act and a function d which defines the enabled actions at each state, only the number of possible actions for each player at every state is specified. We take the liberty of using CEGSs instead, which clearly leads to equivalent semantics since these do not depend on the "names" of the actions.

Given a CEGS $\mathcal{G} = \langle \Pi, \Sigma, Q, (\sim_a)_{a\in\Sigma}, \pi, Act, d, o\rangle$ and a state q in $\mathcal{G}$, we write $\mathcal{G}, q \models \varphi$ to indicate that the state q in $\mathcal{G}$ satisfies the formula φ (however, as usual, we just write $q \models \varphi$ if $\mathcal{G}$ is fixed). The satisfaction of formulas of the form p ($p \in \Pi$), $\neg\varphi$, and $\varphi_1 \vee \varphi_2$ is defined exactly like in ATL (see (S_p), ($S_\neg$), and ($S_\vee$) in Section 3.2). The remaining cases are the following:

(S_{Obs}) $q \models Obs_a\varphi$ iff $q' \models \varphi$ for all $q' \in [q]_a$.

($S^{\gamma}_{\bigcirc}$) $q \models \langle\langle A\rangle\rangle^{\bullet}_{Obs(\gamma)} \bigcirc \varphi$ iff there exists an ir strategy $f_A = (f_a)_{a\in A}$ for the players in A, such that all outcomes $\lambda \in out([q]_\gamma, f_A)$ satisfy $\lambda[1] \models \varphi$.

($S^{\gamma}_{\Box}$) $q \models \langle\langle A\rangle\rangle^{\bullet}_{Obs(\gamma)} \Box \varphi$ iff there exists an ir strategy $f_A = (f_a)_{a\in A}$ for the players in A, such that for all outcomes $\lambda \in out([q]_\gamma, f_A)$ and all positions $i \geq 0$ we have $\lambda[i] \models \varphi$.

($S^{\gamma}_{\mathcal{U}}$) $q \models \langle\langle A\rangle\rangle^{\bullet}_{Obs(\gamma)} \varphi_1 \mathcal{U} \varphi_2$ iff there exists an ir strategy $f_A = (f_a)_{a\in A}$ for the players in A, such that for all outcomes $\lambda \in out([q]_\gamma, f_A)$ there exists a position $i \geq 0$ such that $\lambda[i] \models \varphi_2$ and for all positions $0 \leq j < i$ we have $\lambda[j] \models \varphi_1$.

(S_Θ) $q \models \Theta_A\varphi$, where $\Theta = CO$, $\Theta = EO$, or $\Theta = DO$, respectively, iff $q' \models \varphi$ for all $q' \in [q]_{C_A}$, $q' \in [q]_{E_A}$, or $q' \in [q]_{D_A}$, respectively.

($S^{\Gamma}_{\bigcirc}$) $q \models \langle\langle A\rangle\rangle^{\bullet}_{\Theta(\Gamma)} \bigcirc \varphi$, where $\Theta = CO$, $\Theta = EO$, or $\Theta = DO$, respectively, iff there exists an ir strategy $f_A = (f_a)_{a\in A}$ for the players in A, such that all outcomes $\lambda \in out([q]_{C_\Gamma}, f_A)$, $\lambda \in out([q]_{E_\Gamma}, f_A)$, or $\lambda \in out([q]_{D_\Gamma}, f_A)$, respectively, satisfy $\lambda[1] \models \varphi$.

($S^{\Gamma}_{\Box}$) $q \models \langle\langle A\rangle\rangle^{\bullet}_{\Theta(\Gamma)} \Box \varphi$, where $\Theta = CO$, $\Theta = EO$, or $\Theta = DO$, respectively, iff there exists an ir strategy $f_A = (f_a)_{a\in A}$ for the players in A, such that for all outcomes $\lambda \in out([q]_{C_\Gamma}, f_A)$, $\lambda \in out([q]_{E_\Gamma}, f_A)$, or $\lambda \in out([q]_{D_\Gamma}, f_A)$, respectively, and all positions $i \geq 0$ we have $\lambda[i] \models \varphi$.

($S^{\Gamma}_{\mathcal{U}}$) $q \models \langle\langle A\rangle\rangle^{\bullet}_{\Theta(\Gamma)} \varphi_1 \mathcal{U} \varphi_2$, where $\Theta = CO$, $\Theta = EO$, or $\Theta = DO$, respectively, iff there exists an ir strategy $f_A = (f_a)_{a\in A}$ for the players in A, such that for all outcomes $\lambda \in out([q]_{C_\Gamma}, f_A)$, $\lambda \in out([q]_{E_\Gamma}, f_A)$, or $\lambda \in out([q]_{D_\Gamma}, f_A)$, respectively, there exists a position $i \geq 0$ such that $\lambda[i] \models \varphi_2$ and for all positions $0 \leq j < i$ we have $\lambda[j] \models \varphi_1$.

Example 5.6. Consider the generals problem described in Example 5.5 and depicted in Figure 5.1 on page 46. It holds that $SS \models \langle\langle 1, 2\rangle\rangle^{\bullet}_{EO(\{1,2\})} \bigcirc win$, but $SS \not\models \langle\langle 1, 2\rangle\rangle^{\bullet}_{CO(\{1,2\})} \bigcirc win$, i.e., it is observed by both generals that they have a winning strategy, but it is not common knowledge. ⚁

Example 5.7. As a second example, we consider the asynchronous rock-paper-scissors variant in Figure 2.4 on page 14. In this game it holds that $q_3 \models \langle\langle 2\rangle\rangle^{\bullet}_{EO(\emptyset)} \bigcirc win_2$, i.e., in state q_3 player 2 has a winning strategy. He also observes that he has such a strategy, i.e., it holds that $q_3 \models Obs_2\langle\langle 2\rangle\rangle^{\bullet}_{EO(\emptyset)} \bigcirc win_2$, but he cannot identify it, which is expressed by $q_3 \models \neg\langle\langle 2\rangle\rangle^{\bullet}_{Obs(2)} \bigcirc win_2$. On the other hand, player 1 can, i.e., it holds that $q_3 \models \langle\langle 2\rangle\rangle^{\bullet}_{Obs(1)} \bigcirc win_2$.

Both players together can ensure that the game never ends. In state q_1 both players can even identify a strategy which achieves this, but in q_2 player 2 cannot: $q_1 \models \langle\langle 1,2\rangle\rangle^{\bullet}_{EO(\{1,2\})} \Box (\neg win_1 \wedge \neg win_2)$, but $q_2 \not\models \langle\langle 1,2\rangle\rangle^{\bullet}_{Obs(2)} \Box (\neg win_1 \wedge \neg win_2)$. ⚁

Example 5.8. As another example, we consider the chess game from Example 2.4 and Figure 2.2 on page 11 (note that we have defined this game in terms of an AETS, but of course we can also reinterpret this to be a CEGS, compare Section 2.3.1).

In this game, one has $a1 \models DO_{\{1,2\}}\neg\langle\langle 1,2\rangle\rangle^{\bullet}_{EO(\emptyset)} (\neg p_b \wedge \neg p_2)\,\mathcal{U}\,(p_d \vee p_4)$, i.e., if the players exchange their knowledge in $a1$, then they know that it is not possible to reach column d or row 4 without crossing column b or row 2. However, player 1 on his own does not know this, i.e., one has $a1 \not\models Obs_1\neg\langle\langle 1,2\rangle\rangle^{\bullet}_{EO(\emptyset)} (\neg p_b \wedge \neg p_2)\,\mathcal{U}\,(p_d \vee p_4)$, because in his view the current state could be $c1$ and $c1 \models \langle\langle 1,2\rangle\rangle^{\bullet}_{EO(\emptyset)} (\neg p_b \wedge \neg p_2)\,\mathcal{U}\,p_d$.

Finally, consider the formula $(p_d \wedge p_4) \rightarrow \langle\langle 2\rangle\rangle^{\bullet}_{Obs(1)} \neg p_1\,\mathcal{U}\,p_b$. This formula expresses that if the current state is $d4$, then player 1 can see a strategy for player 2 which ensures that column b is reached before row 1 is. This formula is valid in the given game. However, player 2 himself does not observe that such a strategy exists, i.e., $p_d \wedge p_4 \rightarrow \langle\langle 2\rangle\rangle^{\bullet}_{Obs(2)} \neg p_1\,\mathcal{U}\,p_b$ is not satisfied in state $d4$, since $d4 \sim_2 d1$ and $d1 \models p_1$. ⚁

5.2.3 Discussion

ATOL allows to express most of the different forms of strategic abilities which we have discussed in the introduction:

- $\langle\langle A\rangle\rangle^{\bullet}_{EO(\emptyset)}\varphi$ expresses the strategic ability (A2), i.e., the fact that coalition A has a uniform strategy which achieves φ, but does not necessarily know this.
- $CO_A\langle\langle A\rangle\rangle^{\bullet}_{EO(\emptyset)}\varphi$, $EO_A\langle\langle A\rangle\rangle^{\bullet}_{EO(\emptyset)}\varphi$, and $DO_A\langle\langle A\rangle\rangle^{\bullet}_{EO(\emptyset)}\varphi$ express the strategic ability (A3), namely that coalition A knows of the existence of a uniform strategy which achieves φ (where "A knows" is understood as common knowledge, everyone's knowledge, and distributed knowledge, respectively).
- $\langle\langle A\rangle\rangle^{\bullet}_{CO(A)}\varphi$, $\langle\langle A\rangle\rangle^{\bullet}_{EO(A)}\varphi$, and $\langle\langle A\rangle\rangle^{\bullet}_{DO(A)}\varphi$ express the strategic ability (A4), i.e., that there is a uniform strategy for coalition A of which they know that it achieves φ (where "they know" is understood as common knowledge, everyone's knowledge, and distributed knowledge, respectively).

Moreover, a rich variety of what coalitions know of each other can be described. For example, $DO_A\langle\langle B\rangle\rangle^{\bullet}_{EO(\emptyset)}\varphi$ expresses that the players in A have distributed knowledge of the fact that coalition B has a strategy which achieves φ, while $\langle\langle B\rangle\rangle^{\bullet}_{DO(A)}\varphi$ describes that the players in A can, by combining their knowledge, even construct such a strategy.

Remark 5.9. ATOL cannot be used to describe strategic abilities of type (A1), simply because only uniform strategies are considered. But this does not seem to be a real drawback, since this type of strategic power is rather uninteresting for most applications. ⌟

ATOL is strictly more expressive than ATL$_{ir}$: Firstly, it subsumes ATL$_{ir}$, since the modality $\langle\langle A\rangle\rangle_{ir}$ can be written in ATOL as $\langle\langle A\rangle\rangle^{\bullet}_{EO(A)}$. Secondly, it allows to express what players and coalitions observe, also taking into account distributed knowledge and everyone's knowledge, which is not possible in ATL$_{ir}$. Surprisingly, in spite of this high expressivity, the model checking problem for ATOL is not harder than the one for ATL$_{ir}$:

Theorem 5.10 (ATOL model checking complexity). *The model checking problem for ATOL is Δ_2^P-complete in the number of transitions in the model and the length of the formula.*

The upper bound (i.e., the fact that the problem is Δ_2^P-easy) has already been given in [JvdH04], where the authors argued that the proof is analogous to the one for ATL$_{ir}$. In the same paper they only showed NP-hardness of the problem, but of course Δ_2^P-hardness follows from Theorem 5.4 and the fact that ATOL subsumes ATL$_{ir}$.

Hence ATOL characterizes many meaningful levels of strategic abilities and has a not too hard model checking problem. However, the authors in [JÅ07] criticized the fact that it only contains modalities for some fixed combinations of strategic abilities and "epistemic modes", while it would actually be nice to allow *arbitrary* combinations: E.g., $\langle\langle A\rangle\rangle^{\bullet}_{EO_A EO_A}\varphi$ is not a well formed ATOL formula, although it is easy to give a meaningful interpretation to this formula. This has led them to the development of CSL, which we describe in the following section.

5.3 CSL

Constructive Strategic Logic (CSL) was introduced by Thomas Ågotnes and Wojciech Jamroga in [JÅ07], where they state that their aim was "to come up with a logic of ability under imperfect information, which is both general and elegant". Their idea is the introduction of the notion of “constructive” knowledge, which is inspired by constructivism: A coalition is said to *constructively know* that a certain strategy exists, if it is able to present it.

Thus, in addition to “standard” knowledge modalities C, E, and D, which refer to common knowledge, everyone’s knowledge, and distributed knowledge as we have used it throughout this thesis, they introduce *constructive knowledge operators* $\mathbb{C}$, $\mathbb{E}$, and $\mathbb{D}$. While $C_A\langle\langle A\rangle\rangle\varphi$ expresses the fact that coalition A has common knowledge of the fact that they have a strategy to achieve φ (strategic ability (A3)), $\mathbb{C}_A\langle\langle A\rangle\rangle\varphi$ means that there is a strategy for A of which it is common knowledge among the players in A that it ensures φ (strategic ability (A4)).

5.3.1 Syntax

Like all other considered logics, CSL is defined with respect to a set Π of propositions and a finite set $\Sigma = \{1, \ldots, k\}$ of players. A CSL formula is then one of the following:

- p, where $p \in \Pi$ is a proposition.
- $\neg\varphi$, where φ is an CSL formula.
- $\varphi_1 \wedge \varphi_2$, where φ_1 and φ_2 are CSL formulas.
- $\langle\langle A\rangle\rangle \bigcirc \varphi$, $\langle\langle A\rangle\rangle \Box \varphi$, or $\langle\langle A\rangle\rangle \varphi_1 \mathcal{U} \varphi_2$, where $A \subseteq \Sigma$ is a set of players and φ, φ_1, φ_2 are CSL formulas.
- $\mathcal{K}_A\varphi$, where $A \subseteq \Sigma$ is a set of players, $\mathcal{K}_A \in \{C_A, E_A, D_A\}$, and φ is a CSL formula.
- $\hat{\mathcal{K}}_A\varphi$, where $A \subseteq \Sigma$ is a set of players, $\hat{\mathcal{K}}_A \in \{\mathbb{C}_A, \mathbb{E}_A, \mathbb{D}_A\}$, and φ is a CSL formula.

Additional boolean connectives like $\vee$, $\rightarrow$, and $\leftrightarrow$ can be defined from $\neg$ and $\wedge$ as always.

C_A, E_A, and D_A are the “usual” epistemic operators for different forms of group knowledge, while $\mathbb{C}_A$, $\mathbb{E}_A$, and $\mathbb{D}_A$ are *constructive* knowledge operators. Note that there is no modality for the knowledge of a single player, but this can be expressed by $K_a := C_{\{a\}}$, $K_a := E_{\{a\}}$, or $K_a := D_{\{a\}}$ for non-constructive knowledge and by $\mathbb{K}_a := \mathbb{C}_{\{a\}}$, $\mathbb{K}_a := \mathbb{E}_{\{a\}}$, or $\mathbb{K}_a := \mathbb{D}_{\{a\}}$ for constructive knowledge.

5.3.2 Semantics

Like most of the other logics considered in this thesis, CSL formulas are interpreted in concurrent epistemic game structures. However, CSL formulas are evaluated in sets of states rather than in single states.

The idea behind this is the fact that a coalition A constructively knows of the fact that there is strategy which achieves a certain goal iff there is one strategy which is successful from all states which are possible from A's point of view. Hence, the semantics of constructive knowledge modalities will have to refer to sets of states rather than to single states.

Given a CEGS $\mathcal{G} = \langle \Pi, \Sigma, Q, (\sim_a)_{a\in\Sigma}, \pi, Act, d, o\rangle$, we write $\mathcal{G}, S \models \varphi$ if the CSL formula φ is satisfied in the set $S \subseteq Q$ (as always, if $\mathcal{G}$ is fixed, we just write $S \models \varphi$). The satisfaction relation $\models$ is inductively defined as follows:

(S_p) $S \models p$, for a proposition $p \in \Pi$, iff $p \in \pi(q)$ for every $q \in S$.

($S_\neg$) $S \models \neg\varphi$ iff $S \not\models \varphi$.

($S_\wedge$) $S \models \varphi_1 \wedge \varphi_2$ iff $S \models \varphi_1$ and $S \models \varphi_2$.

($S_\bigcirc$) $S \models \langle\langle A\rangle\rangle \bigcirc \varphi$ iff there exists an ir strategy $f_A = (f_a)_{a\in A}$ for the players in A, such that all outcomes $\lambda \in out(S, f_A)$ satisfy $\{\lambda[1]\} \models \varphi$.

($S_\Box$) $S \models \langle\langle A\rangle\rangle \Box \varphi$ iff there exists an ir strategy $f_A = (f_a)_{a\in A}$ for the players in A, such that for all outcomes $\lambda \in out(S, f_A)$ and all positions $i \geq 0$ we have $\{\lambda[i]\} \models \varphi$.

($S_\mathcal{U}$) $S \models \langle\langle A\rangle\rangle \varphi_1 \mathcal{U} \varphi_2$ iff there exists an ir strategy $f_A = (f_a)_{a\in A}$ for the players in A, such that for all outcomes $\lambda \in out(S, f_A)$ there exists a position $i \geq 0$ such that $\{\lambda[i]\} \models \varphi_2$ and for all positions $0 \leq j < i$ we have $\{\lambda[j]\} \models \varphi_1$.

($S_\mathcal{K}$) $S \models \mathcal{K}_A\varphi$ (where $\mathcal{K}_A = C_A$, $\mathcal{K}_A = E_A$, or $\mathcal{K}_A = D_A$) iff all $q \in [S]_{\mathcal{K}_A}$ satisfy $\{q\} \models \varphi$.

($S_{\hat{\mathcal{K}}}$) $S \models \hat{\mathcal{K}}_A\varphi$ (where $\hat{\mathcal{K}}_A = \mathbb{C}_A$, $\hat{\mathcal{K}}_A = \mathbb{E}_A$, or $\hat{\mathcal{K}}_A = \mathbb{D}_A$) iff $[S]_{\mathcal{K}_A} \models \varphi$ (where $\mathcal{K}_A = C_A$, $\mathcal{K}_A = E_A$, or $\mathcal{K}_A = D_A$, respectively).

Finally, one is also interested in the satisfaction of formulas in single states (e.g., in the initial state of a game), and hence one defines the satisfaction relation for states simply by $q \models \varphi$ iff $\{q\} \models \varphi$. Given a CEGS $\mathcal{G}$, a state q in $\mathcal{G}$ and a CSL formula φ, the model checking problem for CSL asks whether φ holds in q.

Remark 5.11. The authors in [JÅ07] also define an alternative type of negation $\sim$ by $S \models\sim \varphi$ iff $q \not\models \varphi$ for all $q \in S$. However, this negation does not behave like classical negation since it does neither obey the law of double negation nor the law of excluded middle, hence we will not use it. ⌟

Due to the fact that CSL formulas are in general evaluated in sets of states, one can consider two different types of validity of a formula: A CSL formula φ is said to be *weakly valid* iff $\mathcal{G}, q \models \varphi$ for every CGS $\mathcal{G}$ and every state q in $\mathcal{G}$. φ is *strongly valid* iff $\mathcal{G}, S \models \varphi$ for every CGS $\mathcal{G}$ and every set S of states in $\mathcal{G}$. Clearly, strong validity implies weak validity, but not vice versa. For example, the formula $\langle\langle\emptyset\rangle\rangle \neg p \,\mathcal{U}\, \neg p \leftrightarrow \neg p$ is weakly but not strongly valid [JÅ07].

Example 5.12. We consider the chess game from Example 2.4 which is depicted in Figure 2.2 on page 11 (again, we reinterpret this AETS to be a CEGS).

In this game, one has $a1 \models E_{\{1,2\}}\langle\langle 1,2\rangle\rangle \neg(p_b \wedge p_2)\,\mathcal{U}\, p_d$, i.e., each of the players knows in state $a1$ that they can work together in order to reach column d without visiting $b2$. Even more, each of them can construct a strategy which achieves this: E.g., it holds that $a1 \models \mathbb{K}_2\langle\langle 1,2\rangle\rangle \neg(p_b \wedge p_2)\,\mathcal{U}\, p_d$, since $\{a1, a2, a3, a4\} \models \langle\langle 1,2\rangle\rangle \neg(p_b \wedge p_2)\,\mathcal{U}\, p_d$.

In state $b2$ the formula $\langle\langle 1,2\rangle\rangle \bigcirc ((p_a \wedge p_1) \vee (p_c \wedge p_3))$ is satisfied, and both players know this, i.e., $b2 \models E_{\{1,2\}}\langle\langle 1,2\rangle\rangle \bigcirc ((p_a \wedge p_1) \vee (p_c \wedge p_3))$. However, each of them individually does not have constructive knowledge of a corresponding strategy, i.e., $b2 \not\models \mathbb{K}_1\langle\langle 1,2\rangle\rangle \bigcirc ((p_a \wedge p_1) \vee (p_c \wedge p_3))$ and $b2 \not\models \mathbb{K}_2\langle\langle 1,2\rangle\rangle \bigcirc ((p_a \wedge p_1) \vee (p_c \wedge p_3))$. But if they combine their knowledge, then they can construct such a strategy, i.e., $b2 \models \mathbb{D}_{\{1,2\}}\langle\langle 1,2\rangle\rangle \bigcirc ((p_a \wedge p_1) \vee (p_c \wedge p_3))$. ⚁

Example 5.13. As another example, we again consider Indiana Jones' treasure hunt from Figure 4.1 on page 33. The formula $\neg rich \rightarrow \langle\langle 1\rangle\rangle \bigcirc \langle\langle 1\rangle\rangle \bigcirc dead$ is (weakly) valid in this game, i.e., as long as Indiana is not rich he has the possibility to die in (at most) two steps. He also knows that he has this possibility in that case, i.e., the formula $\neg rich \rightarrow K_1\langle\langle 1\rangle\rangle \bigcirc \langle\langle 1\rangle\rangle \bigcirc dead$ is also weakly valid. However, in general he can not construct a strategy which ensures that he will die: For example, we have $d_0 \not\models \neg rich \rightarrow \mathbb{K}_1\langle\langle 1\rangle\rangle \bigcirc \langle\langle 1\rangle\rangle \bigcirc dead$, since $\{d_0, \ldots, d_{999}\} \not\models \langle\langle 1\rangle\rangle \bigcirc \langle\langle 1\rangle\rangle \bigcirc dead$. Indiana Jones has to be careful not to get rich by accident ;-) ⚁

5.3.3 Discussion

CSL allows one to express different forms of strategic abilities in a very elegant way, namely by using suitable combinations of the modalities:

- $\langle\langle A\rangle\rangle\varphi$ expresses the strategic ability (A2), namely the fact that coalition A has a uniform strategy which achieves φ, but the players in A do not necessarily know that such a strategy exists.
- $C_A\langle\langle A\rangle\rangle\varphi$, $E_A\langle\langle A\rangle\rangle\varphi$, and $D_A\langle\langle A\rangle\rangle\varphi$ express the strategic ability (A3), i.e., that coalition A knows of the existence of a uniform strategy which achieves φ (where "A knows" is understood as common knowledge, everyone's knowledge, and distributed knowledge, respectively).
- $\mathbb{C}_A\langle\langle A\rangle\rangle\varphi$, $\mathbb{E}_A\langle\langle A\rangle\rangle\varphi$, and $\mathbb{D}_A\langle\langle A\rangle\rangle\varphi$ express the strategic ability (A4), namely that there is a uniform strategy of which coalition A knows that it ensures φ (where "A knows" is understood as common knowledge, everyone's knowledge, and distributed knowledge, respectively).

Moreover, many other notions of what coalitions know of each other can be described. For example, $\mathbb{D}_A\langle\langle B\rangle\rangle\varphi$ expresses that the players in A can, by combining their knowledge, come up with a strategy for coalition B which achieves φ, while $D_A\langle\langle B\rangle\rangle\varphi$ only means that the players of A have distributed knowledge of the existence of such a strategy.

Remark 5.14. Like ATOL, CSL cannot be used to describe strategic abilities of type (A1), since only uniform strategies are considered in the semantics. ⌟

Apart from being a very elegant solution for describing different forms of strategic powers, CSL is also more expressive than the other logics for games of imperfect recall we considered:

Theorem 5.15 ([JÅ07]). *CSL is strictly more expressive than ATL$_{ir}$ and ATOL.*

Sketch of proof. Jamroga and Ågotnes give a translation of ATL$_{ir}$ and ATOL to CSL, which embeds these two logics into CSL and shows that CSL is at least as expressive as ATL$_{ir}$ and ATOL:

- The cooperation modality $\langle\langle A\rangle\rangle_{ir}$ of ATL$_{ir}$ can be written in CSL as $\mathbb{E}_A\langle\langle A\rangle\rangle$.
- The ATOL cooperation modalities $\langle\langle A\rangle\rangle^{\bullet}_{CO(\Gamma)}$, $\langle\langle A\rangle\rangle^{\bullet}_{EO(\Gamma)}$, and $\langle\langle A\rangle\rangle^{\bullet}_{DO(\Gamma)}$ can be expressed in CSL by $\mathbb{C}_\Gamma\langle\langle A\rangle\rangle$, $\mathbb{E}_\Gamma\langle\langle A\rangle\rangle$, and $\mathbb{D}_\Gamma\langle\langle A\rangle\rangle$, whereas the ATOL knowledge modalities CO_A, EO_A, and DO_A can simply be translated to C_A, E_A, and D_A, respectively. This translation is of course sufficient, since the ATOL modalities referring to the observations of a single player (i.e., Obs_a and $\langle\langle A\rangle\rangle^{\bullet}_{Obs(\gamma)}$) are actually redundant in ATOL.

Finally, Jamroga and Ågotnes justify the fact that CSL is strictly more expressive than ATL_{ir} and ATOL by proving that the modality $\mathbb{E}_A\mathbb{E}_A\langle\langle A\rangle\rangle$ has no counterpart in ATL_{ir} and ATOL. □

Note that the above-mentioned translation also yields a reduction of the ATL_{ir} and ATOL model checking problems to CSL model checking, which shows that the model checking problem for CSL is at least Δ_2^P-hard. Interestingly, in spite of CSL's high expressivity, the problem is not harder:

Theorem 5.16 (CSL model checking complexity [JÅ07]). *The model checking problem for CSL is Δ_2^P-complete in the number of transitions in the model and the length of the formula.*

Sketch of proof. The authors show the upper bound by presenting an algorithm which polynomially often guesses a strategy, trims the model according to this strategy, and then calls a CTL model checking procedure (i.e., their algorithm is basically an extension of Schobbens' algorithm for the model checking of ATL_{ir} from [Sch04], cf. Section 5.1.2). □

We want to remark that the high expressivity and elegance of CSL, which results from the possibility of combining constructive knowledge operators and cooperation modalities (instead of defining modalities for all possible combinations like in ATOL), also has two disadvantages, which were already acknowledged by the authors of [JÅ07]: Firstly, CSL contains formulas for which it is hard to come up with an interpretation of what they express. For example, the formula $\mathbb{K}_a\neg\langle\langle a\rangle\rangle\varphi$ reads as "player a has *constructive* knowledge about being *unable* to achieve φ", which seems awkward. Secondly, somewhat strangely, the *truth axiom*

$$\mathbb{K}_a\varphi \rightarrow \varphi \tag{5.1}$$

does not hold in CSL, as can be seen from the following example:

Example 5.17. Consider the simple CEGS $\mathcal{G} = \langle \Pi, \Sigma, Q, (\sim_a)_{a\in\Sigma}, \pi, Act, d, o\rangle$ with $\Pi := \{p\}$, $\Sigma := \{1\}$, $Q := \{q_1, q_2\}$, $\sim_a := Q^2$, $\pi(q_1) := \{p\}$, $\pi(q_2) := \{\}$, and arbitrary Act, d, and o. Since in this game $\{q_1, q_2\} \models \neg p$, it holds that $q_1 \models \mathbb{K}_1\neg p$. On the other hand, $q_1 \not\models \neg p$, and hence the formula $\mathbb{K}_a\neg p \rightarrow \neg p$ is not satisfied in q_1. ⌟

Both problems are caused by formulas in which constructive knowledge operators are directly followed by negation. Jamroga and Ågotnes thus suggest that in order to solve these problems, one could restrict CSL to the language CSL$^-$, in which between every occurrence of constructive knowledge and negation there is always at least one operator other than conjunction. They show that every instance of (5.1) in CSL$^-$ is strongly valid, and moreover they argue that by only considering CSL$^-$ one circumvents the problem of formulas without intuitive meaning.

As a concluding remark we want to note that, in spite of its high expressivity, there are subtle notions of strategic power which still can not be expressed in CSL: In [JvdH04], Jamroga and van der Hoek point out that even common knowledge of a strategy for φ among the members of a coalition A (i.e., the satisfaction of the CSL formula $\mathbb{C}_A\langle\langle A\rangle\rangle\varphi$) in a state q does not necessarily imply that coalition A will be able to ensure φ in q without further communication. This is shown in the following example:

Example 5.18. Consider the coin game from Example 2.6 and the corresponding CGS in Figure 2.1b on page 9 (of course, we can interpret this CGS to be a CEGS). In state s of this game, the formula $\mathbb{C}_{1,2}\langle\langle 1, 2\rangle\rangle \bigcirc win$ is satisfied, as the strategy (f_1, f_2) with $f_1(s) = f_2(s) =$ heads and $f_1(w) = f_2(w) =$ do_nothing shows. Despite this fact, the players are clearly not able to win this game if they are not allowed to communicate. ⚁

Of course, the problem here is that there are two different strategies which enforce φ, thus the players need some kind of communication in order to decide which one they will take. Hence we see that CSL only formalizes what coalitions *can in principle achieve as dictated by the rules of the game*, at the same time neglecting how cooperations among players are realized. But this is a conceptual detail which actually seems to appear in all logics which are inspired by ATL.

CHAPTER 6

Approximating ATL$_{iR}$

In this chapter we want to address the problem that the model checking problem for ATL$_{iR}$ is undecidable, even though it is (arguably) the simplest logic for coalitional games of imperfect information which uses uniform perfect recall strategies. While this is a very discouraging fact, Schobbens notes in [Sch04] that ATL$_{iR}$ can at least be approximated, since the following implications hold:

$$\langle\langle A\rangle\rangle_{ir}\varphi \quad\Rightarrow\quad \langle\langle A\rangle\rangle_{iR}\varphi \quad\Rightarrow\quad \langle\langle A\rangle\rangle_{IR}\varphi.$$

Of course, these implications are straightforward: Firstly, if a coalition A without memory can achieve φ, then A can also ensure φ under the assumption of perfect recall. Secondly, if the players in A can achieve φ under imperfect information, then they can also do it if they have perfect information. Hence, if in a state q of a CEGS we have $q \models \langle\langle A\rangle\rangle_{ir}\varphi$ or $q \not\models \langle\langle A\rangle\rangle_{IR}\varphi$, then we know the truth value of the formula $\langle\langle A\rangle\rangle_{iR}\varphi$. This fact allows for a model checker which, at least in some cases, returns the correct truth value of $\langle\langle A\rangle\rangle_{iR}\varphi$ by using the mentioned approximations, and answers "I don't know" if the approximations do not yield a result.

Remark 6.1. Of course the given approximations only make sense if φ contains no cooperation modalities, since otherwise $\langle\langle A\rangle\rangle_{ir}\varphi$ and $\langle\langle A\rangle\rangle_{IR}\varphi$ will not be well-formed formulas. But this is no problem, since we can evaluate every ATL$_{iR}$ formula "bottom-up", i.e., by first evaluating every subformula φ' in all states of the given model, adding a new proposition $p_{\varphi'}$ to the model in order to "store" the results, and replacing each occurrence of φ' in the original formula by $p_{\varphi'}$. ⌟

Our aim is to refine the above approximations, i.e., we will present two alternative logics ATL$_{i\bar{R}}$ and ATL$_{iR^C}$ with modalities $\langle\langle A\rangle\rangle_{i\bar{R}}$ and $\langle\langle A\rangle\rangle_{iR^C}$, respectively, which satisfy

$$\langle\langle A\rangle\rangle_{ir}\varphi \quad \underset{\not\Leftarrow}{\Rightarrow} \quad \langle\langle A\rangle\rangle_{i\bar{R}}\varphi \quad \Rightarrow \quad \langle\langle A\rangle\rangle_{iR}\varphi \quad \Rightarrow \quad \langle\langle A\rangle\rangle_{iR^C}\varphi \quad \underset{\not\Leftarrow}{\Rightarrow} \quad \langle\langle A\rangle\rangle_{IR}\varphi.$$

The logic ATL$_{iR^C}$ is a modification of ATL$^D_{iR}$ from [DEG10] (see Section 4.3), which uses the idea of considering strategies which depend on the distributed knowledge among the coalition members. The lower bound approximation ATL$_{i\bar{R}}$ uses a special kind of bounded memory strategy which to our knowledge has not yet been considered in literature on logics for coalitional games of imperfect information.

6.1 ATL$_{i\bar{R}}$

Our motivation for the introduction of ATL$_{i\bar{R}}$ is the following: Suppose that the players of a coalition need perfect recall only in order to gain more information about *where they really are* in the game. Then they actually do not have to recall the whole history, rather it is sufficient to remember what they knew in the previous state and derive their knowledge for the present state from this information.

Example 6.2. Once more, we consider Indiana Jones' treasure hunt (see Example 4.7 and Figure 4.1 on page 33). In this game, Indiana Jones has an iR strategy to find the treasure (at least if the game does not start in d_{trap}): He just has to walk clockwise until he finds the white door, and then he continues and opens the 950th black door he encounters. But this does not actually require an iR strategy: As soon as Indiana finds the white door, he knows exactly where he is in the game, and it is sufficient to update this information while walking on in order to win the game. ⚁

Thus, our idea is the introduction of strategies which are based on *updated a-views*:

Definition 6.3 (Updated a-view). *Let λ be a computation of length n in a concurrent epistemic game structure $\mathcal{G} = \langle \Pi, \Sigma, Q, (\sim_a)_{a\in\Sigma}, \pi, Act, d, o\rangle$. Then we denote by $[[\lambda]]_a$ the set of states which can be reached in $\mathcal{G}$ by any computation of length n which player a cannot distinguish from λ. Formally,*

$$[[\lambda]]_a := \{\lambda'[n] \mid \lambda' \textit{ is a computation in } \mathcal{G}, \lambda'[i] \sim_a \lambda[i], 0 \le i \le n\}.$$

We say that $[[\lambda]]_a$ is an updated a-view.

Note that updated a-views can be computed iteratively while the game proceeds, via $[[q_0]]_a = [q_0]_a$, and

$$[[q_0 \dots q_n]]_a = \{s \in [q_n]_a \mid s \text{ is reachable from a state } s' \in [[q_0 \dots q_{n-1}]]_a\},$$

for $n \geq 1$.

Example 6.4. Consider Indiana Jones' treasure hunt (see Figure 4.1 on page 33). The longer Indiana moves through the corridor, the better his updated view becomes, e.g., $[[d_0]]_a = [d_0]_a$, $[[d_0 d_1]]_a = [d_0]_a \setminus \{d_{51}\}$, $[[d_0 d_1 d_2]]_a = [d_0]_a \setminus \{d_{51}, d_{52}\}$, etc. When he finally arrives at the white door, he suddenly knows exactly where he is in the corridor, i.e., $[[d_0 \dots d_{50}]] = \{d_{50}\}$, and he can update this information in the subsequent steps, i.e., $[[d_0 \dots d_{51}]] = \{d_{51}\}$, $[[d_0 \dots d_{52}]] = \{d_{52}\}$, and so on. ⚁

Remark 6.5. The set $[[q_0 \dots q_n]]_a$ contains exactly those states which player a deems possible given the history $q_0 \dots q_n$ if he remembers the sequence of equivalence classes $[q_0]_a \dots [q_n]_a$ the game went through from his point of view. Of course, the player could in general further restrict this set if he also remembered the actions he took in each step of the game. But in order to exploit this idea we would have to "store" the actions of the players in the histories of the game (like it is done in the runs of game arenas for the logic ATL$^D_{iR}$, cf. Section 4.3.1), such that we can later use them in the semantics. However, since we eventually want to approximate ATL$_{iR}$, where histories do not contain the actions of the players, we will for our purpose assume that the players forget what they did in the past. ⌟

In the following, we let $\bar{Q}_a$ denote the set of all subsets of a-views, i.e.,

$$\bar{Q}_a := \{S \in 2^Q \mid \exists q \in Q : S \subseteq [q]_a\}.$$

This set contains all state sets which can occur as updated a-views during a game in the given CEGS. Using this, we are now able to define our new type of strategy:

Definition 6.6 ($i\bar{R}$ strategy). *An* imperfect information and bounded recall strategy *(for short, an $i\bar{R}$* strategy*) for a player $a \in \Sigma$ in a CEGS is a function $f_a : \bar{Q}_a \to Act$ with the property that for each $S \in \bar{Q}_a$, the action $f_a(S)$ is enabled in the states contained in S.*

An $i\bar{R}$ strategy allows a player to choose a different action for each updated a-view which he encounters during the game. As usual, we let an $i\bar{R}$ strategy for a coalition $A \subseteq \Sigma$ just be a tuple $(f_a)_{a \in A}$ of $i\bar{R}$ strategies for the players in A. Note that in order to execute such a strategy, the size of the memory which each player a needs for storing his information about the past is bounded by the size of the largest a-view.

Like all other types of strategies in CEGSs, we can view each $i\bar{R}$ strategy f_a as a function $Q^+ \to Act$ by defining, for each $\lambda \in Q^+$,

$$f_a(\lambda) := f_a([[\lambda]]_a).$$

This way, the set of outcomes $out(q, f_A)$ from state q of a strategy $f_A = (f_a)_{a \in A}$ for coalition A is properly defined by Definition 2.17.

6.1.1 Syntax

The syntax of ATL$_{i\bar{R}}$ is exactly the same as the one of ATL$_{iR}$. However, in order to make the distinction of the two logics explicit, we mark cooperation modalities with with "$i\bar{R}$" as a subscript, i.e., ATL$_{i\bar{R}}$ formulas starting with a cooperation modality are of the form $\langle\langle A\rangle\rangle_{i\bar{R}} \bigcirc \varphi$, $\langle\langle A\rangle\rangle_{i\bar{R}} \Box \varphi$ and $\langle\langle A\rangle\rangle_{i\bar{R}} \varphi_1 \mathcal{U} \varphi_2$.

6.1.2 Semantics

Like formulas of ATL$_{iR}$, ATL$_{i\bar{R}}$ formulas, which are defined with respect to the set Π of propositions and the set $\Sigma = \{1, \ldots, k\}$ of players, are interpreted in the states of a CEGS $\mathcal{G} = \langle \Pi, \Sigma, Q, (\sim_a)_{a \in \Sigma}, \pi, Act, d, o\rangle$. The satisfaction of formulas of the form p ($p \in \Pi$), $\neg\varphi$ and $\varphi_1 \vee \varphi_2$ is defined exactly like in ATL and ATL$_{iR}$ (see (S_p), ($S_\neg$), and ($S_\vee$) in Section 3.2). For the other cases, the definition of the satisfaction relation $\models$ differs from the one for ATL$_{iR}$ only in the type of strategies which are used:

($S'_\bigcirc$) $q \models \langle\langle A\rangle\rangle_{i\bar{R}} \bigcirc \varphi$ iff there exists an $i\bar{R}$ strategy $f_A = (f_a)_{a \in A}$ for the players in A, such that all outcomes $\lambda \in out([q]_{E_A}, f_A)$ satisfy $\lambda[1] \models \varphi$.

($S'_\Box$) $q \models \langle\langle A\rangle\rangle_{i\bar{R}} \Box \varphi$ iff there exists an $i\bar{R}$ strategy $f_A = (f_a)_{a \in A}$ for the players in A, such that for all outcomes $\lambda \in out([q]_{E_A}, f_A)$ and all positions $i \geq 0$ we have $\lambda[i] \models \varphi$.

($S'_\mathcal{U}$) $q \models \langle\langle A\rangle\rangle_{i\bar{R}} \varphi_1 \mathcal{U} \varphi_2$ iff there exists an $i\bar{R}$ strategy $f_A = (f_a)_{a \in A}$ for the players in A, such that for all outcomes $\lambda \in out([q]_{E_A}, f_A)$ there exists a position $i \geq 0$ such that $\lambda[i] \models \varphi_2$ and for all positions $0 \leq j < i$ we have $\lambda[j] \models \varphi_1$.

Example 6.7. In the Indiana Jones example (see Figure 4.1 on page 33) the formula $\neg dead \rightarrow \langle\langle 1\rangle\rangle_{i\bar{R}} \top \mathcal{U}\, rich$ is valid, i.e., as long as Indiana is not dead he has an $i\bar{R}$ strategy which ensures that he will find the treasure. The strategy f_1 which achieves this is simply given by

$$f_1(S) = \begin{cases} \text{open}, & \text{if } S = \{d_0\}, \\ \text{enjoy}, & \text{if } S = \{d_{treasure}\}, \\ \text{rip}, & \text{if } S = \{d_{trap}\}, \\ \text{walk}, & \text{else.} \end{cases}$$

6.1.3 Relation to ATL$_{ir}$ and ATL$_{iR}$

In this section we show the implications

$$\langle\langle A\rangle\rangle_{ir}\varphi \quad \underset{\not\Leftarrow}{\Rightarrow} \quad \langle\langle A\rangle\rangle_{i\bar{R}}\varphi \quad \Rightarrow \quad \langle\langle A\rangle\rangle_{iR}\varphi.$$

Proposition 6.8. *ATL$_{i\bar{R}}$ approximates ATL$_{iR}$ from below, in the sense that for each CEGS $\mathcal{G} = \langle \Pi, \Sigma, Q, (\sim_a)_{a\in\Sigma}, \pi, Act, d, o\rangle$, each state $q \in Q$ and each formula φ of the form $\bigcirc\varphi_1$, $\Box\,\varphi_1$, or $\varphi_1\,\mathcal{U}\,\varphi_2$, where φ_1 and φ_2 are propositional formulas, we have the implication*

$$q \models \langle\langle A\rangle\rangle_{i\bar{R}}\varphi \quad \Rightarrow \quad q \models \langle\langle A\rangle\rangle_{iR}\varphi.$$

Proof. This simply holds because every $i\bar{R}$ strategy for a player a is also an iR strategy for a (if we view both types of strategies as functions $f_a : Q^+ \to Act$).

More formally: Suppose that $q \models \langle\langle A\rangle\rangle_{i\bar{R}}\varphi$, i.e., there is an $i\bar{R}$ strategy $f_A = (f_a)_{a\in A}$ for the players in A, such that all outcomes $\lambda \in out([q]_{E_A}, f_A)$ fulfill the required condition for the satisfaction of $\bigcirc\varphi_1$, $\Box\,\varphi_1$, or $\varphi_1\,\mathcal{U}\,\varphi_2$, respectively. We define an iR strategy $\tilde{f}_A = (\tilde{f}_a)_{a\in A}$ for A by setting, for any $[q_0]_a \dots [q_n]_a \in Q_a^+$,

$$\tilde{f}_a([q_0]_a \dots [q_n]_a) := f_a([[q_0 \dots q_n]]_a).$$

Note that $\tilde{f}_a$ is well-defined due to the fact that whenever two histories $q_0 \dots q_n, q'_0 \dots q'_n$ are indistinguishable to player a, i.e., $[q_0]_a \dots [q_n]_a = [q'_0]_a \dots [q'_n]_a$, then also the a-views given these histories coincide, i.e., $[[q_0 \dots q_n]]_a = [[q'_0 \dots q'_n]]_a$. By the construction of $\tilde{f}_a$ it clearly holds that $out([q]_{E_A}, f_A) = out([q]_{E_A}, \tilde{f}_A)$, and hence all outcomes $\lambda \in out([q]_{E_A}, \tilde{f}_A)$ fulfill the required satisfaction condition for $\bigcirc\varphi_1$, $\Box\,\varphi_1$, or $\varphi_1\,\mathcal{U}\,\varphi_2$, respectively. Thus we have $q \models \langle\langle A\rangle\rangle_{iR}\varphi$. ☐

Remark 6.9. If φ is an ATL$_{iR}$ formula and $\bar{\varphi}$ results from φ by replacing all occurrences of $\langle\langle A\rangle\rangle_{iR}$ by $\langle\langle A\rangle\rangle_{i\bar{R}}$, it does in general *not* hold that $q \models \bar{\varphi}$ implies $q \models \varphi$. We leave the construction of a counter-example to the reader. ⌟

Proposition 6.10. *ATL$_{i\bar{R}}$ approximates ATL$_{iR}$ strictly better than ATL$_{ir}$, i.e., for each CEGS $\mathcal{G} = \langle \Pi, \Sigma, Q, (\sim_a)_{a\in\Sigma}, \pi, Act, d, o\rangle$, each state $q \in Q$ and each φ of the form $\bigcirc\varphi_1$, $\Box\,\varphi_1$, or $\varphi_1\,\mathcal{U}\,\varphi_2$, where φ_1 and φ_2 are propositional formulas, we have the implication*

$$q \models \langle\langle A\rangle\rangle_{ir}\varphi \quad \Rightarrow \quad q \models \langle\langle A\rangle\rangle_{i\bar{R}}\varphi,$$

whereas, in general,

$$q \models \langle\langle A\rangle\rangle_{ir}\varphi \quad \not\Leftarrow \quad q \models \langle\langle A\rangle\rangle_{i\bar{R}}\varphi.$$

Proof. The implication $q \models \langle\langle A\rangle\rangle_{ir}\varphi \Rightarrow q \models \langle\langle A\rangle\rangle_{i\bar{R}}\varphi$ follows from the fact that every ir strategy for a player a is also an $i\bar{R}$ strategy for a (if we view both types of strategies as functions $f_a : Q^+ \to Act$).

More formally: Suppose that $q \models \langle\langle A\rangle\rangle_{ir}\varphi$, i.e., there is an ir strategy $f_A = (f_a)_{a\in A}$ for the players in A, such that all outcomes $\lambda \in out([q]_{E_A}, f_A)$ fulfill the required condition for the satisfaction of $\bigcirc\varphi_1$, $\Box\,\varphi_1$, or $\varphi_1\,\mathcal{U}\,\varphi_2$, respectively. We then define an $i\bar{R}$ strategy $\tilde{f}_A = (\tilde{f}_a)_{a\in A}$ for A by setting, for any $[[q_0 \dots q_n]]_a \in \bar{Q}_a$,

$$\tilde{f}_a([[q_0 \dots q_n]]_a) := f_a([q_n]_a).$$

Note that $\tilde{f}_a$ is well-defined due to the fact that whenever two histories $q_0 \dots q_n, q'_0 \dots q'_m$ result in the same updated a-view, i.e., $[[q_0 \dots q_n]]_a = [[q'_0 \dots q'_m]]_a$, then the last states in these histories lie in the same a-view, i.e., $[q_n]_a = [q'_m]_a$. We now clearly have $out([q]_{E_A}, f_A) = out([q]_{E_A}, \tilde{f}_A)$, and hence all outcomes $\lambda \in out([q]_{E_A}, \tilde{f}_A)$ fulfill the satisfaction condition for $\bigcirc\varphi_1$, $\Box\,\varphi_1$, or $\varphi_1\,\mathcal{U}\,\varphi_2$, respectively. Thus it holds that $q \models \langle\langle A\rangle\rangle_{i\bar{R}}\varphi$.

The second part of the proposition is trivial. For example, in Indiana Jones' treasure hunt (see Example 4.7 and Figure 4.1 on page 33), we have $d_0 \models \langle\langle 1\rangle\rangle_{i\bar{R}}\top\,\mathcal{U}\,rich$, but $d_0 \not\models \langle\langle 1\rangle\rangle_{ir}\top\,\mathcal{U}\,rich$. ☐

6.1.4 Decidability

Proposition 6.11. *The model checking problem for $\mathrm{ATL}_{i\bar{R}}$ is decidable.*

Proof. This is actually trivial, since there are only finitely many $i\bar{R}$ strategies for any coalition A in any CEGS $\mathcal{G} = \langle \Pi, \Sigma, Q, (\sim_a)_{a\in\Sigma}, \pi, Act, d, o\rangle$. We sketch a "brute force" algorithm which checks the satisfaction of a formula $\langle\langle A\rangle\rangle_{i\bar{R}}\varphi$ in a state q_0: From $\mathcal{G}$ we construct a CGS $\mathcal{G}' = \langle \Pi', \Sigma', Q', \pi', Act', d', o'\rangle$, which, in addition to the states and transitions of $\mathcal{G}$, encodes the updated a-views of the players in A during the game. We construct $\mathcal{G}'$ in the following way:

- $\Pi' := \Pi$, $\Sigma' := \Sigma$, and $Act' := Act$.
- $Q' = \{(q, (S_a)_{a\in A}) \mid q \in Q, S_a \subseteq [q]_a \text{ for all } a \in A\}$, i.e., Q' contains all possible tuples of states q (which encode the actual states of the game) together with the updated a-views of the players in A.
- The labelling function $\pi' : Q' \to \Pi'$ is defined by $\pi'((q, (S_a)_{a\in A})) := \pi(q)$ for all $(q, (S_a)_{a\in A}) \in Q'$, i.e., the labelling is just "copied" from $\mathcal{G}$ since it does not depend on what the players believe.

- The function $d' : \Sigma' \times Q' \to 2^{Act'} \setminus \{\emptyset\}$ which determines the enabled actions for each player in each state is defined by $d'(a, (q, (S_a)_{a\in A})) := d(a, q)$ for all players $a \in \Sigma'$ and all $(q, (S_a)_{a\in A}) \in Q'$, since it also does not depend on the updated a-views.

- The transition function o' which determines for each state $(q, (S_a)_{a\in A}) \in Q'$ and each tuple $(\alpha_a)_{a\in\Sigma}$ of enabled actions the next state of the game is defined in the following way: $o'((q, (S_a)_{a\in A}), (\alpha_a)_{a\in\Sigma}) := (\tilde{q}, (\tilde{S}_a)_{a\in A})$, where
  - $\tilde{q} := o(q, (\alpha_a)_{a\in\Sigma})$, i.e., the actual next state of the game is just computed like in $\mathcal{G}$, and
  - $\tilde{S}_a := \{t \in [\tilde{q}]_a \,|\, t \text{ is reachable (in } \mathcal{G}) \text{ from a state } s \in S_a\}$ for all $a \in A$, i.e., $\tilde{S}_a$ encodes for each player a the updated a-view after the transition from q to $\tilde{q}$ if in q the updated a-view was S_a.

By this construction, it is immediate that the computations λ in $\mathcal{G}$ starting in a state s (i.e., $\lambda[0] = s$) are in bijection with the computations λ' in $\mathcal{G}'$ starting in $(s, ([s]_a)_{a\in A})$, via

$$\lambda'[i] = (\lambda[i], ([[\lambda[0..i]]]_a)_{a\in A}). \tag{6.1}$$

Given any $i\bar{R}$ strategy $f_A = (f_a)_{a\in A}$ for coalition A in $\mathcal{G}$, we construct an ir strategy $f'_A = (f'_a)_{a\in A}$ for A in $\mathcal{G}'$ by setting $f'_{a_0}((q, (S_a)_{a\in A})) := f_{a_0}(S_{a_0})$ for all $a_0 \in A$ and all $(q, (S_a)_{a\in A}) \in Q'$. This strategy clearly simulates f_A, i.e., a computation λ is in $out(s, f_A)$ in $\mathcal{G}$ iff the corresponding computation λ' (as defined in (6.1)) is in $out((s, ([s]_a)_{a\in A}), f'_A)$.

Hence, in order to check whether the formula $\langle\langle A\rangle\rangle_{i\bar{R}}\varphi$ holds in state q_0 of $\mathcal{G}$, we can now, for each of the (finitely many) $i\bar{R}$ strategies f_A for A in $\mathcal{G}$,

1. construct f'_A from f_A,
2. trim $\mathcal{G}'$ according to f'_A, i.e., by removing all transitions which are avoided if the players of A stick to the strategy f'_A,
3. use a CTL model checking algorithm in order to check whether the formula φ holds in all states $(q, ([q]_a)_{a\in A})$ with $q \sim_{E_A} q_0$.

If the CTL checks returns "true" for one of the strategies f'_A, then the corresponding strategy f_A in $\mathcal{G}$ is a witness for the satisfaction of $\langle\langle A\rangle\rangle_{i\bar{R}}\varphi$. If the procedure always returns "false", then the formula $\langle\langle A\rangle\rangle_{i\bar{R}}\varphi$ does not hold in q_0. $\square$

Remark 6.12. The CGS $\mathcal{G}'$ in the above proof will in general be exponentially larger than $\mathcal{G}$, and it will contain many states which are not even reachable from the states $(q, ([q]_a)_{a\in A})$ with $q \sim_{E_A} q_0$ (and hence are not needed for the CTL check). Of course, it would be an interesting task to construct a more efficient algorithm which does not create so many unnecessary states, but we refrain from doing so at this point. ⌟

6.2 ATL$_{iR^C}$

We now introduce ATL$_{iR^C}$, which uses the idea from [DEG10] (see Section 4.3) to use "imperfect information and perfect recall strategies for communicating coalitions". We adapt this idea to concurrent epistemic game structures in order to obtain an upper bound for the satisfaction of ATL$_{iR}$ formulas.

In the following, we let Q_A denote the partition of Q induced by the equivalence relation $\sim_{D_A}$, i.e., $Q_A := \{[q]_{D_A} \mid q \in Q\}$.

Definition 6.13 (iR^C strategy). *An* imperfect information and perfect recall strategy with communication *(for short, an iR^C* strategy*) for a coalition $A \subseteq \Sigma$ in a CEGS $\mathcal{G} = \langle \Pi, \Sigma, Q, (\sim_a)_{a\in\Sigma}, \pi, Act, d, o\rangle$ is a tuple $(f_a)_{a\in A}$ of functions $f_a : Q_A^+ \to Act$ with the property that for each player $a \in A$ and every sequence $\lambda = [q_0]_{D_A} \ldots [q_n]_{D_A} \in Q_A^+$ the action $f_a(\lambda)$ is enabled for a in the states of $[q_n]_{D_A}$.*

As always, we can view each component f_a of an iR^C strategy as a function $f_a : Q^+ \to Act$ by defining, for each $\lambda = q_0q_1 \ldots q_n \in Q^+$,

$$f_a(\lambda) := f_a([q_0]_{D_A}[q_1]_{D_A} \ldots [q_n]_{D_A}),$$

and then the set of outcomes $out(q, f_A)$ from state q of an iR^C strategy $f_A = (f_a)_{a\in A}$ for coalition A is properly defined by Definition 2.17.

An iR^C strategy allows a coalition A to base their strategy on their view of the game history with respect to their distributed knowledge. This corresponds to the idea that the players in A communicate and exchange their knowledge while following their strategy. Like explained in Section 4.3.4, this requires a somewhat artificial interpretation of how such strategies are executed. However, the exact interpretation is actually not important for our purpose since our main aim is the approximation of ATL$_{iR}$. As we will see, the fact that in an iR^C strategy each of the individual strategies for the players depends on the same information makes the model checking problem decidable, in contrast to ATL$_{iR}$ where the problem seems to be related to the fact that a strategy for a coalition of players is just a tuple of independent individual strategies.

6.2.1 Syntax

The syntax of ATL_{iR^C} is exactly the same as the one of ATL_{iR}. As always, we will make the distinction to the other logics explicit by marking the cooperation modalities with a subscript, i.e., ATL_{iR^C} formulas starting with a cooperation modality are of the form $\langle\langle A\rangle\rangle_{iR^C} \bigcirc \varphi$, $\langle\langle A\rangle\rangle_{iR^C} \Box \varphi$, and $\langle\langle A\rangle\rangle_{iR^C} \varphi_1 \mathcal{U} \varphi_2$.

6.2.2 Semantics

Formulas of ATL_{iR^C}, defined with respect to the set Π of propositions and the set $\Sigma = \{1, \dots, k\}$ of players, are interpreted in the states of a concurrent epistemic game structure $\mathcal{G} = \langle \Pi, \Sigma, Q, (\sim_a)_{a\in\Sigma}, \pi, Act, d, o\rangle$. The satisfaction of formulas of the form p ($p \in \Pi$), $\neg\varphi$, and $\varphi_1 \vee \varphi_2$ is defined exactly like in ATL and ATL_{iR} (see (S_p), ($S_\neg$), and ($S_\vee$) in Section 3.2). For the other cases, the definition of the satisfaction relation $\models$ differs from the one for ATL_{iR} in the type of strategies which are used and in the set of states from which a strategy has to be successful:

($S'_\bigcirc$) $q \models \langle\langle A\rangle\rangle_{iR^C} \bigcirc \varphi$ iff there exists an iR^C strategy $f_A = (f_a)_{a\in A}$ for the players in A, such that all outcomes $\lambda \in out([q]_{D_A}, f_A)$ satisfy $\lambda[1] \models \varphi$.

($S'_\Box$) $q \models \langle\langle A\rangle\rangle_{iR^C} \Box \varphi$ iff there exists an iR^C strategy $f_A = (f_a)_{a\in A}$ for the players in A, such that for all outcomes $\lambda \in out([q]_{D_A}, f_A)$ and all positions $i \geq 0$ we have $\lambda[i] \models \varphi$.

($S'_\mathcal{U}$) $q \models \langle\langle A\rangle\rangle_{iR^C} \varphi_1 \mathcal{U} \varphi_2$ iff there exists an iR^C strategy $f_A = (f_a)_{a\in A}$ for the players in A, such that for all outcomes $\lambda \in out([q]_{D_A}, f_A)$ there exists a position $i \geq 0$ such that $\lambda[i] \models \varphi_2$ and for all positions $0 \leq j < i$ we have $\lambda[j] \models \varphi_1$.

Hence, a formula of the form $\langle\langle A\rangle\rangle_{iR^C}\varphi$ is satisfied iff a perfect recall strategy for A depending on their distributed knowledge exists, of which the players in A know (also by their distributed knowledge) that it will ensure φ.

Example 6.14. Consider the generals problem described in Example 5.5 and depicted in Figure 5.1 on page 46. The formula $\langle\langle 1,2\rangle\rangle_{iR^C} \bigcirc win$ is satisfied in the states SS, Ss, and sS, i.e., in each of these states the generals will win the fight if they have a virtual supervisor which instructs them to attack as soon as one of them signals the success of his spy. ⚁

Example 6.15. Once more, we consider the chess game from Example 2.4 and Figure 2.2 on page 11. In this game, we have $a1 \not\models \langle\langle 1,2\rangle\rangle_{iR^C} \neg p_2 \mathcal{U} (p_a \wedge p_3)$, i.e., from state $a1$ it is not possible to reach $a3$ without crossing the second row. However, the players can reach $a3$ if they are allowed to cross row two in column d, i.e., $a1 \models \langle\langle 1, 2\rangle\rangle_{iR^C} (p_d \vee \neg p_2) \,\mathcal{U}\, (p_a \wedge p_3)$. ⚁

6.2.3 Relation to ATL$_{iR}$ and ATL$_{IR}$

In this section we show the implications

$$\langle\langle A\rangle\rangle_{iR}\varphi \quad \Rightarrow \quad \langle\langle A\rangle\rangle_{iR^C}\varphi \quad \underset{\not\Leftarrow}{\Rightarrow} \quad \langle\langle A\rangle\rangle_{IR}\varphi$$

Proposition 6.16. *ATL$_{iR^C}$ approximates ATL$_{iR}$ from above, in the sense that for each CEGS $\mathcal{G} = \langle \Pi, \Sigma, Q, (\sim_a)_{a\in\Sigma}, \pi, Act, d, o\rangle$, each state $q \in Q$, and each φ of the form $\bigcirc\varphi_1$, $\Box\varphi_1$, or $\varphi_1 \,\mathcal{U}\, \varphi_2$, where φ_1 and φ_2 are propositional formulas, we have the implication*

$$q \models \langle\langle A\rangle\rangle_{iR}\varphi \quad \Rightarrow \quad q \models \langle\langle A\rangle\rangle_{iR^C}\varphi.$$

Proof. This simply holds because every tuple $(f_a)_{a\in A}$ of iR strategies is an iR^C strategy for A (if we view all individual strategies as functions $f_a : Q^+ \to Act$), and because $\sim_{D_A} \subseteq \sim_{E_A}$.

More formally: Suppose that $q \models \langle\langle A\rangle\rangle_{iR}\varphi$, i.e., there is an iR strategy $f_A = (f_a)_{a\in A}$ for the players in A, such that all outcomes $\lambda \in out([q]_{E_A}, f_A)$ fulfill the required condition for the satisfaction of $\bigcirc\varphi_1$, $\Box\varphi_1$, or $\varphi_1 \,\mathcal{U}\, \varphi_2$, respectively. We define an iR^C strategy $\tilde{f}_A = (\tilde{f}_a)_{a\in A}$ for A by setting, for any $[q_0]_{D_A} \dots [q_n]_{D_A} \in Q_A^+$,

$$\tilde{f}_a([q_0]_{D_A} \dots [q_n]_{D_A}) := f_a([q_0]_a \dots [q_n]_a).$$

Note that $\tilde{f}_a$ is well-defined due to the fact that whenever two states q_i, q_i' are $\sim_{D_A}$-indistinguishable, i.e., $[q_i]_{D_A} = [q_i']_{D_A}$, then they are also $\sim_a$-indistinguishable, i.e., $[q_i]_a = [q_i']_a$, for all $a \in A$.

Now, it clearly holds that $out([q]_{E_A}, f_A) = out([q]_{E_A}, \tilde{f}_A)$ due to the construction of $\tilde{f}_A$, and moreover $out([q]_{E_A}, \tilde{f}_A) \supseteq out([q]_{D_A}, \tilde{f}_A)$, since $\sim_{E_A} \supseteq \sim_{D_A}$. Hence all outcomes $\lambda \in out([q]_{D_A}, \tilde{f}_A)$ fulfill the satisfaction condition for $\bigcirc\varphi_1$, $\Box\varphi_1$, or $\varphi_1 \,\mathcal{U}\, \varphi_2$, respectively. Thus we have $q \models \langle\langle A\rangle\rangle_{iR^C}\varphi$. □

Remark 6.17. If φ is an ATL$_{iR}$ formula and φ^C results from φ by replacing all occurrences of $\langle\langle A\rangle\rangle_{iR}$ by $\langle\langle A\rangle\rangle_{iR^C}$, it does in general *not* hold that $q \models \varphi$ implies $q \models \varphi^C$. The proof is left to the reader. ⌟

Proposition 6.18. *ATL$_{iR^C}$ approximates ATL$_{iR}$ strictly better than ATL$_{IR}$, i.e., for each CEGS $\mathcal{G} = \langle \Pi, \Sigma, Q, (\sim_a)_{a\in\Sigma}, \pi, Act, d, o\rangle$, each state $q \in Q$, and each φ of the form $\bigcirc\varphi_1$, $\Box\varphi_1$, or $\varphi_1 \mathcal{U} \varphi_2$, where φ_1 and φ_2 are propositional formulas, we have the implication*

$$q \models \langle\langle A\rangle\rangle_{iR^C}\varphi \quad \Rightarrow \quad q \models \langle\langle A\rangle\rangle_{IR}\varphi,$$

whereas, in general,

$$q \models \langle\langle A\rangle\rangle_{iR^C}\varphi \quad \not\Leftarrow \quad q \models \langle\langle A\rangle\rangle_{IR}\varphi.$$

Proof. The implication $q \models \langle\langle A\rangle\rangle_{iR^C}\varphi \Rightarrow q \models \langle\langle A\rangle\rangle_{IR}\varphi$ follows from the fact that every iR^C strategy $(f_a)_{a\in A}$ for a coalition A is also an IR strategy for A (if we view all individual strategies as functions $f_a : Q^+ \to Act$).

More formally: Suppose that $q \models \langle\langle A\rangle\rangle_{iR^C}\varphi$, i.e., there is an iR^C strategy $f_A = (f_a)_{a\in A}$ for A, such that all outcomes $\lambda \in out([q]_{D_A}, f_A)$ fulfill the required condition for the satisfaction of $\bigcirc\varphi_1$, $\Box\varphi_1$, or $\varphi_1 \mathcal{U} \varphi_2$, respectively. We define an IR strategy $\tilde{f}_A = (\tilde{f}_a)_{a\in A}$ for A by setting, for any $q_0 \ldots q_n \in Q^+$,

$$\tilde{f}_a(q_0 \ldots q_n) := f_a([q_0]_{D_A} \ldots [q_n]_{D_A}).$$

Clearly, we have $out(q, \tilde{f}_A) \subseteq out([q]_{D_A}, \tilde{f}_A) = out([q]_{D_A}, f_A)$, and hence all outcomes $\lambda \in out(q, \tilde{f}_A)$ fulfill the condition for the satisfaction of $\bigcirc\varphi_1$, $\Box\varphi_1$, or $\varphi_1 \mathcal{U} \varphi_2$, respectively. Thus it holds that $q \models \langle\langle A\rangle\rangle_{IR}\varphi$.

The second part of the proposition is trivial. For example, in Indiana Jones' treasure hunt (see Figure 4.1 on page 33), we have $d_0 \models \langle\langle 1\rangle\rangle_{IR} \bigcirc rich$, but on the other hand $d_0 \not\models \langle\langle 1\rangle\rangle_{iR^C} \bigcirc rich$. $\Box$

6.2.4 Decidability

Proposition 6.19. *The model checking problem for ATL$_{iR^C}$ is decidable.*

Proof. We reduce the model checking problem for ATL$_{iR^C}$ to the model checking problem for ATL$^D_{iR}$, which is decidable (see Theorem 4.19):

Let φ be an arbitrary ATL$_{iR^C}$ formula, $\mathcal{G} = \langle \Pi, \Sigma, Q, (\sim_a)_{a\in\Sigma}, \pi, Act, d, o\rangle$ a concurrent epistemic game structure, and s_0 a state in $\mathcal{G}$ for which we want to check whether $\mathcal{G}, s_0 \models \varphi$. We start with three observations:

- First of all, it suffices to consider the case where the formula φ is of the form $\langle\langle A\rangle\rangle_{iR^C} \bigcirc \varphi_1$, $\langle\langle A\rangle\rangle_{iR^C} \Box\varphi_1$, or $\langle\langle A\rangle\rangle_{iR^C} \varphi_1 \mathcal{U} \varphi_2$, where φ_1 and φ_2 are propositional formulas. This is due to the fact that we can evaluate every ATL$_{iR^C}$ formula φ "bottom-up", i.e., by first determining (recursively) for each subformula

φ' of φ the set of states which satisfy φ', adding a new propositional variable $p_{\varphi'}$ to the model in order to "store" this information, and just replacing φ' in φ by $p_{\varphi'}$.

- Secondly, we may assume that for each player a and every state q all actions $\alpha \in Act$ are enabled for a in q, i.e., $d(a, q) = Act$. If this is not yet the case, we can modify $\mathcal{G}$ by adding additional transitions which do not change the powers of any player. This can be done by repeatedly applying the following procedure:

  – Choose a player a and a state q for which $d(a, q) \neq Act$.

  – Choose an action $\alpha^- \in Act \setminus d(a, q)$ and an action $\alpha^+ \in d(a, q)$ (remember that $d(a, q)$ is always non-empty).

  – For each tuple of actions $(\alpha_{\tilde{a}})_{\tilde{a}\in\tilde{\Sigma}}$ which are enabled for the players $\tilde{a} \in \tilde{\Sigma} := \Sigma \setminus \{a\}$ in q, set $o(q, (\alpha^-, (\alpha_{\tilde{a}})_{\tilde{a}\in\tilde{\Sigma}})) := o(q, (\alpha^+, (\alpha_{\tilde{a}})_{\tilde{a}\in\tilde{\Sigma}}))$.

  – Add α^- to $d(a, q)$.

  It is clear that this procedure does not change the value of any formula in any state of $\mathcal{G}$, since it basically assigns "alternative names" to already existing transitions (of course, in general the "intended meaning" of the actions will not be preserved, but this is irrelevant at this point). Also note that while repeatedly executing this procedure, the intermediary result will not always be a valid CEGS since the requirement $d(a, q) = d(a, q')$ for $q \sim_a q'$ may be violated. However, when we finally have constructed a $\mathcal{G}$ where $d(a, q) = Act$ for all players a and all states q, this property is assured.

- Furthermore, we may assume that for each player a and each a-view $[q]_a \in Q_a$, Π contains an atomic proposition $p_{[q]_a}$ which is true exactly in the states of $[q]_a$, i.e., for all $q' \in Q$ we have $p_{[q]_a} \in \pi(q')$ iff $q' \in [q]_a$. If $\mathcal{G}$ does not have this property, we can just introduce new variables and add them to $\mathcal{G}$ accordingly (by adding new variables, we certainly do not change the value of φ in s_0).

  We will later use this assumption in order to be able to drop the equivalence relations $(\sim_a)_{a\in\Sigma}$ without losing any information about the game structure.

We now construct a game arena $\mathcal{G}' = \langle \Sigma', Q', (Act'_a)_{a\in\Sigma'}, \delta', Q'_0, (\Pi'_a)_{a\in\Sigma'}, \pi' \rangle$ from the above CEGS $\mathcal{G}$ in the following way:

- $\mathcal{G}'$ contains the exact same states as $\mathcal{G}$, i.e., $Q' := Q$.

- The labelling of the states in $\mathcal{G}'$ is the same as in $\mathcal{G}$, i.e., $\pi' := \pi$.

- Σ' is created from Σ by adding an additional player (we will denote this player by 0, i.e., $\Sigma' := \Sigma \cup \{0\}$). The only purpose of this player will be to observe all propositions, since in a game arena every proposition has to be observable to at least one player.

- For each player $a \in \Sigma'$, Act'_a contains all actions of $\mathcal{G}$, i.e., $Act'_a := Act$.

- Each player $a \neq 0$ can observe exactly the propositions $p_{[q]_a}$ from the above assumption, i.e., $\Pi'_a := \{p_{[q]_a} \mid q \in Q\}$. Player 0 can observe all variables of Π, i.e., $\Pi'_0 := \Pi$. This way, we have $\Pi' := \bigcup_{a\in\Sigma'} \Pi'_a = \Pi$. We use the same abbreviations as in Definition 4.10, i.e., $\Pi'_A := \bigcup_{a\in A} \Pi'_a$ and $\pi'_A(q) := \pi'(q) \cap \Pi'_A$.

  The definitions above ensure that $q \sim_a q'$ in $\mathcal{G}$ iff player a cannot distinguish q from q' in $\mathcal{G}'$ by the values of the variables in Π'_a, i.e., $q \sim_a q'$ iff $\pi'_a(q) = \pi'_a(q')$. Moreover, for any coalition $A \subseteq \Sigma$, we have $q \sim_{D_A} q'$ iff $\pi'_A(q) = \pi'_A(q')$.

- The set of initial states Q'_0 consists of those states which are for coalition A indistinguishable from q_0 with respect to their distributed knowledge (in $\mathcal{G}$), i.e., $Q'_0 := [s_0]_{D_A}$.

- The transition relation δ' (which is a transition function in this case) just "copies" the transitions from $\mathcal{G}$ by ignoring the action of player 0, via $\delta'(q, (\alpha_a)_{a\in\Sigma'}) := \{o(q, (\alpha_a)_{a\in\Sigma})\}$. Note that this is possible because $Act_a = Act$ for all $a \in \Sigma$ and due to the assumption that at each state in $\mathcal{G}$ all actions are enabled for every player. Moreover, this ensures that player 0 has no influence on the game at all.

We now prove that the given formula φ (i.e., $\langle\langle A\rangle\rangle_{iR^C} \bigcirc \varphi_1$, $\langle\langle A\rangle\rangle_{iR^C} \Box \varphi_1$, or $\langle\langle A\rangle\rangle_{iR^C} \varphi_1 \mathcal{U} \varphi_2$, respectively) holds in state s_0 of $\mathcal{G}$ iff the formula φ^D which results from φ by replacing the cooperation modality by $\langle\langle A\rangle\rangle_{iR^D}$ (i.e., $\langle\langle A\rangle\rangle_{iR^D} \bigcirc \varphi_1$, $\langle\langle A\rangle\rangle_{iR^D} \Box \varphi_1$, or $\langle\langle A\rangle\rangle_{iR^D} \varphi_1 \mathcal{U} \varphi_2$, respectively) is valid in $\mathcal{G}'$. In the following, we write $\widetilde{out}(f'_A)$ for the infinite sequences of states which can be constructed by dropping the action tuples from the runs in $out(f'_A)$, i.e., a sequence $q_0q_1q_2\dots$ is contained in $\widetilde{out}(f'_A)$ iff there exists a run $q_0 \xrightarrow{c_1} q_1 \xrightarrow{c_2} q_2 \xrightarrow{c_3} \dots \in out(f'_A)$.

"$\Rightarrow$": Suppose that φ holds in state s_0 of $\mathcal{G}$. Then there is an iR^C strategy $f_A = (f_a)_{a\in A}$ for A with the property that all outcomes $\lambda \in out([s_0]_{D_A}, f_A)$ fulfill the required condition for the satisfaction of $\bigcirc\varphi_1$, $\Box \varphi_1$, or $\varphi_1 \mathcal{U} \varphi_2$, respectively. We construct an iR^D strategy f'_A for A in $\mathcal{G}'$ by setting, for each sequence $\pi'_A(q_0)\dots\pi'_A(q_n) \in (2^{\Pi'_A})^+$,

$$f'_A(\pi'_A(q_0)\dots\pi'_A(q_n)) := (f_a([q_0]_{D_A}\dots[q_n]_{D_A}))_{a\in A}.$$

Note that f'_A is well-defined due to the construction of the sets Π'_a, i.e., whenever $\pi'_A(q_i) = \pi'_A(q'_i)$, then $[q_i]_{D_A} = [q'_i]_{D_A}$. By this definition, f'_A simulates f_A, i.e., given any history λ, f'_A dictates coalition A to choose in $\mathcal{G}'$ the same actions as f_A does in $\mathcal{G}$. Since moreover $Q'_0 = [s_0]_{D_A}$ by definition, each sequence of states λ is in $out([s_0]_{D_A}, f_A)$ (in $\mathcal{G}$) iff it is in $\widetilde{out}(f'_A)$ (in $\mathcal{G}'$). Hence, by the assumption that all outcomes $\lambda \in out([s_0]_{D_A}, f_A)$ fulfill the required condition for the satisfaction of $\bigcirc\varphi_1$, $\Box\,\varphi_1$, or $\varphi_1\,\mathcal{U}\,\varphi_2$, respectively, and due to fact that $\pi = \pi'$, also every $\lambda \in out(f'_A)$ which satisfies $\lambda[0] \sim_A \lambda'[0]$ for some initialized infinite run λ' in $\mathcal{G}'$ fulfills the required condition for the satisfaction of $\bigcirc\varphi_1$, $\Box\,\varphi_1$, or $\varphi_1\,\mathcal{U}\,\varphi_2$, respectively. This shows that every initialized infinite run λ' in $\mathcal{G}'$ satisfies $(\mathcal{G}', \lambda', 0) \models \langle\langle A\rangle\rangle_{iR^D} \bigcirc \varphi_1$, $(\mathcal{G}', \lambda', 0) \models \langle\langle A\rangle\rangle_{iR^D} \Box\,\varphi_1$, or $(\mathcal{G}', \lambda', 0) \models \langle\langle A\rangle\rangle_{iR^D}\, \varphi_1\,\mathcal{U}\,\varphi_2$, respectively, i.e., φ^D is valid in $\mathcal{G}'$.

"$\Leftarrow$": Now suppose that φ^D is valid in $\mathcal{G}'$, i.e., $(\mathcal{G}', \lambda', 0) \models \langle\langle A\rangle\rangle_{iR^D} \bigcirc \varphi_1$, $(\mathcal{G}', \lambda', 0) \models \langle\langle A\rangle\rangle_{iR^D} \Box\,\varphi_1$, or $(\mathcal{G}', \lambda', 0) \models \langle\langle A\rangle\rangle_{iR^D}\, \varphi_1\,\mathcal{U}\,\varphi_2$, respectively, for every initialized infinite run λ' in $\mathcal{G}'$ (especially for some λ' which starts in s_0). It follows that there exists an iR^D strategy f'_A for A in $\mathcal{G}'$ with the property that every $\lambda \in out(f'_A)$ which satisfies $\lambda[0] \sim_A s_0$ (and hence, *every* $\lambda \in out(f'_A)$, because of the construction of the sets Π'_a and due to the fact that $Q'_0 = [s_0]_{D_A}$) fulfills the required condition for the satisfaction of $\bigcirc\varphi_1$, $\Box\,\varphi_1$, or $\varphi_1\,\mathcal{U}\,\varphi_2$, respectively. We construct an iR^C strategy $f_A = (f_a)_{a\in A}$ for A in $\mathcal{G}$ by setting, for each $a \in A$,

$$f_a([q_0]_{D_A} \dots [q_n]_{D_A}) := f'_A(\pi'_A(q_0) \dots \pi'_A(q_n))|_a.$$

Again, this strategy is well-defined due to the construction of the sets Π'_a. By this definition, f_A assigns the same actions to every history in $\mathcal{G}$ as f'_A does in $\mathcal{G}'$, and since moreover $[s_0]_{D_A} = Q'_0$ it follows that $out([s_0]_{D_A}, f_A) = \widetilde{out}(f'_A)$. Since furthermore $\pi = \pi'$, it follows that every $\lambda \in out([s_0]_{D_A}, f_A)$ fulfills the required condition for the satisfaction of $\bigcirc\varphi_1$, $\Box\,\varphi_1$, or $\varphi_1\,\mathcal{U}\,\varphi_2$, respectively, and hence $\mathcal{G}, s_0 \models \varphi$. $\Box$

CHAPTER 7

Conclusion

In this thesis we have considered logics for coalitional games of imperfect information. An overview of these logics and their most important properties is shown in Table 7.1 on the following page.

We have seen that under the assumption of imperfect recall very expressive logics exist which allow one to describe various kinds of strategic abilities of players together with different notions of group knowledge. Most notably CSL, which exploits the idea of using constructive knowledge modalities in addition to the usual ones, is a very elegant solution for this purpose which is at the same time not harder in terms of model checking than the other suggestions.

Under the assumption of perfect recall things are more difficult: Either one uses uniform strategies like in ATL_{iR}, which results in the undecidability of the model checking problem, or one abandons the uniformity of strategies like in ATEL, which leads to the problem that only a very weak notion of strategic ability can be described. As a compromise one can consider uniform strategies based on distributed knowledge as it is done in ATL_{iR}^{D}, which gives rise to a decidable model checking problem but requires an interpretation which seems to some extent artificial.

As our own contribution to the topic we have presented two approximations to ATL_{iR}, which can at least in some cases be used to determine the truth values of ATL_{iR} formulas, despite the fact that the model checking problem for this logic is undecidable. We have left open the question how good these approximations are. This could be a topic for future research, where of course the first question that arises is in what terms the quality of such approximations could be measured.

Logic	Recall	Game structure	Expressible types of strategic abilities	Expressible types of group knowledge	Model checking
ATEL	perfect	AETS	(A1)	E, C	P-complete
ATL_{iR}	perfect	CEGS	(A4) with group knowledge E	E (indirectly)	undecidable
ATL_{iR}^{D}	perfect	Game arena	(A4) with group knowledge D	D	decidable
ATL_{ir}	imperfect	CEGS	(A4) with group knowledge E	E (indirectly)	Δ_2^P-complete
ATOL	imperfect	CEGS	(A2), (A3), (A4) with group knowledge E, D, C	E, D, C	Δ_2^P-complete
CSL	imperfect	CEGS	(A2), (A3), (A4) with group knowledge E, D, C	E, D, C + constructive knowledge	Δ_2^P-complete

Table 7.1: Overview of the considered logics for coalitional games of imperfect information. E, D, and C denote "everyone's knowledge", "distributed knowledge", and "common knowledge", respectively.

APPENDIX A

Complexity classes

We list the complexity classes which are referred to in this thesis. Rigorous definitions of all used terms and more details can be found, e.g., in [Pap94].

We denote by DTIME$(f(n))$ the class of problems which can be solved by a deterministic Turing machine, and by NTIME$(f(n))$ the class of problems which can be solved by a nondeterministic Turing machine, respectively, in time $O(f(n))$ if the size of the input is n. From this, the complexity classes P, NP, EXPTIME, and 2-EXPTIME are defined in the following way:

$$\mathrm{P} := \bigcup_{k\in\mathbb{N}} \mathrm{DTIME}(n^k), \qquad \mathrm{NP} := \bigcup_{k\in\mathbb{N}} \mathrm{NTIME}(n^k),$$
$$\mathrm{EXPTIME} := \bigcup_{k\in\mathbb{N}} \mathrm{DTIME}\left(2^{n^k}\right), \qquad \text{2-EXPTIME} := \bigcup_{k\in\mathbb{N}} \mathrm{DTIME}\left(2^{2^{n^k}}\right).$$

Moreover, if $\mathcal{C}$ and $\mathcal{D}$ are two (deterministic or nondeterministic) time complexity classes, we let $\mathcal{C}^{\mathcal{D}}$ denote the class of problems which can be solved by machines of the same sort and time bound as in $\mathcal{C}$, only that the machines are allowed to use a $\mathcal{D}$-machine as an oracle (i.e., $\mathcal{C}^{\mathcal{D}}$-machines are allowed to solve at each time step a problem of the class $\mathcal{D}$, whose computation cost is not counted). From this, the complexity classes Σ_i^P and Δ_i^P are inductively defined, for $i \in \mathbb{N}$, via $\Sigma_0^P := \Delta_0^P :=$ P, and

$$\Sigma_{i+1}^P := \mathrm{NP}^{\Sigma_i^P}, \qquad \Delta_{i+1}^P := \mathrm{P}^{\Sigma_i^P}.$$

In particular, $\Delta_2^P = \mathrm{P}^{\mathrm{NP}}$.

APPENDIX B

CTL

Computation Tree Logic (CTL, introduced in [CE82]) is defined over a set Π of atomic propositions by the following grammar:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathrm{A}\,\mathrm{X}\,\varphi \mid \mathrm{E}\,\mathrm{X}\,\varphi \mid \mathrm{A}[\varphi\,\mathrm{U}\,\varphi] \mid \mathrm{E}[\varphi\,\mathrm{U}\,\varphi],$$

where p ranges over the set Π. The additional boolean connectives $\vee$ and $\rightarrow$ as well as the truth constants $\top$ and $\perp$ are defined as usual.

A ("all") and E ("exists") are quantifiers referring to computations, while X ("next") and U ("until") are temporal operators. In addition, an operator G ("globally") can be defined by $\mathrm{A}\,\mathrm{G}\,\varphi := \neg\,\mathrm{E}[\top\,\mathrm{U}\,\neg\varphi]$, and $\mathrm{E}\,\mathrm{G}\,\varphi := \neg\,\mathrm{A}[\top\,\mathrm{U}\,\neg\varphi]$.

CTL formulas are interpreted in *finite Kripke structures*. Such a structure is a triple $\langle Q, R, \pi\rangle$ with the following components:

- Q is a finite set of *states*.
- $R \subseteq S \times S$ is an *accessibility relation* which describes possible transitions between states. It is required that R is a serial relation, i.e., for each $q_1 \in Q$ there exists a $q_2 \in Q$ such that $q_1 R q_2$.
- $\pi : Q \rightarrow 2^{\Pi}$ is a *labelling function* which maps each state to the set of propositions that are true in the state.

A *computation* in a Kripke structure is an infinite sequence $q_0 q_1 q_2 \ldots$ of states with the property that for each $i \geq 0$, q_{i+1} is reachable from q_i (i.e., $q_i R q_{i+1}$).

Given a fixed finite Kripke structure $\langle Q, R, \pi \rangle$, a state $q \in Q$, and a CTL formula φ, we write $q \models \varphi$ to indicate that the state q satisfies the formula φ. The satisfaction relation $\models$ is inductively defined by the following rules:

- $q \models p$, for a proposition $p \in \Pi$, iff $p \in \pi(q)$.
- $q \models \neg\varphi$ iff $q \not\models \varphi$.
- $q \models \varphi_1 \wedge \varphi_2$ iff $q \models \varphi_1$ and $q \models \varphi_2$.
- $q \models \mathrm{A}\,\mathrm{X}\,\varphi$ iff all states q' with qRq' satisfy $q' \models \varphi$.
- $q \models \mathrm{E}\,\mathrm{X}\,\varphi$ iff there is a state q' with qRq' which satisfies $q' \models \varphi$.
- $q \models \mathrm{A}[\varphi_1\,\mathrm{U}\,\varphi_2]$ iff for all computations $q_0q_1q_2\ldots$ with $q_0 = q$ there exists an $i \geq 0$ such that $q_i \models \varphi_2$ and for all $0 \leq j < i$ we have $q_j \models \varphi_1$.
- $q \models \mathrm{E}[\varphi_1\,\mathrm{U}\,\varphi_2]$ iff there is a computation $q_0q_1q_2\ldots$ with $q_0 = q$ with the property that there exists an $i \geq 0$ such that $q_i \models \varphi_2$ and for all $0 \leq j < i$ we have $q_j \models \varphi_1$.

A nice property of CTL is the fact that it can be completely axiomatized:

Theorem B.1 (Axiomatizability of CTL [EH85]). *CTL is completely axiomatizable, i.e., there is an axiomatic system for CTL which is sound and complete with respect to the semantics given above.*

Given a finite Kripke structure $\langle Q, R, \pi \rangle$, a state $q \in Q$ and a CTL formula φ, the model checking problem for CTL asks whether $q \models \varphi$. CTL model checking can be done very efficiently by a *state labelling algorithm*, which, given a CTL formula φ, computes (inductively) for each subformula φ' of φ the set of states which satisfy φ' and labels these states accordingly, such that each occurrence of φ' in φ can then be replaced by a new proposition $p_{\varphi'}$. The details can be found in [CES86].

Theorem B.2 (CTL model checking complexity [CES86, Sch03]). *The model checking problem for CTL is P-complete and can be solved in time $O(m \cdot \ell)$ for a Kripke structure with m transitions and a CTL formula of length ℓ.*

Notation

2^M	power set of M.
Q^+	set of non-empty finite sequences of elements of Q, $Q^+ := \bigcup_{k\geq 1} Q^k$.
Q^*	set of finite sequences of elements of Q, $Q^* := \bigcup_{k\geq 0} Q^k$.
$\lambda[i]$	state of λ at position i, $\lambda[i] := q_i$, for $\lambda = q_0q_1q_2\ldots$ or $\lambda = q_0 \xrightarrow{c_1} q_1 \xrightarrow{c_2} \ldots$.
$\lambda[0..i]$	praefix of λ, $\lambda[0..i] := q_0\ldots q_i$, for $\lambda = q_0q_1q_2\ldots$, and $\lambda[0..i] := q_0 \xrightarrow{c_1} q_1 \xrightarrow{c_2} \ldots \xrightarrow{c_i} q_i$, for $\lambda = q_0 \xrightarrow{c_1} q_1 \xrightarrow{c_2} \ldots$.
$[q]_a$	$\sim_a$-equivalence class of q.
$\sim_{C_A}, \sim_{E_A}, \sim_{D_A}$	epistemic relations for group knowledge, cf. Definition 2.18.
$[q]_{C_A}, [q]_{E_A}, [q]_{D_A}$, $[S]_{C_A}, [S]_{E_A}, [S]_{D_A}$	set of $\sim_{C_A}/\sim_{E_A}/\sim_{D_A}$-indistinguishable states, cf. Section 2.7.
Q_a	partition of Q induced by $\sim_a$, $Q_a := \{[q]_a \mid q \in Q\}$.
Q_A	partition of Q induced by $\sim_{D_A}$, $Q_A := \{[q]_{D_A} \mid q \in Q\}$.

$out(q, f_A), out(S, f_A)$	set of outcomes of a strategy in an A(E)TS/C(E)GS, cf. Definition 2.17.
$out(f_A)$	set of outcomes of a strategy in game arena, cf. Definition 4.15.
$act(\lambda, i)$	action tuple at position i, $act(\lambda, i) := c_{i+1}$, for $\lambda = q_0 \overset{c_1}{\to} q_1 \overset{c_2}{\to} \ldots$.
$c\|_A$	restriction of c to the players in $A \subseteq \Sigma$, $c\|_A := (c_a)_{a \in A}$, for $c = (c_a)_{a \in \Sigma}$.
Π_A	set of propositions observable to coalition A in a game arena, $\Pi_A := \bigcup_{a \in A} \Pi_a$, cf. Definition 4.10.
π_A	labelling function restricted to the observability of A, $\pi_A(q) := \pi(q) \cap \Pi_A$, cf. Definition 4.10.
$[[\lambda]]_a$	updated a-view given history λ, cf. Definition 6.3.

List of Figures

1.1 A game tree. 1
1.2 An automaton-like game structure. 2
1.3 Another automaton-like game structure. 3
1.4 An automaton-like game structure with imperfect information. 5

2.1 ATS and CGS representation of the coin game. 9
2.2 AETS for the chess game. 11
2.3 CGS for a variant of rock-paper-scissors. 13
2.4 CEGS for the asynchronous rock-paper-scissors variant. 14

4.1 CEGS for Indiana Jones' treasure hunt. 33

5.1 CEGS representation of the generals problem. 46

Bibliography

[Ågo06] Thomas Ågotnes. Action and Knowledge in Alternating-time Temporal Logic. *Synthese*, 149(2):375–407, 2006.

[AHK97] Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. Alternating-time Temporal Logic. In *Proceedings of the 38th IEEE Symposium on Foundations of Computer Science (FOCS 1997)*, pages 100–109. IEEE Computer Society Press, 1997.

[AHK98] Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. Alternating-time Temporal Logic. In *Compositionality: The Significant Difference,* Lecture Notes in Computer Science, volume 1536, pages 23–60. Springer, 1998.

[AHK02] Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. Alternating-time Temporal Logic. *Journal of the ACM*, 49(5):672–713, 2002.

[AHM+98] R. Alur, T. Henzinger, F. Mang, S. Qadeer, S. Rajamani, and S. Tasiran. MOCHA: Modularity in model checking. In *Computer Aided Verification*, volume 1427 of *Lecture Notes in Computer Science*, pages 521–525. Springer, 1998.

[BDJ10] Nils Bulling, Jürgen Dix, and Wojciech Jamroga. Model Checking Logics of Strategic Ability: Complexity. In *Specification and Verification of Multi-agent Systems*, pages 125–159. Springer, 2010.

[Bez98] BezalelPeleg. Effectivity functions, game forms, games, and rights. *Social Choice and Welfare*, 15(1):67–80, 1998.

[CE82] Edmund M. Clarke and E. Allen Emerson. Design and Synthesis of Synchronization Skeletons Using Branching-Time Temporal Logic. In *Logic of Programs*, volume 131 of *Lecture Notes in Computer Science*, pages 52–71. Springer, 1982.

[CES86] E. Clarke, E. Emerson, and A. Sistla. Automatic Verification of Finite-State Concurrent Systems Using Temporal Logic Specifications. *ACM Transactions on Programming Languages and Systems*, 8(2):244–263, 1986.

[DEG10] Cătălin Dima, Constantin Enea, and Dimitar Guelev. Model-Checking an Alternating-time Temporal Logic with Knowledge, Imperfect Information, Perfect Recall and Communicating Coalitions. In *Proceedings of the First Symposium on Games, Automata, Logic, and Formal Verification (GANDALF 2010),* Electronic Proceedings in Theoretical Computer Science, volume 25, pages 103–117, 2010.

[DŢ11] Cătălin Dima and Ferucio Laurenţiu Ţiplea. Model-checking ATL under Imperfect Information and Perfect Recall Semantics is Undecidable. Not yet published. Available at http://arxiv.org/abs/1102.4225 (version v1), 2011.

[EH85] E. Allen Emerson and Joseph Y. Halpern. Decision Procedures and Expressiveness in the Temporal Logic of Branching Time. *Journal of Computer and Systems Science*, 30(1):1–24, 1985.

[Eme90] E. Allen Emerson. Temporal and Modal Logic. In *Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics*, pages 995–1072. Elsevier Science Ltd, 1990.

[Gor01] Valentin Goranko. Coalition games and alternating temporal logics. In *Proceedings of the Eighth Conference on Theoretical Aspects of Rationality and Knowledge (TARK VIII)*, pages 259–272. Morgan Kaufmann, 2001.

[Gra78] Jim Gray. Notes on Data Base Operating Systems. In *Operating Systems, An Advanced Course*, pages 393–481, London, UK, 1978. Springer.

[GvD06] Valentin Goranko and Govert van Drimmelen. Complete axiomatization and decidability of Alternating-time temporal logic. *Theoretical Computer Science*, 353(1-3):93–117, 2006.

[JÅ07] Wojciech Jamroga and Thomas Ågotnes. Constructive Knowledge: What Agents Can Achieve under Imperfect Information. *Journal of Applied Non-Classical Logics*, 17(4):423–475, 2007.

[Jam04a] Wojciech Jamroga. Some Remarks on Alternating Temporal Epistemic Logic. In *Proceedings of Formal Approaches to Multi-Agent Systems (FAMAS 2003)*, pages 133–140, 2004.

[Jam04b] Wojciech Jamroga. *Using multiple models of reality. On agents who know how to play safer.* PhD thesis, University of Twente, 2004.

[Jam06] Wojciech Jamroga. On the Relationship between Playing Rationally and Knowing how to Play: A Logical Account. In *Proceedings of the 29th German Conference on Artificial Intelligence (KI'06),* Lecture Notes in Artificial Intelligence, volume 4314, pages 403–417. Springer, 2006.

[JD08] Wojciech Jamroga and Jürgen Dix. Model checking abilities of agents: A closer look. *Theory of Computing Systems*, 42(3):366–410, 2008.

[JvdH04] Wojciech Jamroga and Wiebe van der Hoek. Agents that Know How to Play. *Fundamenta Informaticae*, 63(2-3):185–219, 2004.

[Mye82] Roger B. Myerson. Cooperative Games with Incomplete Information. Discussion Papers 528, Northwestern University, Center for Mathematical Studies in Economics and Management Science, June 1982.

[Mye97] Roger B. Myerson. *Game theory: Analysis of Conflict*. Harvard University Press, 1. Harvard University Press paperback edition, 1997.

[Pap94] Christos H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[Pau02] Marc Pauly. A Modal Logic for Coalitional Power in Games. *Journal of Logic and Computation*, 12(1):149–166, 2002.

[Sch03] Philippe Schnoebelen. The Complexity of Temporal Logic Model Checking. In *Advances in Modal Logic*, volume 4, pages 393–436. King's College Publications, 2003.

[Sch04] Pierre-Yves Schobbens. Alternating-time logic with imperfect recall. *Electronic Notes in Theoretical Computer Science*, 85(2):82–93, 2004.

[vBBW05] Johan van Benthem, Patrick Blackburn, and Frank Wolter, editors. *Handbook of Modal Logic*. Elsevier Science Ltd, 2005.

[vdHJW05] Wiebe van der Hoek, Wojciech Jamroga, and Michael Wooldridge. A Logic for Strategic Reasoning. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '05)*, pages 157–164. ACM, 2005.

[vdHLW06] Wiebe van der Hoek, Alessio Lomuscio, and Michael Wooldridge. On the Complexity of Practical ATL Model Checking. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'06)*, pages 201–208. ACM, 2006.

[vdHW03a] Wiebe van der Hoek and Michael Wooldridge. Cooperation, Knowledge, and Time: Alternating-time Temporal Epistemic Logic and its Applications. *Studia Logica*, 75(1):125–157, 2003.

[vdHW03b] Wiebe van der Hoek and Michael Wooldridge. Model Checking Cooperation, Knowledge, and Time - A Case Study. In *Research in Economics*, volume 57, pages 235–265, 2003.