

# MODAL LOGIC FOR GAMES AND INFORMATION

Wiebe van der Hoek and Marc Pauly

---

1	Introduction . . . . .	1078
2	Game Theory . . . . .	1080
2.1	Games in Strategic Form . . . . .	1080
2.2	Games in Extensive Form . . . . .	1082
2.3	Cooperation in Games . . . . .	1087
3	Game Forms and Kripke Models . . . . .	1090
3.1	Players and Outcomes . . . . .	1091
3.2	Formalising Solution Concepts in Modal Languages . . . . .	1092
3.3	Games as Process Models . . . . .	1094
3.4	Other Issues . . . . .	1096
4	Epistemic Logic . . . . .	1097
5	Interpreted Systems . . . . .	1101
6	Games with Imperfect Information . . . . .	1107
6.1	Case study: Perfect Recall . . . . .	1108
6.2	Outlook . . . . .	1110
7	Dynamic Epistemic Logic . . . . .	1110
7.1	Public Announcements . . . . .	1111
7.2	General Updates . . . . .	1115
8	Epistemic Foundations of Solution Concepts . . . . .	1121
8.1	Epistemic Foundations for Strategic Games . . . . .	1122
8.2	Epistemic Foundations for Extensive Games . . . . .	1126
8.3	On the Representation of Games with Imperfect Information . . . . .	1129
9	Game Logic . . . . .	1130
9.1	Syntax and Semantics . . . . .	1130
9.2	Metatheory . . . . .	1132
9.3	Other Topics . . . . .	1134
10	Coalition Logic . . . . .	1135
10.1	Syntax and Semantics . . . . .	1135
10.2	Axiomatics . . . . .	1136
10.3	Complexity . . . . .	1137
11	Alternating-time Temporal Logic . . . . .	1138
11.1	Syntax and Semantics . . . . .	1138
11.2	An Example . . . . .	1140
11.3	Axiomatics and Complexity . . . . .	1141
11.4	Extensions: $\mu$ -calculus and imperfect information . . . . .	1142
12	Conclusion . . . . .	1143

---

## 1 INTRODUCTION

Game-theoretic ideas have a long history in logic. A game-theoretic interpretation of quantification goes back at least to C.S. Peirce, and game-theoretic versions for all essential logical notions (truth in a model, validity, model comparison) have been developed subsequently. The connections between game theory and *modal* logic, on the other hand, have been developed only more recently. At the time of writing, the area is still an active one, so active in fact, that one might even argue that it is too early to be included in a handbook of modal logic. In spite of this concern, we believe that the volume of research falling into the category of modal logic and game theory justifies a survey. Our attempt in this chapter is to put some structure on the various strands of research, to create an organisation which highlights what we consider to be the essential lines of research. As with all endeavours of this sort, one cannot include all the research considered valuable, and we are well aware that our choice of topics also reflects our own interests and expertise.

Game theory has developed a wealth of interesting ideas for describing interactions which may involve a conflict of interest. So far, the logic community has restricted its attention to relatively few of these, mainly studying 2-player extensive games of perfect information which are strictly competitive (even win/lose). In fact, even this restricted class of games has turned out to be extremely rich, as set theory and computer science can testify. Still, given this traditionally narrow focus of logic, it is encouraging to see that more recent work in logic has extended the game-theoretic toolbox considerably, introducing, e.g., cooperative game theory, imperfect information and games involving more than 2 players. While even this can only be a beginning, by now the game-theoretic ideas used in logic certainly go beyond the intuitively natural idea of a winning strategy, and hence we will start our chapter with a section explaining the necessary background in game theory.

Having packed our game-theoretic baggage, our tour starts in Section 3 with new sights of a familiar landscape, possibly the most natural way to link games to modal logic. Game trees can be viewed as Kripke models, where the possible moves are modeled by an accessibility relation and additional information about payoffs and turn taking are encoded by propositional atoms. Structural equivalence notions such as bisimulation then turn into game equivalence notions, and we can investigate extensions of the modal language which can capture game-theoretic solution concepts such as the subgame-perfect equilibrium.

Leaving the first three sections together with the concluding Section 12 aside, the rest of this chapter can be divided into three reading tracks, epistemic logics (Sections 4-8), game logic (Section 9), and coalition logics (Sections 10-11), which can be pursued independently. The epistemic logic track describes approaches based on (dynamic) epistemic logic for dealing with imperfect information in games. The section on game logic describes an extension of Propositional Dynamic Logic for reasoning about games, focusing on operations for combining games like programs. The coalition logic track, finally, discusses a range of logics developed for modeling coalitional power in games, possibly also adding temporal or epistemic operators.

In Section 4 we introduce a widely accepted logic for knowledge in the area of games, where the assumptions impose that players are fully introspective and their knowledge is veridical. The more interesting properties are to be found when studying knowledge of

groups of players though, with *common knowledge* being the main and most intriguing notion in this palette. Section 5 introduces *interpreted systems*, a dominant paradigm in computer science to deal with knowledge and time, where time corresponds with steps in a protocol, or, for our purposes, indeed a game.

Such epistemic notions as introduced in Section 4 and 5 play an important role in games of imperfect information, and, as some major results in early game theory indicate, even beyond that (see Section 8). For instance, a procedure that yields a Nash equilibrium in extensive games, called *backward induction*, finds its justification in the assumption about common knowledge about rationality of the players. We will see, however, that nowadays epistemologists put the need for the inherently infinite conjunctions that come along with common knowledge into perspective and a non-trivial analysis of games can be given without falling back on such strong assumptions.

Where the emphasis in Section 4 is on the knowledge that the players have *about* the game, in Sections 7 and 8 the emphasis will be on how knowledge evolves *during* a game. Hiding one's knowledge can be beneficial for a player within a game, but revealing his ignorance can also be disastrous, and may benefit other players. Moreover, in certain games (like *Cluedo* and many card games) the winning conditions are purely epistemic: the game ends in a win for that player who is the first who happens to know some crucial information.

The dynamic logic of games discussed in Section 9 takes Propositional Dynamic Logic as its starting point. By a change in the underlying semantics, programs become 2-player games which can be combined using the old program operations of sequential composition, test, etc. Besides these program operations, a new duality operator is added which interchanges the roles of the players. Using this new operator, nondeterministic choice splits into two versions depending on which player makes the choice. A typical formula  $[(a \cap b); (a \cup c)]p$ , for instance, expresses that player 2 has a strategy for achieving  $p$  in the game where first, player 1 chooses between  $a$  and  $b$ , and then player 2 chooses between  $a$  and  $c$  (for details, see Section 9).

Section 10 introduces Coalition Logic, a basic modal logic for reasoning about the ability of groups in different kinds of games. For a set of individuals  $C$ , the formula  $[C]p$  expresses that the members of  $C$  have a joint strategy for achieving  $p$  at the next stage of game. In Section 11, this language is extended to Alternating-time Temporal Logic (ATL) by adding operators for talking about the long-term future, where we can state, e.g., that a coalition can achieve  $p$  eventually. ATL is a game-theoretic generalisation of Computation Tree Logic (CTL), with applications in the formal verification of multi-agent systems. Further extensions of ATL to ATL\*, the alternating  $\mu$ -calculus and ATEL are presented. ATEL adds epistemic operators to ATL in order to express, e.g., that a coalition has a strategy for getting an agent to know something eventually.

While our focus in this chapter is on *modal logic for games*, there are also many *games for modal logic*. The reader interested in this reverse connection is referred to other chapters of this handbook (i.e., Chapter 12 and 17) for more details. The similarity between programs and games and the relevance of epistemic and temporal issues, on the other hand, suggest that modal logic may provide an interesting new perspective on games, and it is this perspective we would like to present in this chapter.

## 2 GAME THEORY

The purpose of this section is to introduce the basic game-theoretic notions needed for the logics discussed in later sections. Hence, this section will also give the reader an indication of the size and nature of the game-theoretic territory which has come under logical investigation.

In Sections 2.1 and 2.2, we discuss games in strategic and extensive forms, respectively. We cover some central solution concepts developed for these models, namely Nash equilibria and subgame-perfect equilibria. For a more detailed discussion of these notions, standard texts on game theory (e.g., [56, 11]) can be consulted. Section 2.3 focuses on a game-theoretic model of cooperation (effectivity functions) which has been investigated in social choice theory [52] and which will play a central role in the logics discussed in Sections 9, 10 and 11.

### 2.1 Games in Strategic Form

One of the most general models for situations of strategic interaction is that of a strategic game. Because of its generality, strategic games form the standard model in non-cooperative game theory. In a strategic game, the different players choose one of their available alternative actions/strategies, and taken together, these actions determine the outcome of the game. Note that we do not distinguish actions from strategies in strategic games; in extensive games, we will distinguish these two notions. Also, note that game *forms* can be conceived of as ‘uninterpreted games’: they only deal with the structure of the game, determining which moves are possible in which states, but they do not specify which states are ‘good’ or ‘bad’ for any player, i.e., they say nothing about winning, losing, a payoff or utility, when a particular state is reached.

**DEFINITION 1** (Strategic Game Form). A *strategic game form*  $F = (N, \{\Sigma_i | i \in N\}, o, S)$  consists of a nonempty finite set of agents  $N$ , a nonempty set of strategies or actions  $\Sigma_i$  for every player  $i \in N$ , a nonempty set of states  $S$  and an outcome function  $o : \prod_{i \in N} \Sigma_i \rightarrow S$  which associates to every tuple of strategies of the players (strategy profile) an outcome state in  $S$ .  $\dashv$

For notational convenience, let  $\sigma_C := (\sigma_i)_{i \in C}$  denote the strategy tuple for coalition  $C \subseteq N$  which consists of player  $i$  choosing strategy  $\sigma_i \in \Sigma_i$ . Then given two strategy tuples  $\sigma_C$  and  $\sigma_{\bar{C}}$  (where  $\bar{C} := N \setminus C$ ),  $o(\sigma_C, \sigma_{\bar{C}})$  denotes the outcome state associated with the strategy profile induced by  $\sigma_C$  and  $\sigma_{\bar{C}}$ . We shall also write  $-i$  for  $N \setminus \{i\}$ .

Figure 1 below provides an example of a strategic game form among three players in the usual matrix depiction. Unless noted otherwise, we will assume that player 1 chooses the row, player 2 the column, and the third player chooses between the left and the right table. In this example, let  $\sigma_1$  be the strategy where player 1 chooses  $B$ ,  $\sigma_2$  the strategy where player 2 chooses  $M$ , and let  $\sigma_3$  be the strategy of player 3 choosing the left table. Then we have  $o(\sigma_{\{1,2\}}, \sigma_{\{3\}}) = o((\sigma_1, \sigma_2, \sigma_3)) = s_1$ .

To make a strategic game form into a strategic game, we need to add preference relations or utility functions which express the players’ preferences over the game’s outcomes. In the first case, given a preference relation  $\succeq_i \subseteq S \times S$  for every player  $i \in N$  and a strategic game form  $F = (N, \{\Sigma_i | i \in N\}, o, S)$ , we call  $G = (F, (\succeq_i)_{i \in N})$  a *strategic game*. We interpret  $s \succeq_i t$  to mean that player  $i$  prefers outcome  $s$  at least as much as outcome  $t$ , and one usually assumes that  $\succeq_i$  is a linear order (although this assumption

	<i>L</i>	<i>M</i>	<i>R</i>		<i>L</i>	<i>M</i>	<i>R</i>
<i>T</i>	$s_1$	$s_2$	$s_1$	<i>T</i>	$s_3$	$s_2$	$s_1$
<i>B</i>	$s_2$	$s_1$	$s_3$	<i>B</i>	$s_2$	$s_3$	$s_3$

Figure 1. A strategic game form for three players.

shall not be essential for our development later). Similarly, given a strategic game form  $F$  and a utility function  $u_i : S \rightarrow \mathbb{R}$  for every player  $i \in N$ , we can construct a *strategic game*  $G = (F, (u_i)_{i \in N})$  where player  $i$  prefers outcome  $s$  at least as much as outcome  $t$  iff  $u_i(s) \geq u_i(t)$ . Provided that  $\succeq_i$  is indeed a linear order, the two definitions are interchangeable, and we shall freely switch between the two formats.

Figure 2 below shows three well-known 2-player games. The matrices list the players' utilities or payoffs, e.g., the top leftmost entry of the leftmost game,  $(-4, -4)$  denotes the pair  $(u_1(D, D), u_2(D, D))$ .

	<i>D</i>	<i>C</i>		<i>B</i>	<i>F</i>		<i>H</i>	<i>T</i>
<i>D</i>	$-4, -4$	$0, -8$		<i>B</i>	$2, 1$	$0, 0$	<i>H</i>	$-1, 1$
<i>C</i>	$-8, 0$	$-1, -1$		<i>F</i>	$0, 0$	$1, 2$	<i>T</i>	$1, -1$

Figure 2. Three strategic 2-player games: Prisoner's Dilemma (left), Battle of the Sexes (middle) and Matching Pennies (right)

In the Prisoner's Dilemma, two prisoners are interrogated by the police. If the prisoners cooperate (C) and remain silent, they can only be sentenced for a minor offence and will receive one year in prison each. If both defect and confess (D), each will receive 4 years in prison. Finally, if only one prisoner defects, he will go free in order to be used as a witness against his fellow prisoner who will receive 8 years in prison. In the Battle of the Sexes, a couple needs to decide whether to go see a ballet performance (B) or a football match (F) in the evening. Both of them mainly want to spend the evening together, but she prefers the football match and he prefers the ballet performance. Lastly, in the Matching Pennies example, two children each have a penny, and they decide simultaneously whether to show heads (H) or tails (T). One child wins (payoff 1 for the winner,  $-1$  for the loser) in case the sides match, the other child wins in case they differ. Matching Pennies is an example of a *zero-sum* or *strictly competitive* game: For every outcome state  $s$  we have that  $u_1(s) + u_2(s) = 0$ .

A strategic game allows us to model multi-agent interaction using strategies and preferences. Game theory has developed a number of solution concepts which specify a "predicted" set of outcomes for such a game (views differ as to how exactly such a solution has to be interpreted). The following notion is one of the cornerstones of modern game theory.

**DEFINITION 2 (Nash Equilibrium).** A strategy profile  $\sigma_N$  is a *Nash equilibrium* of a strategic game  $G = (N, \{\Sigma_i | i \in N\}, o, S, (\succeq_i)_{i \in N})$  iff  $\forall i \in N \forall \tau_i \in \Sigma_i : o(\sigma_i, \sigma_{-i}) \succeq_i o(\tau_i, \sigma_{-i})$   $\dashv$

Intuitively, a strategy profile  $(\sigma_1, \sigma_2)$  is a Nash equilibrium in a 2-player game in case  $\sigma_1$  is a best response to  $\sigma_2$  and vice versa; no player can improve his payoff by unilaterally

changing his strategy. In the three examples given in Figure 2, the reader may wish to verify that  $(D, D)$  is the only Nash equilibrium in the Prisoner's Dilemma, and that both  $(B, B)$  and  $(F, F)$  are Nash equilibria in the Battle of the Sexes. There is no Nash equilibrium in Matching Pennies.

While for a given player  $i$ , a Nash equilibrium only requires  $i$ 's action to be optimal given the other players' actions, a *dominant* strategy is optimal regardless of what the other players do. Formally, for two strategies  $x, y \in \Sigma_i$ ,  $x$  *strictly dominates*  $y$  iff  $\forall \sigma_{-i} : o(x, \sigma_{-i}) \succ_i o(y, \sigma_{-i})$ . We call a strategy strictly dominated iff it is strictly dominated by some other strategy. In the prisoner's dilemma, cooperation is strictly dominated by defection. No dominated strategies exist in the other games mentioned so far.

Our concern in this chapter is mainly with *pure* strategies which are non-probabilistic. In contrast, a *mixed strategy* allows a player to randomise over his set of strategies, playing each strategy with some probability  $p$  where  $0 \leq p \leq 1$ . In the Matching Pennies game, each player may decide to choose Heads with probability  $\frac{1}{2}$ . This strategy profile  $((\frac{1}{2}, \frac{1}{2}), (\frac{1}{2}, \frac{1}{2}))$  is a Nash equilibrium over mixed strategies, and Nash's celebrated result states that it is no coincidence that Matching Pennies has such an equilibrium.

**THEOREM 3 ([53]).** *Every strategic game has a Nash equilibrium over mixed strategies.*

The computational complexity of *finding* a mixed strategy Nash equilibrium (from Theorem 3 we are ensured of its existence) for a 2-player strategic game with finite (pure) strategy sets is in NP, but it is presently not known whether the problem is also NP-hard. According to [57], this is one of the most important concrete open questions on the boundary of P today.

## 2.2 Games in Extensive Form

Strategic games consider strategic interaction as involving only a single choice for every player. There may be situations, however, where we want to model the fine structure of strategic interaction which involves modelling the sequential structure of decision making. Extensive form games provide us with this level of added detail.

### Perfect Information

Given a finite or infinite sequence of actions  $h = (a_1, a_2, \dots)$ , let  $h|k = (a_1, a_2, \dots, a_k)$  denote the initial subsequence of length  $k$  of  $h$ .

**DEFINITION 4** (Extensive Game Form of Perfect Information). An *extensive game form of perfect information* is a triple  $F = (N, H, P)$ , where as before,  $N$  is the set of players.

$H$  is a set of sequences (finite or infinite) over a set  $A$  of actions which we shall call *histories* (or: *runs*, *plays*) of the game. We require that  $H$  satisfies three requirements: (1) the empty sequence  $() \in H$ . (2)  $H$  is closed under initial subsequences, i.e., if  $h \in H$  has length  $l$ , then for all  $k < l$  we have  $h|k \in H$ . If  $h \in H$  is infinite,  $h|k \in H$  for all  $k$ . (3) If all finite initial subsequences of an infinite sequence  $h$  are in  $H$ , then so is  $h$ : given an infinite sequence  $h$  such that for all  $k$  we have  $h|k \in H$ , then  $h \in H$ . Let  $Z \subseteq H$  be the set of *terminal histories*, i.e.  $h \in Z$  iff for all  $h' \in H$  and  $k$  such that  $h'|k = h$  we have  $h' = h$  (so, infinite histories are terminal).  $P : H \setminus Z \rightarrow N$  is the player function which assigns to every nonterminal history the player whose turn it is to move.  $\dashv$

As with strategic games, we turn an extensive game form into an extensive game

by adding preference relations. Formally, let  $\succeq_i \subseteq Z \times Z$  be a preference relation on the set of terminal histories  $Z$ . As before, we shall sometimes use utility functions  $u_i$  instead of preference relations. Given an extensive game form  $F = (N, H, P)$ , we call  $G = (F, (\succeq_i)_{i \in N}) = (N, H, P, (\succeq_i)_{i \in N})$  an *extensive game*. We call an extensive game  $G$  *finite* iff its set of histories  $H$  is finite.  $G$  has a *finite horizon* iff all histories in  $H$  have finite length.

Figure 3 shows the tree representation of an extensive game, where the branches are labelled by actions and the payoffs of the players are shown at the terminal nodes.

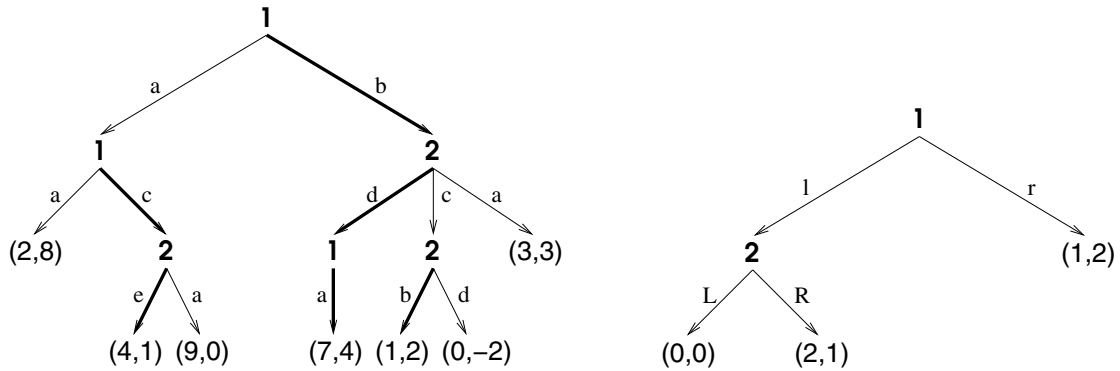


Figure 3. An extensive game for two player with payoffs (left) and an extensive game with an implausible Nash equilibrium (right)

Given a finite history  $h = (a_1, \dots, a_n)$  and an action  $x \in A$ , let  $(h, x) = (a_1, \dots, a_n, x)$ . Furthermore, let  $A(h) = \{x \in A \mid (h, x) \in H\}$  be the set of actions possible after  $h$ . Now we can define a strategy for player  $i$  as a function  $\sigma_i : P^{-1}[\{i\}] \rightarrow A$  such that  $\sigma_i(h) \in A(h)$  ( $P^{-1}$  denotes the pre-image of  $P$ ). As before, we let  $\Sigma_i$  denote the set of strategies of player  $i$ . Given a strategy profile  $\sigma = (\sigma_i)_{i \in N}$ , let  $o(\sigma) \in H$  be the history which results when the players use their respective strategies.

In the game on the left of Figure 3, the strategy for player 1 indicated by bold arrows is given by  $\sigma_1(( )) = b$ ,  $\sigma_1((a)) = c$  and  $\sigma_1((b, d)) = a$ . Player 2's strategy indicated in the game is given by  $\sigma_2((a, c)) = e$ ,  $\sigma_2((b)) = d$  and  $\sigma_2((b, c)) = b$ .

The notion of a Nash equilibrium can now be lifted easily from strategic games to extensive games. Given an extensive game  $G = (N, H, P, (\succeq_i)_{i \in N})$ , a strategy profile  $\sigma$  is a *Nash equilibrium* iff  $\forall i \in N \forall \tau_i \in \Sigma_i : o(\sigma_i, \sigma_{-i}) \succeq_i o(\tau_i, \sigma_{-i})$ . However, it turns out that for extensive games, Nash equilibria often lack plausibility, as shown by the game on the right in Figure 3, taken from [56].

The Nash equilibria of the game are  $(l, R)$  and  $(r, L)$ . The second equilibrium however does not seem reasonable. At the position where player 2 has to move, she will choose  $R$  since this will give her a higher payoff. Knowing this, player 1 should choose  $l$  at the beginning of the game, and so we would want to advocate only  $(l, R)$  as the solution of the game. The strategy profile  $(r, L)$  turns out to be a Nash equilibrium because of the threat that player 2 will choose  $L$  rather than  $R$ , but this threat is not credible since choosing  $L$  would hurt her own interest. To rule out such pathological equilibria, we need to strengthen our equilibrium notion. The problem with profile  $(r, L)$  is that it prescribes an unreasonable choice in a subgame of the original game, whereas we would want our

equilibrium strategies to be optimal in every subgame.

To obtain a more robust equilibrium notion, we introduce the notion of a subgame more formally. Given two sequences  $h$  and  $h'$ , with  $h$  finite, let  $(h, h')$  denote the concatenation of  $h$  and  $h'$ . Consider a history  $h$  in the extensive game  $G = (N, H, P, (\succeq_i)_{i \in N})$ . To isolate the subgame starting after  $h$ , we define  $G(h) = (N, H|_h, P|_h, (\succeq_i|_h)_{i \in N})$  where  $H|_h = \{h' | (h, h') \in H\}$ ,  $P|_h(h') = P(h, h')$  for each  $h' \in H|_h$  and finally  $h' \succeq_i|_h h''$  iff  $(h, h') \succeq_i (h, h'')$ . Similarly, strategies  $\sigma_i$  (and strategy profiles) can be restricted to subgames by setting  $\sigma_i|_h(h') = \sigma_i(h, h')$ .

**DEFINITION 5** (Subgame-Perfect Equilibrium). A strategy profile  $\sigma$  is a *subgame-perfect Nash equilibrium* of a game  $G$  iff for every history  $h \in H$ , the restriction  $\sigma|_h$  of  $\sigma$  is a Nash equilibrium of  $G(h)$ .  $\dashv$

In the game of Figure 3 (right), only  $(l, R)$  is a subgame-perfect equilibrium. Note that all subgame-perfect equilibria are also Nash equilibria. A game is called *generic* if no player is indifferent between any two terminal histories, i.e., for all  $i \in N$  and  $h, h' \in H$ , we have  $u_i(h) = u_i(h')$  iff  $h = h'$ .

**THEOREM 6** ([42]). *Every finite extensive game of perfect information has a subgame-perfect equilibrium. Moreover, in any generic game, this equilibrium is unique.*

While the formal proof of this theorem is somewhat technical, the general method used to establish the result is easy to explain and is known as *backward induction*. We build up the equilibrium profile by induction on the length of a game (i.e., the length of its longest history). If a game has length 0, it only consists of a terminal node and there are no strategic decisions to be made. Consider the payoff vector of the terminal node as the backward induction vector (short: bi-vector) of the game. Now if the game  $G$  has length  $n + 1$ , assume that player  $i$  has to move at the root of  $G$ . By induction hypothesis, all the proper subgames of  $G$  have a bi-vector and we have associated strategy profiles for them. To get an equilibrium profile for  $G$ , let  $i$  choose a successor with the highest bi-vector for  $i$ , and consider that payoff vector as the bi-vector of  $G$ . Starting at the terminal nodes, this backward induction method moves up through the game tree and inductively defines a strategy profile which turns out to be a subgame-perfect equilibrium and a bi-vector which is its associated payoff vector.

Note that the intuitive reasoning which we used to argue against  $(r, L)$  as a solution of the game in Figure 3 (right) was already an example of backward induction reasoning. In Figure 3, the backward induction profile in the game to the left has been indicated by boldface arrows. The payoff vector of the backward induction profile is  $(7, 4)$ .

As a corollary to Theorem 6, we obtain the following well-known result due to Zermelo which (in a slightly generalised version) can be used to show, e.g., that in the game of chess, either black or white must have a strategy which guarantees at least a draw. We say that a 2-player extensive game of perfect information is a *win-loss game* provided that it is strictly competitive and for all histories  $h$ , either  $u_1(h) = 1$  or  $u_2(h) = 1$ , i.e., win and loss are the only two possible outcomes. In such a game, a strategy is a *winning strategy* for player  $i$  provided it guarantees a history  $h$  such that  $u_i(h) = 1$ .

**THEOREM 7** ([107]). *Every finite 2-player win-loss game is determined, i.e., one of the players has a winning strategy.*

One can even show that the problem of determining whether such a game is a win for a specific player can be determined in polynomial time: in fact, it is a ‘canonical  $P$ -complete



problem' in the sense that it is a popular candidate to be related to other computational problems (like AND/OR Graph Solvability) in order to show *their* polynomial complexity. See [27, Appendix 11.1].

### *Almost Perfect Information*

Extensive games of perfect information impose a strict order on the moves which take place in a game. As a first generalisation, we may extend the definition of an extensive game to allow for simultaneous moves of the players. These *extensive games of almost-perfect information* (or *extensive games with simultaneous moves*) will be important for our discussion of Coalition Logic and ATL. In these games, players are completely informed about the past, but they may be unsure about the present, i.e., about the actions the other players are simultaneously taking.

Formally, an extensive game of almost-perfect information is a tuple  $G = (N, H, P, (\succeq_i)_{i \in N})$  just like an extensive game of perfect information, with the only difference that for every nonterminal history  $h \in H$ ,  $P(h)$  is a nonempty subset of  $N$ . Furthermore, for each  $i \in P(h)$ , we have a set  $A_i(h)$  of actions possible for player  $i$  at  $h$ , and we define the set of actions possible after  $h$  to be  $A(h) = \prod_{i \in P(h)} A_i(h)$ . Histories of the game are now sequences of vectors, consisting of the actions chosen simultaneously by the appropriate players. A strategy for player  $i$  is now a function  $\sigma_i$  such that  $\sigma_i(h) \in A_i(h)$ . The definitions of Nash equilibrium and subgame-perfect equilibrium can easily be adjusted to these extensive games with simultaneous moves.

### *Imperfect Information*

So far, we have assumed that the players always know where they are in the game tree. This amounts to assuming that the players are always informed about the actions which have been taken so far, both by the other players and by themselves; in short, we have considered perfect information games where a player has no private information (e.g., cards which only she knows), nor does she ever forget which moves she has made earlier (this is the essence of the game "Memory"). The game model we shall introduce now is an extension of the extensive game model to cover situations of imperfect information.

**DEFINITION 8** (Extensive Game of Imperfect Information). An *extensive game of imperfect information* is a tuple  $G = (N, H, P, (\mathcal{I}_i)_{i \in N}, (\succeq_i)_{i \in N})$ . The only new component  $\mathcal{I}_i$  is a partition of the set of histories where  $i$  has to move, i.e. of  $P^{-1}[\{i\}]$ , with the property that for all  $h, h' \in I \in \mathcal{I}_i$ ,  $A(h) = A(h')$ .  $\dashv$

The elements of  $\mathcal{I}_i$  are called *information sets*. If player  $i$  has to make a decision in a game at history  $h \in I \in \mathcal{I}_i$ , she does not know which of the histories in  $I$  is the real history, i.e. she considers all histories in  $I$  as possible alternatives to  $h$ . In order for this interpretation to make sense, we have to assume that the histories in  $I$  cannot be distinguished by what actions are possible in the various histories, a requirement which we enforced by demanding that for all  $h, h' \in I$ ,  $A(h) = A(h')$ . Observe that if all information sets are singletons, we have in fact a game of perfect information.

Since a player cannot distinguish between two histories which are in the same information set, her strategies have to be uniform within every information set. Hence, we define a strategy for an extensive game of imperfect information to be a function  $\sigma_i : \mathcal{I}_i \rightarrow A$  such that  $\sigma_i(I) \in A(I)$ . In words, a strategy picks an action for every information set;

since histories within one information set allow for the same actions to be taken, the action prescribed by the strategy can always be executed, no matter where the player really is in the game tree.

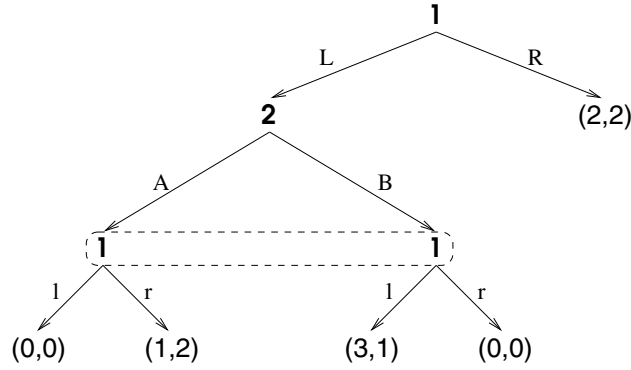


Figure 4. Example of an extensive game of imperfect information

Figure 4 contains an example of an imperfect information game. The information sets which are not singletons have been indicated by drawing a dashed box around the histories which are in the same information set. So in this game, player 1 makes the first move, and assuming she chooses to play  $L$ , player 2 moves afterwards. Player 1 however obtains no information about the choice made by player 2, maybe she was not present when the choice was made, maybe she forgot, etc. Her strategy therefore would have to specify either  $l$  or  $r$  for both cases, since she is unable to distinguish them. Two more examples of imperfect information games are given in Figure 5. The left game exhibits

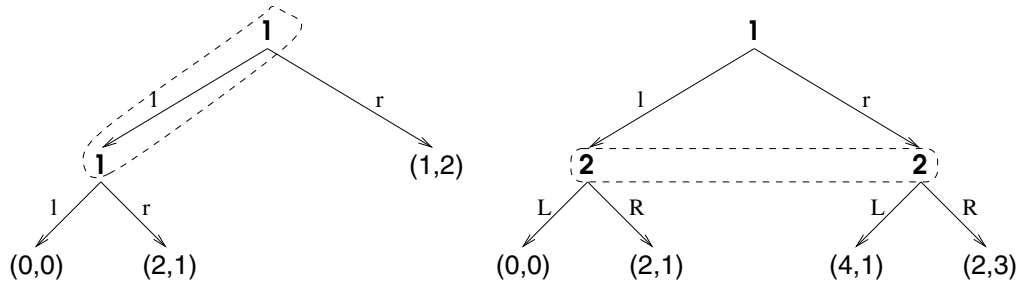


Figure 5. A game with imperfect recall (left) and one with simultaneous moves (right)

imperfect recall: After doing  $l$ , player 1 does not know whether she has done  $l$  already or not. Since this notion will play a role later, we will define it more formally here. Given game  $G = (N, H, P, (\mathcal{I}_i)_{i \in N}, (\succeq_i)_{i \in N})$  and history  $h \in H$ , let  $X_i(h)$  record the experience of player  $i$  along history  $h$ , i.e., the sequence of information sets the player encounters in  $h$  and the actions he takes at that information set. So with the game of Figure 4, we have  $X_1(LAr) = (\emptyset, L, \{LA, LB\}, r)$ . Then we say that  $G$  has *perfect recall* iff for every player  $i$  we have  $X_i(h) = X_i(h')$  whenever there is some  $I \in \mathcal{I}_i$  such that  $h, h' \in I$ . So while Figure 4 presents a game with perfect recall, the left game of Figure 5 does not, since  $X_1(\emptyset) = (\emptyset)$  but  $X_1(l) = (\emptyset, l)$ .

The right game in Figure 5 shows that the imperfect information game model can also be used to model games with simultaneous moves. After player 1 moves, player 2 has to move without any information about the choice made by the first player. Thus, since player 1 also does not know about the decision that player 2 will make later, we can interpret this game as one where the two players simultaneously choose an action.

The concepts of Nash equilibrium and mixed strategy Nash equilibrium can easily be extended to imperfect information games. The story is more complicated for the subgame-perfect equilibrium. The game in Figure 6 demonstrates the problem that one can run into when wanting to apply backward induction to a game of imperfect information.

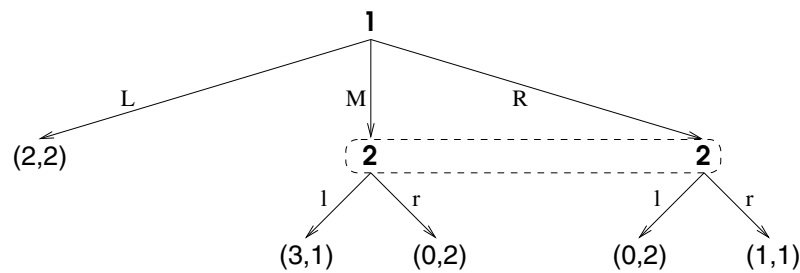


Figure 6. An imperfect information game where backward induction runs into problems

Player 2's only information set contains two subgames. In the left subgame,  $r$  is the strategy prescribed by backward induction, whereas in the right subgame,  $l$  is optimal. Since both subgames lie in the same information set, the backward induction strategy has to be uniform for both subgames. In order to label one of the two strategies as optimal, player 2 would have to know where she is in the game, but this is exactly what she does not know.

To deal with this problem, one can introduce a belief system which specifies at each information set the probability with which the player believes that a history has happened. The strategy choice can then make use of these probabilities. This leads to the notion of a *sequential equilibrium* which we shall not define formally here. Note simply that for the game in Figure 6, if player 2 believes that it is more likely that player 1 will play  $M$  rather than  $R$ , strategy  $r$  should be preferable to player 2. We refer the reader to [56] for the details.

### 2.3 Cooperation in Games

So far we have assumed that agents determine individually what strategy they want to follow. We made no attempt to account for the possibility that agents might cooperate in bringing about a desirable state of affairs. Effectivity functions, the model we discuss in this subsection, aim at capturing explicitly the powers which agents can obtain by forming coalitions.

Effectivity functions model the power distribution among individuals and groups of individuals. In social choice theory, they have been used in particular to model voting procedures. The exposition we will give here focuses on providing the necessary background to understand the link between effectivity functions and the neighbourhood

models used in non-normal modal logics like the ones we will discuss in Sections 9, 10 and 11.

Effectivity functions have been studied extensively in game theory and social choice theory [52, 1, 66]. The following exposition is based on [63, 62].

**DEFINITION 9 (Effectivity Function).** Given the finite nonempty set of players  $N$  and a nonempty set of alternatives or states  $S$ , an *effectivity function* is any function  $E : \mathcal{P}(N) \rightarrow \mathcal{P}(\mathcal{P}(S))$  which satisfies the following two conditions: (1)  $\forall C \subseteq N : \emptyset \notin E(C)$ , and (2)  $\forall C \subseteq N : S \in E(C)$ .  $\dashv$

The function  $E$  associates to every group of players the sets of outcomes for which the group is effective; coalition  $C$  is effective for  $X$  if it can bring about an alternative in  $X$ , even though it may have no control over which alternative of  $X$  is realised. The literature differs somewhat in the conditions placed on the function  $E$ . The conditions chosen here aim at Theorem 12 and the logics discussed in Sections 10 and 11. Informally, condition (1) of Definition 9 states that no group  $C$  can ensure that *nothing* is brought about, while condition (2) expresses that every group  $C$  can at least bring about *something* by ‘choosing’ the complete set of alternatives  $S$ , putting no constraints on what the players outside  $C$  can achieve.

**EXAMPLE 10.** Consider the following example from [24] about Angelina, Edwin and the judge: If Angelina does not want to remain single, she can decide either to marry Edwin or the judge. Edwin and the judge each can similarly decide whether they want to stay single or marry Angelina. If we assume that the three individuals live in a society where nobody can be forced to marry against his/her will, this situation can be modelled using effectivity functions as follows: The set of players is  $N = \{a, e, j\}$  and the set of alternatives is  $S = \{s_s, s_e, s_j\}$ , where  $s_s$  denotes the situation where Angelina remains single,  $s_e$  where she marries Edwin, and  $s_j$  where she marries the judge. Angelina (a) has the right to remain single, so  $\{s_s\} \in E(\{a\})$ , whereas Edwin can only guarantee that he does not marry Angelina; whether she marries the judge or remains single is not up to him. Consequently, we have  $\{s_s, s_j\} \in E(\{e\})$  and there is no proper subset  $X$  of  $\{s_s, s_j\}$  such that  $X \in E(\{e\})$ . Analogously for the judge, we have  $\{s_s, s_e\} \in E(\{j\})$ . Angelina and Edwin together can achieve any situation except the one where Angelina marries the judge (since this alternative would require the judge’s consent), and hence  $\{s_s\}, \{s_e\} \in E(\{a, e\})$ . Again, the situation is similar for the judge:  $\{s_s\}, \{s_j\} \in E(\{a, j\})$ .  $\dashv$

In most situations, coalitional effectivity functions will satisfy some additional properties. Among the central properties are the following:

**MONOTONICITY:** Since a superset of states places fewer constraints on a coalition’s ability, we can usually assume that effectivity functions are *monotonic*: For every coalition  $C \subseteq N$ , if  $X \subseteq X' \subseteq S$ ,  $X \in E(C)$  implies  $X' \in E(C)$ .

**MAXIMALITY:** An effectivity function  $E$  is *C-maximal* if for all  $X$ , if  $\overline{X} \notin E(\overline{C})$  then  $X \in E(C)$ .  $E$  is *maximal* iff for all coalitions  $C$  it is *C-maximal*. Instantiating this condition for 2 players over  $S = \{win_1, win_2\}$ ,  $\{1\}$ -maximality expresses that the game is determined: if one player does not have a winning strategy, then the other player does.

**SUPERADDITIVITY:** The most interesting principle governs the formation of coalitions. It states that coalitions can combine their strategies to (possibly) achieve more:  $E$  is *superadditive* if for all  $X_1, X_2, C_1, C_2$  such that  $C_1 \cap C_2 = \emptyset$ ,  $X_1 \in E(C_1)$  and  $X_2 \in E(C_2)$  imply that  $X_1 \cap X_2 \in E(C_1 \cup C_2)$ .

Given utility functions  $(u_i)_{i \in N}$  for the players, effectivity functions also allow us to define solution concepts. Given an effectivity function  $E : \mathcal{P}(N) \rightarrow \mathcal{P}(\mathcal{P}(S))$ , call an alternative  $s \in S$  *dominated* if there is a set  $X \subseteq S$  and a coalition  $C$  such that  $X \in E(C)$  and for all  $i \in C$  and  $x \in X$  we have  $u_i(x) > u_i(s)$ .

**DEFINITION 11 (Core).** Given an effectivity function  $E : \mathcal{P}(N) \rightarrow \mathcal{P}(\mathcal{P}(S))$  and utility functions  $(u_i)_{i \in N}$ , the *core* of  $(E, (u_i)_{i \in N})$  is the set of undominated alternatives.  $\dashv$

In connection with the core, it is usually also assumed that for all sets  $X \neq \emptyset$ , we have  $X \in E(N)$ , so in particular, any state can be achieved by the grand coalition of all players.

An effectivity function  $E$  is *stable* if for any set of utility functions  $(u_i)_{i \in N}$ , the core of  $(E, (u_i)_{i \in N})$  is nonempty. Given  $E$  and  $(u_i)_{i \in N}$ , one can determine in polynomial time whether the core of  $(E, (u_i)_{i \in N})$  is nonempty. Determining whether  $E$  is stable, however, is an NP-complete problem [50].

### From Strategic Games to Effectivity Functions

Effectivity functions can be derived from a strategic game form in a number of different ways. Given a strategic game form  $G$ , a coalition  $C \subseteq N$  will be  $\alpha$ -*effective* for a set  $X \subseteq S$  iff the coalition has a joint strategy which will result in an outcome in  $X$  no matter what strategies the other players choose. Formally, for a strategic game form  $G = (N, (\Sigma_i)_{i \in N}, o, S)$ , its  $\alpha$ -*effectivity function*  $E_G^\alpha : \mathcal{P}(N) \rightarrow \mathcal{P}(\mathcal{P}(S))$  is defined as follows:

$$X \in E_G^\alpha(C) \text{ iff } \exists \sigma_C \forall \sigma_{\overline{C}} \ o(\sigma_C, \sigma_{\overline{C}}) \in X.$$

We say that an effectivity function  $E : \mathcal{P}(N) \rightarrow \mathcal{P}(\mathcal{P}(S))$   $\alpha$ -*corresponds* to a strategic game  $G$  iff  $E = E_G^\alpha$ .

Analogously, a coalition  $C \subseteq N$  will be  $\beta$ -*effective* for a set  $X \subseteq S$  iff for every joint strategy of the other players, the coalition has a joint strategy which will result in an outcome in  $X$ . Hence, in contrast to  $\alpha$ -effectivity, the coalition's strategy may depend on the strategy of the other players. Formally, the  $\beta$ -*effectivity function*  $E_G^\beta : \mathcal{P}(N) \rightarrow \mathcal{P}(\mathcal{P}(S))$  of a game form  $G$  is defined as follows

$$X \in E_G^\beta(C) \text{ iff } \forall \sigma_{\overline{C}} \exists \sigma_C \ o(\sigma_C, \sigma_{\overline{C}}) \in X.$$

It is easy to see that  $E_G^\alpha \subseteq E_G^\beta$ , i.e.  $\alpha$ -effectivity implies  $\beta$ -effectivity, but the converse does not hold, as the example in Figure 7 illustrates. In that game  $G$ , player 1 chooses the row, player 2 the column, and the third player chooses between the left and the right table. For every joint strategy of players 1 and 3, player 2 has a strategy which yields outcome  $s_2$ . Note, however, that this strategy depends on the strategies chosen by players 1 and 3, i.e., player 2 has no strategy which will guarantee outcome  $s_2$  independent of the strategies of players 1 and 3. In terms of  $\alpha$ - and  $\beta$ -effectivity, we have  $\{s_2\} \in E_G^\beta(\{2\})$ , but  $\{s_2\} \notin E_G^\alpha(\{2\})$ . The coalition consisting of players 1 and 2 on the other hand does

	$l$	$m$	$r$		$l$	$m$	$r$
$l$	$s_1$	$s_2$	$s_1$	$l$	$s_3$	$s_1$	$s_2$
$r$	$s_2$	$s_1$	$s_3$	$r$	$s_2$	$s_3$	$s_3$

Figure 7. A strategic game where  $\alpha$ - and  $\beta$ -effectivity differ

have a joint strategy  $(r, l)$  which guarantees  $s_2$  independent of player 3's strategy, i.e.  $\{s_2\} \in E_G^\alpha(\{1, 2\})$ .

While this discussion shows that every strategic game form can be linked to an effectivity function via  $\alpha$ -correspondence, not every effectivity function will be the  $\alpha$ -effectivity function of a strategic game form. The properties required to obtain a precise characterisation result are the following.

**THEOREM 12 ([63]).** *An effectivity function  $\mathcal{P}(N) \rightarrow \mathcal{P}(\mathcal{P}(S))$   $\alpha$ -corresponds to a strategic game form if and only if it is monotonic,  $N$ -maximal and superadditive.*

### 3 GAME FORMS AND KRIPKE MODELS

Figure 3 invites the modal logician to apply his tools and analysis as provided in especially Chapter 1 of this handbook in a natural way: games in extensive form are just models over time or processes in disguise [85]. In other words, if one is interested in the moves and their outcomes, extensive games can be conceived of as Kripke models for dynamic logic: see also Section 9. We take this perspective in this section, swiftly moving the focus of our analysis from the area of games to that of logic. To start with, we abstract from the specific actions, and reason about what specific agents can achieve. We do this in an example driven, and semantic way, in Section 3.1. When combining the players' powers with their preferences, modal logic can help to shed light on solution concepts, an exercise we will undertake in Section 3.2. In these two subsections, it is not the extensive form of the game that matters for the modal analysis so much, but more which end-nodes can be reached. We make some remarks concerning equivalence and expressivity taking the full structure of the extensive game into account, in Section 3.3. Yet another point of view takes the strategies, or, rather, the paths in an extensive game as first class citizens: see Section 11.

#### *Games in Strategic Form*

However, before exploiting the modal structure of games in extensive form, let us follow the structure of Section 2.1 and point out that the applicability of modal logic also extends to games in strategic form. The contribution [13] introduces a dynamic relation  $R_i$  for every player  $i$ , with intended meaning that  $sR_it$  holds iff from state  $s$ , player  $i$  can unilaterally bring about state  $t$ . States are states of the game, and  $\sigma$  assigns to every state the strategy profile that is being played in any state: e.g.,  $\sigma_i(s) = s_i$  denotes that  $i$  plays strategy  $s_i$  in  $s$ . Regarding the atomic propositions, [13] assumes to have atoms  $q \leq p$  for any natural numbers  $p$  and  $q$  (with straightforward interpretation), and, for every player  $i$  and number  $p$ , an atom  $u_i = p$  denoting  $i$ 's utility (or pay-off) at any state: such an atom is true at  $s$  iff  $u_i(\sigma(s)) = p$ , where  $u_i$  assigns a utility  $p \in \mathbb{N}$  to

every strategy profile  $\sigma$ . We assume to only need finitely many values  $Val = \{p, q, \dots\}$ . Finally, the atom  $Nash$  is true at all the profiles at which this equilibrium is played.

For instance, using a suggestive name for states, in our representation of the Prisoner's Dilemma (see Figure 2), in state  $(D, D)$  it is true that  $u_1 = -4 \wedge u_2 = -4$  and, everywhere in the game, we have

$$\Box_2 \Diamond_1 \left( \bigvee_{q \in Val} (u_1 = q) \wedge (q \geq -4) \right) \wedge \Diamond_2 \Box_1 \left( \bigwedge_{p \in Val} (u_1 = p) \rightarrow (p \leq -4) \right) \quad (1)$$

Equation (1) expresses that, no matter what player 2 does, player one can guarantee himself a payoff of -4, but at the same time, he can do not better than that: 2 can play a move such that the best that 1 can achieve is -4. Note that we *do* have, in the game of Figure 2, that  $\Diamond_1 \Diamond_2 ((u_1 > -4) \wedge (u_2 > -4))$  (if the players were to cooperate, they could achieve more than -4 each). Let  $p_1, \dots, p_n$  be a set of  $n$  payoff values. The following is valid in every strategic game  $G$ :

$$\bigwedge_{i \in N} \left( (u_i = p_i) \wedge \Box_i \bigwedge_{q \in Val} (u_i = q \rightarrow (q \leq p_i)) \right) \leftrightarrow \left( \bigwedge_{i \in N} (u_i = p_i) \wedge Nash \right) \quad (2)$$

This formula is equivalent to saying that, given that every player  $i$ 's outcome is  $p_i$ , the strategy played is a Nash-outcome if and only if no player  $i$  can unilaterally deviate and achieve something ( $q$ ) that is better than  $p_i$ .

### 3.1 Players and Outcomes

Given an extensive game form  $F = (N, H, P)$ , we straightforwardly associate with it a *game frame*  $\mathcal{F}_F^N = \langle W, R_{i \in N} \rangle$  with the obvious addition that  $H = W$  and  $R_i$  st iff  $P(s) = i$  &  $t = (s, a)$  for some action  $a$ , i.e., if player  $i$  is to move in  $s$  and he can choose a move that leads to  $t$ . Such a frame is *generated* from a root and, moreover, *turn-based*: if  $R_i$  st then for no other  $j$  and for no  $u$  also  $R_j su$ . Basic propositions like **turn<sub>i</sub>** (player  $i$  is to move) and **end** (we are in a leaf) can easily be defined, as  $\Diamond_i \top$  and  $\bigwedge_{i \in N} \Box_i \perp$ , respectively. We can also start from an extensive game  $G$  and then obtain  $\mathcal{F}_G^N$  by augmenting the frame with a preference relation. Moreover, we can assume to have atoms  $u_i = p_i$  or **win<sub>i</sub>** in the language, and interpret them in an appropriate manner in a model  $\mathcal{M}_F^N$  or  $\mathcal{M}_G^N$ . For instance, in the game on the right of Figure 3, we have in the root that  $\Diamond_1 (\Box_2 (u_2 \leq 2) \wedge \Diamond_2 \top) \wedge \Diamond_1 \Box_2 \perp$ : player 1 can enforce a state such that player 2 can move but is unable to obtain more than 2 units, but player 1 can also move to a state in which player 2 cannot make any move anymore.

The paper [13] argues that modal logic not only is a useful tool to *describe* the rational behaviour of players (see Section 3.2), but also when it comes to *prescribing* the players how to act. To do so, [13] adds a relation  $R_*$  to a game frame  $\mathcal{F}^N$  representing (paths according to) a *recommendation* that the players are given, i.e.,  $R_* \subseteq R^N$ , where  $R^N$  is the transitive closure of  $N^T = (\bigcup_{i \in N} R_i)$  with the following properties: (1)  $R_*$  is transitive, (if it is recommended to reach  $w_2$  from  $w_1$  and  $w_3$  from  $w_2$ , then it is recommended to reach  $w_3$ , in  $w_1$ ); (2) if for some  $w^T$ ,  $N^T w w^T$  (i.e.,  $w$  is a decision node  $\in T = H \setminus Z$ ), then also  $R_* w w_*$  for some  $w_*$  (if a player is to move at  $w$ , then a recommendation must be made); and, finally (3) if  $R_* w_1 w_3$  and at the same time

$R^N w_1 w_2$  and  $R^N w_2 w_3$ , then we have both  $R_* w_1 w_2$  and  $R_* w_2 w_3$  (if it is recommended in  $w_1$  to reach  $w_3$ , then any path to do that is a recommended path).

Moreover, [13] allows for atomic propositions ( $q \leq p$ ) and  $u_i = p_i$ , the latter only being true in a state  $s$  iff  $s$  is a leaf, with  $u_i(s) = p_i$ . Now consider the following scheme:

$$\Diamond_*(u_i = p_i) \rightarrow \Box_i(((u_i = q_i) \vee \Diamond_*(u_i = q_i)) \rightarrow q_i \leq p_i) \quad (3)$$

[13] refers to (3) as *internal consistency* of a recommendation, “in the sense that no player can increase his payoff by deviating from the recommendation, *using the recommendation itself to predict his future payoff after the deviation*” ([13, page 17]).

Of course, it is up to the game theoretician to come up with the ‘right recommendation’, but an obvious choice would be that they play a Nash equilibrium. In a generic game (cf Theorem 13), the backward induction algorithm determines for every decision node a unique immediate successor: let us call this the backward induction relation  $BI$ . We say that a recommendation relation  $R_*$  is the backward induction recommendation if it is the transitive closure of  $BI$ . The next Proposition tells us that scheme (3) can be understood as characterising backward induction.

**THEOREM 13.** [13, Proposition 4.8] *Let  $G$  be a generic perfect information game and  $\mathcal{F}_G^N$  its associated Kripke model, with a recommendation relation  $R_*$ . Then the following are equivalent:*

- (1)  $R_*$  is the backward induction recommendation
- (2) scheme (3) is valid in  $\mathcal{F}_G^N$

### 3.2 Formalising Solution Concepts in Modal Languages

The aim of [35] is also to formalise solution concepts in a modal logic. We again start with frames  $\mathcal{F}_G$  based on a game  $G$  with preference relations  $\succeq_{i \in N}$ . It assumes that every  $\succeq_i$  is reflexive, transitive and connected (i.e., for all  $u, v$ ,  $u \succeq_i v$  or  $v \succeq_i u$ ). Every  $\succeq_i$  gives rise to an operator  $[i]$ , with intuitive reading of  $[i]\varphi$ : “ $\varphi$  holds in all states at least as preferable to the present one”.

The other first class citizens in [35] are *strategy profiles* in the game  $G$ . For any such profile  $\sigma$ , let  $R_\sigma st$  iff following  $\sigma$  in  $s$  would eventually lead to the end state  $t$ . Thus,  $[\sigma]\varphi$  reads “if from here all players adhere to  $\sigma$ , the play will eventually end in a state in which  $\varphi$  holds”. Finally, for every player  $i$  and strategy profile  $\sigma$ , recall that  $(\tau_i, \sigma_{-i})$  is the profile where all players stick to  $\sigma$ , except for  $i$  who deviates and plays  $\tau_i$ . We use this to define a third accessibility relation:  $R_{(i, \sigma)} st$  iff for some  $\tau_i$ ,  $R_{(\tau_i, \sigma_{-i})} st$ . Hence, the meaning of  $[i, \sigma]\varphi$  in  $s$  becomes that “ $\varphi$  holds in all the states that will be reached if all the players except possibly  $i$  play the strategy  $\sigma$ ”. Given an extensive game  $G = (N, H, P, (\succeq_i)_{i \in N})$  we now define  $\mathcal{F}_G$  as  $\langle H, (\succeq_i), (R_\sigma), R_{(i, \sigma)} \rangle$ , where  $i \in N$  ranges over the players, and  $\sigma$  over the strategy profiles. Note that the binary relations  $R_\sigma$ , and  $R_{(i, \sigma)}$  on such a frame have the leaves as their co-domain (cf. Figure 8). By decorating such a frame with a valuation  $\pi : H \rightarrow 2^A$  for some set of atoms  $A$ , we obtain a *game model*  $M_G$  for  $G$ .

Now, before stating a result about these relations, we first recall a result from *correspondence theory* (see Chapter 5).



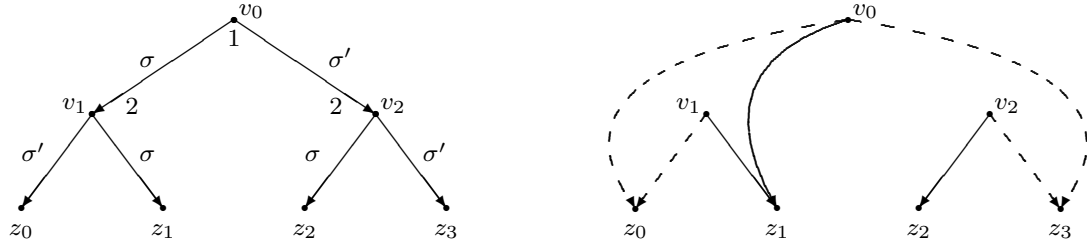


Figure 8. Transformation of an extensive game (left) to a game-frame (right) with respect to two strategy profiles  $\sigma$  and  $\sigma'$  and their corresponding accessibility relations  $R_\sigma$  and  $R_{(1, \sigma')}$  (dashed). Reflexive arrows at the leaves are omitted.

**THEOREM 14.** *Suppose we have three accessibility relations  $R_k, R_l$  and  $R_m$  with corresponding modalities. Then the scheme  $\langle k \rangle [l] \varphi \rightarrow [m] \varphi$  characterises frames satisfying  $(k, l, m)$ -Euclidicity, i.e., frames in which  $\forall s, t, u ((R_k s t \ \& \ R_m s u) \Rightarrow R_l t u)$ .*

Remember that a Nash-equilibrium in an extensive game is a strategic profile  $\sigma$  such that for all players  $i$  and profiles  $\tau$ , we have  $o(\sigma_i, \sigma_{-i}) \succeq_i o(\tau_i, \sigma_{-i})$ , i.e. no player can improve his situation by unilaterally deviating from  $\sigma$ . The profile  $\sigma$  is said to be a *best response* for player  $i$  iff for all profiles  $\tau$ , we have  $o(\sigma_i, \sigma_{-i}) \succeq_i o(\tau_i, \sigma_{-i})$ . This is an individual pendant of Nash-equilibrium: clearly, a strategy profile  $\sigma$  is a Nash-equilibrium iff it is a best response for all players.

**THEOREM 15** ([35], Theorem 3.1). *Let  $\mathcal{F}_G$  be obtained from the extensive game  $G$  as indicated above, and let  $v_0$  be its root-node. Let ‘s.p.’ stand for ‘sub-game perfect’.*

- (i)  $\sigma$  is a best response for  $i$  in  $G$  iff  $\mathcal{F}_G, v_0 \models \langle i, \sigma \rangle [i] \varphi \rightarrow [\sigma] \varphi$
- (ii)  $\sigma$  is an s.p. best response for  $i$  in  $G$  iff  $\mathcal{F}_G \models \langle i, \sigma \rangle [i] \varphi \rightarrow [\sigma] \varphi$
- (iii)  $\sigma$  is a Nash equilibrium in  $G$  iff  $\mathcal{F}_G, v_0 \models \bigwedge_{i \in N} (\langle i, \sigma \rangle [i] \varphi \rightarrow [\sigma] \varphi)$
- (iv)  $\sigma$  is an s.p. (Nash) equilibrium in  $G$  iff  $\mathcal{F}_G \models \bigwedge_{i \in N} (\langle i, \sigma \rangle [i] \varphi \rightarrow [\sigma] \varphi)$

Note that the first item, together with Theorem 14 says that  $\sigma$  is a best strategy for  $i$  in  $v_0$  iff when  $\sigma$  leads us to a leaf  $z$ , and any deviation by  $i$  from  $\sigma$  leads to a leaf  $z'$ , then in  $z'$  player  $i$  is not better off. Also note that the sub-game perfect notions are the global variants of the notions that hold in the root.

$Taut$	any classical tautology	$D!_\sigma$	$[\sigma] \varphi \leftrightarrow \langle \sigma \rangle \varphi$
$K$	$[\beta](\varphi \rightarrow \psi) \rightarrow ([\beta] \varphi \rightarrow [\beta] \psi)$	$F1$	$[\sigma, i] \varphi \rightarrow [\sigma] \varphi$
$T_i$	$[i] \varphi \rightarrow \varphi$	$F2$	$[\sigma, i] (\langle \sigma', i' \rangle \varphi \leftrightarrow \varphi)$
$4_i$	$[i] \varphi \rightarrow [i][i] \varphi$	$F3$	$[\beta][\beta'] ([i] \varphi \rightarrow \psi) \vee [\beta'''] [\beta'''] ([i] \psi \rightarrow \varphi)$
$MP$	$\vdash \varphi \rightarrow \psi, \vdash \varphi \Rightarrow \vdash \psi$	$Nec$	$\vdash \varphi \Rightarrow \vdash [\beta] \varphi$

Table 1. Axioms for Extensive Game Logic. The variables  $\beta, \beta' \dots$  range over the modalities  $[i], [\sigma]$  and  $[i, \sigma]$

The logic that comes with the semantics described here is dubbed *Extensive Game*

*Logic* in [35] (see Table 1). The axioms *Taut* and *K* and the rules *MP* and *Nec* denote that we have a normal modal logic;  $T_i$  and  $4_i$  reflect that the preference relation  $\succeq_i$  is reflexive and transitive. Axiom  $D!_\sigma$  says that  $R_\sigma$  is functional: any strategy profile  $\sigma$  prescribes a unique outcome. *F1* expresses that if any deviation by  $i$  from  $\sigma$  leads to a result  $\varphi$ , then this is in particular the case if  $i$  sticks to  $\sigma_i$ , i.e., does not deviate. Property *F2* reflects that any strategy  $(\tau_i, \sigma_{-i})$  takes us to a leaf. And from any leaf, any further moves are void, i.e.,  $[\sigma', i']\varphi \leftrightarrow \varphi$  holds in it. Finally, *F3* denotes a kind of *connectivity*. Correspondence theory (cf. Chapter 5) tells us that the modal scheme  $\Box_1(\Box_3\varphi \rightarrow \psi) \vee \Box_2(\Box_3\psi \rightarrow \varphi)$  corresponds to the property that all  $R_1$  and  $R_2$ -successors are  $R_3$ -connected:  $\forall w, u, v : ((R_1wu \& R_2wv) \Rightarrow (R_3uv \text{ or } R_3vu))$ . Hence *F3* expresses that all  $R_{\beta'} \circ R_\beta$  and  $R_{\beta''} \circ R_{\beta'}$ -successors are  $\succeq_i$ -connected. By connectivity of  $\succeq_i$ , this axiom is sound; the fact that we are in the realm of game frames makes *F3* also ensure this connectivity.

**THEOREM 16.** ([35, Theorem 4.1]) *Extensive Game Logic as presented in Table 1 is strongly complete with respect to the semantics based on game models as defined on page 1092.*

### 3.3 Games as Process Models

Although [35] takes games in extensive form as its starting point, a little reflection on Figure 8 should convince the reader that the language only allows to reason about the *outcomes* of the game, not about *intermediate states*. Modal logic provides a wide range of languages, allowing to discriminate graphs, and hence games, on many levels of abstraction. This section discusses ideas from [85], by which also the following games  $G_1$  and  $G_2$  are inspired.

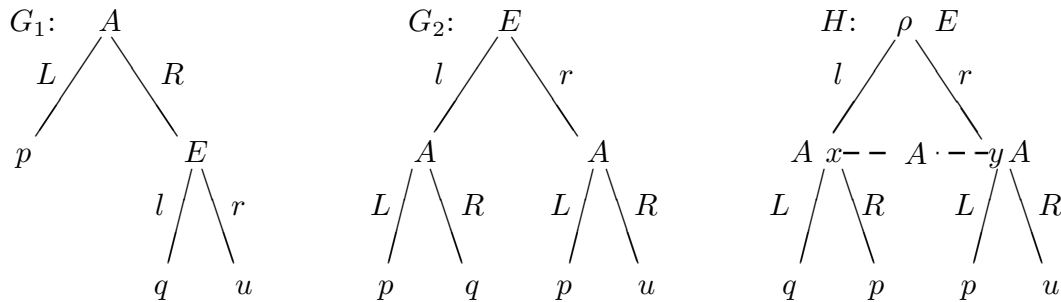


Figure 9. Two ‘similar’ games  $G_1$  and  $G_2$ .  $H$  is an imperfect information game: see Section 6

Figure 9 represents two extensive game forms  $G_1$  and  $G_2$  for two players  $E$  and  $A$ , to which some propositional information has been added to the leaves. The question put forward in [85] is when two games are the same. When looking at the power of players, encoded in  $\alpha$ -effectivity functions (cf. Section 2), the games  $G_1$  and  $G_2$  of Figure 9 are the same: the powers of  $E$  are  $\{\{p, q\}, \{p, u\}\}$  while  $A$  is effective for  $\{\{p\}, \{q, u\}\}$ . Also, the two games represent *evaluation games* (cf. [37]) for two formulas that are equivalent, viz.,  $p \wedge (q \vee u)$  and  $(p \wedge q) \vee (p \wedge u)$ : verifier has a winning strategy for both formulas in exactly the same models. Hence, if we are interested in the outcomes of games only,

the games  $G_1$  and  $G_2$  can be coined the same. (We come back to the ‘power level’ for describing games in Section 10.)

On the other hand, there is admittedly a difference between the two games, if alone for the feeling that  $A$  can hand over control to  $E$  to achieve  $u$  in the left hand figure, whereas in the right hand side the converse looks more an appropriate description. Also, the two games differ by the mere fact that different players start in each. A property true in  $G_1$  but not in  $G_2$  for instance is that, if  $E$  is ever to move, he can guarantee  $u$ . Following [85], we see that when the turns and the moves of the players are at stake, the two extensive game models above are not equivalent.

When notions relating to the *process* of a game are important, formalisms, mainly arising from computer science, like process algebra, the Hoare-Dijkstra-Floyd calculus, dynamic modal logic and temporal logics naturally come to surface, each with its own semantic notions concerning equality of structures, like (finite) trace equivalence, observational equivalence or bisimulation. Although also the more recent *BDI-logics* (logics addressing Beliefs, Desires and Intentions, see also Chapter 18) for reasoning about multi-agent systems have similar concerns, at this moment there is not yet a dominant framework in which actions of agents (or moves of players), the powers of coalitions and the informational and motivational attitudes of the players are all treated on a par. We restrict ourselves to some semantical observations here.

Taking moves as atomic actions in a PDL-like logic (see also Chapter 12) a property true in the root of  $G_2$  would be  $\langle l \cup r \rangle [L \cup R] (p \vee q)$ : indicating that there is an execution of the choice  $l \cup r$  so that every execution of the choice  $L \cup R$  leads to either  $p$  or  $q$ . Apart from the choice operator  $\cup$ , one usually also has the constructs  $;$  and  $*$  between programs, denoting sequential and iterative composition, respectively, and the test  $?$  for propositions. To stay close to the previous section, where there are no labels for the player’s choices, let us first mention that [35] assumes the following *atomic* actions (in our notation). For every player  $i \in N$  we add an action  $\mathbf{mo}_i$ , where  $R_{\mathbf{mo}_i}st$  is true iff if  $i$  is to move at  $s$ , he has a choice that leads to  $t$ . Next, for every strategy profile  $\sigma$ , relation  $R_{\text{step}(\sigma)}$  denotes the *one-step* transitions of  $\sigma$ , that is,  $R_{\text{step}(\sigma)}st$  iff  $\sigma(s) = t$ .

We now can distinguish the two models  $G_1$  and  $G_2$  with for instance the formula  $[\mathbf{mo}_E][\mathbf{mo}_A \cup \mathbf{mo}_E]\perp$  which is only true in  $G_1$ , expressing that there are no moves possible after  $E$  has played. Following and specialising for instance [12], we can even identify formulas  $\Phi_1$  and  $\Phi_2$  such that  $\Phi_i$  is true in a model  $M_i$  iff  $M_i$  is *bisimilar* to  $G_i$  above. The idea is simple, constructing a formula  $\varphi_{G,s}$  for every state  $s$  in a finite  $G$  inductively as follows. If  $s$  is a leaf, let  $\varphi_{G,s}$  be the finite conjunction of literals over atoms true in  $s$ , together with the property that there are no moves to be made, i.e.,  $\varphi_{G,s} = \bigwedge_{G,s \models p} p \wedge \bigwedge_{G,s \not\models p} \neg p \wedge \bigwedge_{i \in N} [\mathbf{mo}_i]\perp$ . Next, let  $t$  be a node such that for all its successors  $s_1, \dots, s_k$  in the tree,  $\varphi_{G,s_h}$  ( $h \leq k$ ) is already defined, and suppose that  $i$  is to move in  $t$ . Then

$$\varphi_{G,t} =_{\text{def}} \bigwedge_{G,t \models p} p \wedge \bigwedge_{G,t \not\models p} \neg p \wedge \bigwedge_{j \neq i} [\mathbf{mo}_j]\perp \wedge \bigwedge_{h \leq k} \langle \mathbf{mo}_i \rangle \varphi_{G,s_h} \wedge [\mathbf{mo}_i] \left( \bigvee_{h \leq k} \varphi_{G,s_h} \right)$$

It will be clear from this construction, that for any extensive finite game  $G$  with root  $r_0$  we have that for any model  $M$  and state  $s$ ,  $M, s \models \varphi_{G,r_0}$  iff  $M, s$  is bisimilar to  $G, r_0$ . On game models, bisimulation boils down to *isomorphism*, but the notion is useful when reasoning about games as *graph automata* (see also Chapter 12 and [44]).

Observing that  $\neg\langle\mathbf{step}(\sigma)\rangle\top$  characterises the leaves in the game tree, and using **while**  $\varphi$  **do**  $\pi$  as shorthand for  $(\varphi? ; \pi)^* ; \neg\varphi?$  (‘as long as  $\varphi$ , perform  $\pi$ ’), we can define, for every coalition  $M = \{i_1, \dots, i_m\} \subseteq N$  a program that, given  $\sigma$ ,  $N \setminus M$  adheres to it, but  $M$  is allowed to deviate:

$$\Pi(\sigma, M) =_{df} \mathbf{while} \langle \mathbf{step}(\sigma) \rangle \top \mathbf{do} (\mathbf{step}(\sigma) \cup \mathbf{mo}(i_1) \cup \dots \cup \mathbf{mo}(i_m))$$

The operator  $[\sigma]$  of the previous section can now be defined as  $[\Pi(\sigma, \emptyset)]$ , whereas  $[\sigma, i]$  becomes  $[\Pi(\sigma, \{i\})]$ , enabling us to express the characterisations of (s.p.) Nash equilibrium and best response, as in the previous section.

But one does not have to reason about just one strategy. Let us define **end** as being in a leaf:  $\mathbf{end} =_{df} \bigwedge_{i \in N} [\mathbf{mo}_i] \perp$ . We can then express that, when every player  $i$  adheres to his strategy  $\sigma_i$ , the game will terminate in a state satisfying  $\varphi$ , using the formula  $[(\bigcup_{i \in N} (\mathbf{mo}_i)? ; \mathbf{step}(\sigma_i))^*](\mathbf{end} \rightarrow \varphi)$ . Every player  $i$  playing strategy  $\sigma_i$  of course induces a strategy profile  $\sigma$ , and as such,  $\pi$  can also be expressed in the framework of [35], but we can now also express properties of intermediate states. Let us define  $\mathbf{plays}(i, \sigma_i)$  as  $(\mathbf{mo}_i)? ; \mathbf{step}(\sigma_i)$ , and, for a set of strategies  $\Sigma_M$ , one  $\sigma_i$  for each  $i \in M \subseteq N$ ,  $\mathbf{Plays}(M, \Sigma_M)$  as  $\bigcup_{i \in M} ((\mathbf{mo}_i)? ; \mathbf{step}(\sigma_i))$ , meaning that every agent  $i$  in  $M$  will play his strategy  $\sigma_i$ . Let us denote each agent  $j$ ’s atomic actions (or choices) with  $Ac_j$ . Also, let  $\mathbf{any}(j)$  mean  $(\mathbf{mo}_j)? ; \bigcup_{a_j \in Ac_j}$ , whereas  $\mathbf{Any}(M) =_{df} \bigcup_{j \in M} \mathbf{any}(j)$ . Then,

$$[(\mathbf{Plays}(M, \Sigma_M) \cup \mathbf{Any}(N \setminus M))^*]\varphi \quad (4)$$

expresses that coalition  $M$  can, by choosing the strategies  $\Sigma_M$ , ensure that, no matter what the other players  $j \in N \setminus M$  will do,  $\varphi$  will invariantly be true. This implies that  $M$  has a strategy to ensure  $\varphi$ , which, as we will see in Section 11, is the basic expression of *ATL*. When abbreviating (4) as  $\langle M, \Sigma_M \rangle \varphi$ , that basic expression of *ATL* can be expressed as  $\bigvee_{\Sigma_M} \langle M, \Sigma_M \rangle \varphi$ , where  $\Sigma_M$  ranges over sets of strategies for players in  $M$ . It is interesting to note that every strategy in a finite game model is definable in *PDL*, using the characteristic formulae  $\varphi_{G,s}$  for  $G, s$ : simply observe that every transition labelled with choice  $a_i$  from  $s$  to  $t$  can be denoted by  $(\varphi_{G,s})? ; a_i(\varphi_{G,t})?$ . However, since in general there will be exponentially many strategies, this is arguably more of technical interest than of practical value.

### 3.4 Other Issues

Until now we have focused mainly on (Nash) equilibria and best response strategies. Recall from Zermelo’s theorem (Theorem 7) that in every finite win-loss game for two players, exactly one player has a winning strategy. Let us denote player’s  $i$  win by  $\mathbf{win}_i$ . Following [85] once more, we can define predicates  $\mathbf{Win}_i$ , meaning that, at the current node,  $i$  has a winning strategy (let  $i \neq j$ ):

$$\mathbf{Win}_i \equiv (\mathbf{end} \wedge \mathbf{win}_i) \vee (\mathbf{turn}_i \wedge \langle \mathbf{any}(i) \rangle \mathbf{Win}_i) \vee (\mathbf{turn}_j \wedge [\mathbf{any}(j)] \mathbf{Win}_i)$$

This hints at an inductive definition for  $\mathbf{Win}_i$  using a *least fixed-point* schema

$$\mathbf{Win}_i =_{def} \mu P \cdot (\mathbf{end} \wedge \mathbf{win}_i) \vee (\mathbf{turn}_i \wedge \langle \mathbf{any}(i) \rangle P) \vee (\mathbf{turn}_j \wedge [\mathbf{any}(j)] P)$$

The reader immediately recognises the above as a formula in the  $\mu$ -calculus, which is the topic of Chapter 12 of this handbook. As another example, the expression  $\mu P \cdot (\text{end} \wedge \varphi) \vee (\text{turn}_i \wedge \langle \text{any}(i) \rangle P) \vee (\text{turn}_j \wedge [\text{any}(j)]P)$  says that  $i$  has a strategy for guaranteeing a set of outcomes in which  $\varphi$  is true. Note that this is already expressible in a *PDL*-like logic: just choose  $N = M = \{i, j\}$  in equation (4). In Section 11 we will discuss the logic *ATL* which is specifically designed to reason about what agents, and indeed, *coalitions* can guarantee to hold.

Still, the relation between the  $\mu$ -calculus and games is an interesting one. The calculus provides a very natural way, using its fixed point definitions, to reflect the equilibrium character of game-theoretic notions. Specifically the connections between the  $\mu$ -calculus and games of possibly infinite duration are appealing: going back to an idea of [16], we know that *any* formula of the  $\mu$ -calculus expresses the existence of a strategy in a certain game.

As seen above, if the goal of a player is to reach some desirable position in finite time, the set of positions that guarantee the win can be computed as a *least fixed point*. However, when the aim is to stay forever within a set of some safe positions, the winning set can be presented as a *greatest fixed point*. (This is very reminiscent of the distinction between *liveness* and *safety properties* used in computer science, as first introduced in [43]. See also [40] for a survey and Chapter 12 of this handbook.) More sophisticated winning conditions arise naturally in games modelling potentially infinite behavior of reactive systems. In general, mutually dependent least and greatest fixed point operators are necessary. [55] (from which the current two paragraphs borrow heavily), suggests that this interplay between least and greatest fixed points may well be the secret of the success of the  $\mu$ -calculus: "...in contrast to first-order or temporal logic, the  $\mu$ -calculus did not emerge by a formalization of the natural language". The  $\mu$ -calculus also can benefit from game theory, since the game semantics reduces  $\mu$ -calculus model checking to solving (parity) games: for more on this, see Chapter 12 of this handbook. A first impression of the complexity of such problems is given in Section 11. Another example of using games to settle complexity issues is provided in [12], where a *two person corridor tiling game* is used to prove the EXPTIME-hardness of PDL.

## 4 EPISTEMIC LOGIC

We saw in Section 2 that it makes sense to be explicit about the amount of information that each player has, at a given state of the game. Epistemic logic studies the notion of knowledge, and since [36], a mainstream in formal approaches to knowledge and belief is grounded in a possible world semantics. In the 1990's, these approaches were further developed in areas like computer science, cf. [21] – originally motivated by the need to reason about communication protocols – and artificial intelligence, cf. [49] and [51] – to reason about epistemic preconditions of actions. From the early days of game theory (cf. [4]) it has been recognised that the amount of knowledge that agents have is crucial in many solution concepts. But the formalisation of knowledge in game theory only took off since the late nineties, partially due to the TARK and LOFT events ([77, 45]).

The monograph [11] distinguishes between the notions of perfect/imperfect information on the one hand, and those of complete/incomplete information on the other. A game is of *perfect information* if the rules specify that the players always know 'where they are': for games in extensive form this means that each player is free in every node

axioms and rules for $\mathbf{S5}_m$ , where $i \leq m$		multi-agent notions	
A1	any axiomatisation for propositional logic	A6	$E\varphi \leftrightarrow (K_1\varphi \wedge \dots \wedge K_m\varphi)$
A2	$(K_i\varphi \wedge K_i(\varphi \rightarrow \psi)) \rightarrow K_i\psi$	A7	$(C\varphi \wedge C(\varphi \rightarrow \psi)) \rightarrow C\psi$
A3	$K_i\varphi \rightarrow \varphi$	A8	$C\varphi \rightarrow \varphi$
A4	$K_i\varphi \rightarrow K_iK_i\varphi$	A9	$C\varphi \rightarrow EC\varphi$
A5	$\neg K_i\varphi \rightarrow K_i\neg K_i\varphi$	A10	$C(\varphi \rightarrow E\varphi) \rightarrow (\varphi \rightarrow C\varphi)$
R1	$\vdash \varphi, \vdash \varphi \rightarrow \psi \Rightarrow \vdash \psi$		
R2	$\vdash \varphi \Rightarrow \vdash K_i\varphi$		
		R3	$\vdash \varphi \Rightarrow \vdash C\varphi$

Table 2.

to make a decision independent of that in other nodes. A game is of *complete information* if everything is known about the circumstances under which the game is played, like the probability that nature chooses a certain outcome, and who the opponent is and how risk-averse he is. In a game with incomplete information, players do not necessarily know which game they are playing, or who the other players are. Such games, although realistic, are up to now mainly the domain of a research area called *evolutionary game theory*, utilising theories of learning and evolutionary computation (see for an overview, [79]). Epistemic logic is nowadays widely used to express various degrees of imperfect information in a game; assumptions about the completeness of information are still often made on a meta-level.

Modal epistemic logic, the logic of knowledge, provides a very natural interpretation to the accessibility relation in Kripke models. For an agent  $i$ , two worlds  $w$  and  $v$  are connected (written  $R_i wv$ ), if the agent cannot (epistemically) distinguish them. In other words, we have  $R_i wv$  if, according to  $i$ 's information at  $w$ , the world might as well be in state  $v$ , or that  $v$  is compatible with  $i$ 's information at  $w$ . Using this interpretation of access,  $R_i$  is obviously an equivalence relation. Readers familiar with game theory may be best acquainted with epistemic notions in this field as summarised in [5]. In that terminology, our set of states  $S$  is a space  $\Omega$  of states of the world, our equivalence relation  $R_i$  is [5]'s partition  $\mathcal{F}_i$  of  $\Omega$ , and our *formulas* correspond to [5]'s *events*. Also, our equivalence relations connect in an obvious way to the partitions mentioned in Definition 8.

The epistemic modal language for  $m$  agents is obtained by allowing a modal operator  $K_i$  for every agent  $i \leq m$ , with  $K_i\varphi$  meaning: agent  $i$  knows  $\varphi$ . The corresponding Kripke models are  $M = \langle W, R_1, R_2, \dots, R_m, \pi \rangle$ , with each  $R_i$  being an equivalence relation. Thus, we are in the realm of the multi-modal logic  $\mathbf{S5}_m$ , of which the axioms are summarised in Table 2. They express that knowledge is *closed under consequences* (A2), it is *veridical* (A3) and agents are both *positively* and *negatively* introspective (A4 and A5, respectively). Moreover, all agents know the  $\mathbf{S5}_m$ -theorems. Clearly, these properties represent *logically omniscient* agents (see also Chapter 18), and hence adopting this particular logic for the knowledge of players assumes an ideal case of perfect reasoners. For an overview of weakening the axioms to tackle logical omniscience, we refer the reader to [21, 49].

**THEOREM 17.** *We have the following facts for  $\mathbf{S5}_m$ , which is the logic summarised in*

the left hand side of Table 2.

1. The system  $\mathbf{S5}_m$  is sound and complete with respect to Kripke models in which every  $R_i$  is an equivalence relation. The complexity of the satisfiability problem for  $\mathbf{S5}_m$  if  $m > 1$  is PSPACE-complete (cf. [21]).
2. Taking  $m = 1$ , the logic  $\mathbf{S5}_1$  is also sound and complete for the semantics where  $R$  is the universal relation. The complexity of the satisfiability problem for  $\mathbf{S5}_1$  is the same as for propositional logic: it is NP-complete. Moreover, every formula  $\varphi$  is equivalent in  $\mathbf{S5}_1$  to a formula without any nesting of modal operators (cf. [49]).

EXAMPLE 18. Let us, as an example, consider the  $S5_3$  model **hexa**, taken from [98], which focuses on the knowledge and ignorance of players in games with imperfect information, by introducing the notion of *knowledge games*, games in which the knowledge of the players, and the effect of their moves upon this knowledge, is described. In **hexa**, we have three players (1, 2 and 3) and three cards, each with a neutral side and a coloured face:  $r$  (red),  $w$  (white) or  $b$  (blue). If a player holds a card, he is the (initially only) player that knows its colour. See Figure 10 for the Kripke model **hexa** representing the knowledge of the players after the three cards have been dealt. The state  $rw b$  represents the deal where player 1 holds  $r$ , 2 holds  $w$ , and 3 holds  $b$ ; this distribution is denoted in the object language as  $\delta_{rw b} = r_1 \wedge w_2 \wedge b_3$ . From now on, we will underline the ‘actual’ state of the model: **hexa** thus represents the knowledge of the players given that the actual deal is  $rw b$ . Reflexive access is not represented, thus it is understood that both  $rw b$  and  $rbw$  are 1-accessible from  $rw b$ : given that 1 has the red card, he does not know whether the deal is  $rw b$  or  $rbw$ .

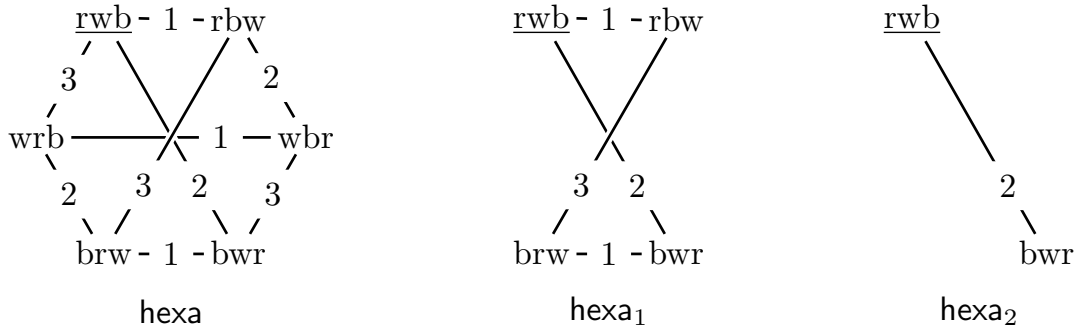


Figure 10. The initial model **hexa**; **hexa<sub>1</sub>** and **hexa<sub>2</sub>** are obtained using updates (cf. Section 7)

Interestingly, **hexa** not only tells us that every agent knows its own card (we have for instance  $\text{hexa}, rw b \models K_1 r_1$ , and, more generally,  $\text{hexa}, c_1 c_2 c_3 \models K_i c_i$  ( $c_i \in \{r, w, b\}, i \leq 3$ ), but also that everybody knows this. So, for instance, we have, and this is independent of the actual deal,  $\text{hexa} \models K_1 \bigwedge_{c_2, c_3 \in \{r, w, b\}} ((c_2 \rightarrow K_2 c_2) \wedge (c_3 \rightarrow K_3 c_3))$ . And again, it is also the case that player 3 knows this. The aim of the game initiated by **hexa** is to find out the distribution of the cards, and we will return to the model when studying the dynamics of epistemics, in Section 7.  $\dashv$

Indeed, the description of the situation that we gave is *common knowledge*, which is an intriguing multi-agent epistemic notion. Let us define ‘everybody knows’ ( $E\varphi$ ) as is done in axiom A6 in Table 2 then the remaining axioms and rule on the right side of this table capture the intuition that common knowledge of  $\varphi$  models  $E\varphi \wedge EE\varphi \wedge EEE\varphi \wedge \dots$ . Indeed, if we denote the axiom system represented in Table 2 by  $\mathbf{S5}C_m$ , then one easily checks that  $\vdash_{\mathbf{S5}C_m} C\varphi \rightarrow E^n\varphi$ , for arbitrary  $n \in \mathbb{N}$ , and, conversely, that  $\{E^n\varphi \mid n \in \mathbb{N}\} \models C\varphi$ : Semantically, the accessibility relation for  $R_E$ , with respect to which  $E$ -knowledge is the necessity, is  $\bigcup_{i \leq m} R_i$ , and then  $R_C$  (the relation for common knowledge) is the transitive closure  $R_E^*$  of  $R_E$ . We will denote any dual operator  $\neg X \neg$  of  $X$  with  $\hat{X}$ .

**EXAMPLE 19** (The muddy children). In this example the principal players are a father and  $k$  children, of whom  $m$  (with  $m \leq k$ ) have mud on their foreheads. The father calls all the children together. None of them knows whether it is muddy or not, but they can all accurately perceive the other children and judge whether they are muddy. This all is common knowledge. Now the father has a very simple announcement (5) to make:

*At least one of you is muddy. If you know that you are muddy, step forward.* (5)

After this, nothing happens (except in case  $m = 1$ ). When the father notices this, he literally repeats the announcement (5). Once again, nothing happens (except in case  $m = 2$ ). The announcement and subsequent silence are repeated until the father’s  $m$ -th announcement. Suddenly all  $m$  muddy children step forward!

Let us analyse the muddy children problem semantically, where we have 3 children. In Figure 11, the initial situation is modelled in **twomud** (we come back to a formal analysis of the story in Section 7). Worlds are denoted as triples  $xyz$ . The world 110 for instance denotes that child  $a$  and  $b$  are muddy, and  $c$  is not. Given the fact that every child sees the others but not itself, we can understand that agent  $a$  ‘owns the horizontal lines’ in the figure, since  $a$  can never distinguish between two states  $0yz$  and  $1yz$ . Similar arguments apply to agents  $b$  and  $c$ .

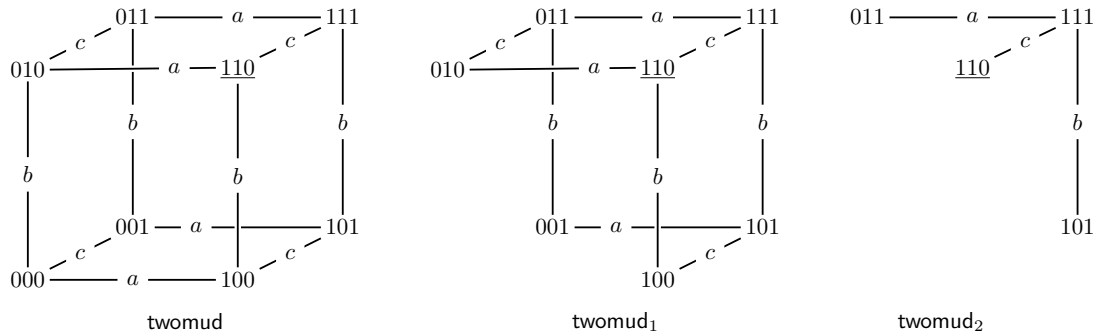


Figure 11. Muddy Children: initial situation **twomud** and after two updates (**twomud<sub>1</sub>** and **twomud<sub>2</sub>**)

Let us see what epistemic truths we have in the state (**twomud**,  $s$ ), with  $s = 110$ . The only propositional atoms we use are  $m_i$  ( $i = a, b, c$ ) with meaning ‘child  $i$  is muddy’. In state  $s$ , we then have for instance  $\neg(K_a m_a \vee K_a \neg m_a)$  (agent  $a$  does not know whether it is muddy), and also  $K_a(m_b \wedge \neg K_b m_b) \wedge K_a(\neg m_c \wedge \neg K_c \neg m_c)$  ( $a$  knows that  $b$  is muddy



without knowing it, and also that  $c$  is muddy without knowing that). Regarding group notions, we observe the following, in  $s$ . Let  $\ell$  denote that at least one child is muddy ( $\ell \equiv m_a \vee m_b \vee m_c$ ).

1.  $E\ell \wedge \neg Em_a \wedge \neg Em_b \wedge \neg Em_c$ ; everybody knows that there is at least one muddy child, but nobody is known by everybody to be muddy
2.  $K_c E\ell \wedge \neg K_b E\ell$ ;  $c$  knows that everybody knows that there is at least one muddy child, but  $b$  does not know that everybody knows at least one child to be muddy. To see the second conjunct, note that  $\text{twomud}, 100 \models \neg E\ell$ , hence  $\text{twomud}, s \models \neg K_b E\ell$ .
3.  $\neg C\ell$ ; It is not common knowledge that there is at least one muddy child! This follows immediately from the previous item, but also directly from the model: one can find a path from  $s = 110$  via  $010$  to  $000$ , the latter state being one at which no child is muddy. One easily verifies that we have even  $\text{twomud} \models \neg C\ell$ .  $\dashv$

**THEOREM 20** ([21]). *Let the logic  $\mathbf{S5C}_m$  be summarised in Table 2. Then,  $\mathbf{S5C}_m$  is sound and complete with respect to Kripke models in which every  $R_i$  is an equivalence,  $R_E$  (the relation for ‘everybody knows’) is  $\bigcup_{i \leq m} R_i$  and  $R_C$  (the relation for common knowledge) is  $R_E^*$ , i.e., the reflexive transitive closure of  $R_E$ . The complexity of the satisfiability problem for  $\mathbf{S5C}_m$  if  $m = 1$  is PSPACE-complete, and the complexity for  $\mathbf{S5C}_m$  with  $m > 1$  is EXPTIME-complete.*

As observed in [12, Chapter 6.8], the presence of a pair of modalities, one for a relation and the other for its reflexive transitive closure (in our case:  $E$  and  $C$ , respectively) is often –not always– an indication for entering the realm of EXPTIME-complexity results, since it enables one to force exponentially deep models, and to code the corridor tiling problem.

## 5 INTERPRETED SYSTEMS

Whereas the properties of knowledge as summarised in Table 2, especially that of negative introspection, have been under continuous debate and critique, the *Interpreted Systems* approach to knowledge as advocated by [21] in fact gives a *computationally grounded semantics* to the  $\mathbf{S5}_m$  properties of knowledge. Rather than assuming the equivalence relation to be somehow given, in an interpreted system they naturally arise from the way a system is modeled. The idea is simple: in an interpreted system  $\mathcal{I}$  we have  $m$  agents, or processors, each with its own local state  $s_i$ . A processor is aware of its own local state: two global states  $\mathbf{s}$  and  $\mathbf{s}'$  are the same for  $i$  if its local states in both coincide. And this notion of ‘sameness’ is an equivalence relation, yielding the  $\mathbf{S5}_m$  properties in a natural way.

To formally define an interpreted system for  $m$  agents  $\mathcal{I}_m$  we first give the notion of a global state. Let us assume that every agent  $i$  can be in a number of states  $L_i$ . Apart from the agents’ local states, there is also a set of environment states  $L_e$ , which keeps track of, e.g., whether a communication line is up or down, or what the actual deal of cards is. A global state  $\mathbf{s}$  is then a tuple  $\langle s_e, s_1, s_2, \dots, s_m \rangle \in L_e \times L_1 \times L_2 \times \dots \times L_m$ . The set of global states of interest will be denoted  $\mathcal{G} \subseteq L_e \times L_1 \times L_2 \times \dots \times L_m$ . Local states, both that of the agents and that of the environment, may change over time. A priori, there are no constraints on how the system may evolve: a *run* over  $\mathcal{G}$  is a sequence

of states, or, rather, a function  $r$  from time  $\mathbb{N}$  to global states. The pair  $(r, n)$  of a run and a time point is also referred to as a point. Let  $r(n) = \langle s_e, s_1, s_2, \dots, s_m \rangle$  be the global state at time  $n$  in run  $r$ , then with  $r_j(n)$  we mean  $s_j$ , where  $j$  ranges over  $e, 1, \dots, m$ . Now, a *system*  $\mathcal{R}$  over  $\mathcal{G}$  is a set of runs over  $\mathcal{G}$ .

In general, formulas are now going to be interpreted in a point  $(r, n)$  in an interpreted system  $\mathcal{I}$ . To do so, we need to take care of atomic formulas and the epistemic operators. An *Interpreted System*  $\mathcal{I} = (\mathcal{R}, \pi)$  over  $\mathcal{G}$  is a system  $\mathcal{R}$  over  $\mathcal{G}$  with an interpretation  $\pi$  which decides for each point  $(r, n)$  and atom  $p \in At$ , whether  $p$  is true in  $(r, n)$  or not. Moreover, two points  $(r, n)$  and  $(r', n')$  are indistinguishable for  $i$ , written  $(r, n) \sim_i (r', n')$ , if  $r_i(n) = r'_i(n')$ , or, in other words, if  $i$ 's local states in both points are the same. Definition (6) expresses that agent  $i$  knows  $\varphi$  in a point in  $\mathcal{I}$  if  $\varphi$  is true *given agent  $i$ 's local information*:

$$(\mathcal{I}, r, n) \models K_i \varphi \text{ iff } (\mathcal{I}, r', n') \models \varphi \text{ for all } (r', n') \sim_i (r, n) \quad (6)$$

We assume here that the interpretation function  $\pi$  only depends on the global state  $r(n)$ , and not the history of the point  $(r, n)$ . This is in line with [21], but deviates from [30]. In the context of knowledge and linear time, the valid formulas are the same with and without this assumption: see [31] for an explanation of this. In practice,  $\pi$  will depend on just local information of a global state, denoting, e.g., whether a variable of processor  $i$  has a certain value, or whether player  $j$  holds the ace of hearts in a card game.

Of course, one may want to study epistemic logic for multi-agent systems in a static model  $M_{\mathcal{G}} = \langle \mathcal{G}, \sim_1, \dots, \sim_m, \pi \rangle$  based on a set of global states  $\mathcal{G}$ , and interpret formulas in global states themselves, rather than in a point in a run. If  $\mathcal{G}$  equals the full Cartesian product  $L_e \times L_1 \times L_2 \times \dots \times L_m$ , such models are called *hypercubes* in [47]. If we ignore the environment and only require that every combination of individual local states occurs we have a *full system*. Note that hexa of Figure 10 is a full system, whereas hexa<sub>1</sub> is obviously neither full nor a hypercube.

Full systems are appropriate classes of models to specify initial configurations of multi-agent systems, in which no agent has any information about any other agent, or about the environment. They are obviously **S5**<sub>m</sub>-systems, but interestingly enough they satisfy an additional property. It is not hard to see that the operator  $E$  for *everybody knows* from Section 4 semantically corresponds to interpreting  $E$  as the necessity operator of the relation that is the *union* of all the individual relations  $R_i$ : everybody knows  $\varphi$  if nobody thinks it possible that  $\neg\varphi$  is true. Note that  $\hat{E}\varphi$  means that some agent considers it possible that  $\varphi$ : for some state that is  $R_E = \cup_{i=1\dots m} R_i$ -accessible,  $\varphi$  is true. Axiom 7 then, on top of the axioms of **S5**<sub>m</sub>, is needed to axiomatise hypercubes and full systems (cf. [47]):

$$\bigwedge_{i=1\dots m} \hat{E}\varphi_i \rightarrow \hat{E}\hat{E} \bigwedge_{i=1\dots m} \varphi_i \quad (7)$$

In this scheme,  $\varphi_i$  is an  $i$ -local formula, to which we will come back shortly. Roughly, an  $i$ -local formula  $\varphi_i$  characterises  $i$ 's knowledge: its truth value is constant within  $i$ 's reachable states. The scheme then says that if we can reach, using  $R_E$ , a knowledge state for every agent, we can reach the 'combined knowledge state' in two steps.

When having more involved group notions of knowledge, full systems and hypercubes share some other specific properties. For instance, consider *distributed knowledge*  $D\varphi$ , in

which  $D$  is the necessity operator for the *intersection* of the individual relations  $R_i$ . This notion of knowledge arises when all agents would communicate with each other: if one agent rules out that  $s$  is a possible state, after communication nobody would consider  $s$  anymore. Obviously, in  $\mathbf{S5}_m$ , nobody rules out the current state, so that distributed knowledge in  $\mathbf{S5}_m$  satisfies  $D\varphi \rightarrow \varphi$ . But in a full system with an empty environment, we also have the converse:  $\varphi \rightarrow D\varphi$ . This is easy to see: the only global state that is  $i$ -similar to  $\mathbf{s} = \langle \emptyset, s_1, s_2, \dots, s_m \rangle$  is  $\mathbf{s}$  itself!

Regarding common knowledge, in hypercubes and full systems it is either absent or else globally present: common knowledge is the same in all local states! This is easy also: if  $C\varphi$  would hold in  $\mathbf{s} = \langle s_e, s_1, s_2, \dots, s_m \rangle$  but not in  $\mathbf{s}' = \langle s'_e, s'_1, s'_2, \dots, s'_m \rangle$ , since the system is full, we find a state  $\mathbf{t} = \langle \cdot, s_1, s'_2, \dots, \cdot \rangle$ . Recalling that  $R_C$  is the transitive closure of the  $R_i$ 's, and that  $R_1\mathbf{s}\mathbf{t}$  and  $R_2\mathbf{t}\mathbf{s}'$ , we see that  $\langle s_e, s_1, s_2, \dots, s_m \rangle$  and  $\langle s'_e, s'_1, s'_2, \dots, s'_m \rangle$  must agree on  $C\varphi$ .

The logic of *local propositions* introduced in [20] connects the notion of accessibility in a system  $M_G$ , in fact, in any epistemic model  $M = \langle W, R_1, \dots, R_m, \pi \rangle$ , with a syntactic one. Let a proposition  $U$  in  $M$  just be a subset of  $W$ . It is an  $i$ -local proposition if for all  $u, w \in W$  with  $R_i u w$ , we have  $u \in U$  iff  $w \in U$ . In words: an  $i$ -local proposition is determined by  $i$ 's local state, as is his knowledge. (In *hexa* for instance,  $r_1$ , denoting that 1 holds a red card, is 1-local.) For an atom  $p$ , say that  $M \sim_p M'$  if the only difference between  $M$  and  $M'$  can be  $\pi(p)$  and  $\pi'(p)$ . Define  $M, w \models \exists_i p(\varphi)$  iff for some  $M'$  with  $M' \sim_p M$ , we have  $M', w \models \varphi$ , where  $p$  is an  $i$ -local propositional atom, i.e.,  $\pi(p)$  is an  $i$ -local proposition in  $M$ . Let  $\Box$  be the universal operator, denoting what is true globally. Let  $q$  not occur in  $\varphi$ . Then:

$$M, w \models K_i \varphi \text{ iff } M, w \models \exists_i q (q \wedge \Box (q \rightarrow \varphi)) \quad (8)$$

(A similar ‘reduction’ can be given for distributed and common knowledge, see [20].) We have seen above in (7) that local propositions play a role in axiomatising hypercubes and full systems. But they have greater use: note that (8) implies that if the object language is rich enough to describe the local state of the agents, we can replace epistemic operators by occurrences of  $\Box$  and local propositions. This idea is applied in [38] to ‘reduce’ model-checking of epistemic temporal properties to properties that can be handled by a ‘conventional’ model-checker SPIN that does not address knowledge explicitly: for a knowledge property  $K_i \varphi$  to be checked in  $s$ , the user provides a local proposition  $q_i$  and the problem is then reduced to (1) checking whether  $q_i$  is indeed  $i$ -local, (2) whether it is true in  $s$ , and (3) whether the implication  $q_i \rightarrow \varphi$  globally holds. To illustrate this, in *hexa* for instance, rather than saying, that  $\text{hexa}, rwb \models K_1(w_2 \vee w_3)$  (1 knows that the white card is owned by either player 2 or 3) we can stipulate:  $\text{hexa}, rwb \models r_1 \wedge \Box(r_1 \rightarrow (w_2 \vee w_3))$  (currently, player 1's local state reads ‘red card’, and globally, in such a situation either 2 or 3 holds the white card).

Let us now return to interpreted systems, where the dynamics is modelled through the notion of runs. Runs may look rather abstract, but following [21] one can think about them as being brought about by the agents while following a *protocol*, in which agents take certain *actions*. This is reminiscent of our notion of *strategies* in an extensive game: they restrict the space of all possible evolutions of the system. Having a language with operators for individual, distributed and common knowledge is still too poor to reason about interpreted systems: one easily shows that any two points  $(r, n)$  and  $(r', n')$  with the same global states  $r(n) = r'(n')$  verify the same epistemic properties. Indeed, it is

natural to add *temporal* operators  $\bigcirc$  (next time),  $\Box$  (always),  $\Diamond$  (eventually) and  $\mathcal{U}$  (until), with truth conditions like (for the formal interpretation of the other temporal operators, and a more extensive treatment of temporal logic, we refer to Chapters 11 and 17 of this Handbook):

$$(\mathcal{I}, r, n) \models \bigcirc \varphi \text{ iff } (\mathcal{I}, r, n+1) \models \varphi \quad (9)$$

With these operators, one *can* in general distinguish different points with the same global state: given a specific deal of cards for instance, it is perfectly well possible that in one play of the game player 1 will win, when in another play he will not: we can have  $r(n) = r'(n')$ , and  $\mathcal{I}, r, n \models \Diamond \text{win}_1$  but  $\mathcal{I}, r', n' \not\models \Diamond \text{win}_1$ . In the temporal language, one can express that a certain property  $\varphi$  will occur infinitely often ( $\Box \Diamond \varphi$ ) or almost always ( $\Diamond \Box \varphi$ ). But the full character of the language of course comes to the fore in temporal-epistemic properties. Examples include *not\_cross*  $\mathcal{U} K_i \text{safe}$ , and  $K_i \Box ((\neg K_j p \wedge \bigcirc K_j p) \rightarrow \bigcirc \bigcirc K_h K_j p)$  expressing that  $i$  will not cross the street until he knows it is safe, and that  $i$  knows that as soon as  $j$  learns that  $p$ , this will immediately be communicated to  $h$ .

Axioms for Linear Temporal Logic are given in Table 3: the operators  $\Diamond$  and  $\Box$  can be defined in terms of  $\bigcirc$  and  $\mathcal{U}$ . Regarding the soundness of the inference rule  $RU$ , assume that  $\mathcal{I} \models \varphi \rightarrow (\neg \psi \wedge \bigcirc \varphi)$ . Since every occurrence of  $\varphi$  guarantees its truth in the next time, it is easy to see that we have  $\mathcal{I} \models \varphi \rightarrow \Box \varphi$ . And, since  $\varphi$  comes with  $\neg \psi$  we also have  $\mathcal{I} \models \varphi \rightarrow \Box \neg \psi$ . Now, for an arbitrary  $\chi$ , note that  $\chi \mathcal{U} \psi$  being true would imply that  $\psi$  becomes true some time. But given  $\varphi$ , we just saw that  $\psi$  is always false, so we cannot have  $\chi \mathcal{U} \psi$  when we have  $\varphi$ , hence  $\mathcal{I} \models \varphi \rightarrow \neg(\chi \mathcal{U} \psi)$ .

$T1$	$(\bigcirc \varphi \wedge \bigcirc (\varphi \rightarrow \psi)) \rightarrow \bigcirc \psi$	$T2$	$\bigcirc \neg \varphi \leftrightarrow \neg \bigcirc \varphi$
$T3$	$\varphi \mathcal{U} \psi \leftrightarrow \psi \vee (\varphi \wedge \bigcirc (\varphi \mathcal{U} \psi))$		
$Nec$	$\vdash \varphi \Rightarrow \vdash \bigcirc \varphi$	$RU$	$\vdash \varphi \rightarrow (\neg \psi \wedge \bigcirc \varphi) \Rightarrow \vdash \varphi \rightarrow \neg(\chi \mathcal{U} \psi)$

Table 3. Axioms for Linear Temporal Logic **LTL**.

Note that in definition (6) in general we do not require that  $n = n'$ , so agents are not assumed to know what time it is. More generally, our definition allows the environment to change without any agent noticing this: two global states  $\langle s_e, s_1, s_2, \dots, s_m \rangle$  and  $\langle s'_e, s_1, s_2, \dots, s_m \rangle$  look the same for all agents, but still may have a different environment. In fact, the definition of an interpreted system is so general that some agents  $i$  may sense changes in the environment (e.g., when  $L_e \subseteq L_i$ ), while other agents  $j$  may not (when their local state  $L_j$  has ‘nothing to do’ with  $L_e$ ).

Although in the general case no agent knows the time, in games it is usually assumed that all players at least know how many moves have been played. Indeed, this is, more often than not, assumed to be common knowledge. Games, and many other forms of competition and cooperation in multi-agent systems often assume that such a system is *synchronous*. To capture this, since the agents’ knowledge is determined by their local state, we must encode a clock, or the number of passed ‘rounds’, in such local states. A system  $\mathcal{R}$  is *synchronous* if each agent can distinguish different time points: if  $n \neq n'$  then  $(r, n) \not\sim_i (r', n')$ , for any run  $r$  and  $r'$ . In other words, the local state of any agent must be different at different time points.

If we think of a run  $r$  as a sequence of global states in which each agent  $i$  is aware of its own local state  $r_i(n)$  at time  $n$ , the question arises how much an agent *memorises* when going from  $r_i(n)$  to  $r_i(n+1)$ . In the ideal case,  $i$  would ‘remember’ all local states  $r_i(k)$  with  $k \leq n$  when the time is currently  $n$ . In principle, this is the situation in games like chess and many other board and card games: each player has *perfect recall* of what he experienced during the game (although in practice, of course, humans and even machines may not have ‘enough memory’ for this). In a similar way as the requirement that  $i$  knows how many rounds have passed made us add this information to his local state, if we want the agent to remember exactly what has happened, from his perspective up to time  $n$ , we have to encode his previous local states  $r_i(k)$  ( $k < n$ ) in his current local state at time  $n$ . Let therefore agent  $i$ ’s *local-state sequence* at the point  $(r, n)$  be his stutter-free local past  $sflp_i(r, n) = \langle r_i(0), \pm r_i(1), \dots, \pm r_i(n-1) \rangle$ , where  $\pm r_i(x)$  means that  $r_i(x)$  appears in the sequence iff it is different from its immediate predecessor in the sequence. Then, we say that  $i$  has *perfect recall in the system*  $\mathcal{R}$  if  $(r, n) \sim_i (r', n')$  implies that  $sflp_i(r, n) = sflp_i(r', n')$ , that is, if the agent remembers his local-state sequence, he has no uncertainty about what happened. We abstract from stuttering, since if the agent does not notice a change in his local state, he does not know ‘how much is happening’—except in the synchronous case, when  $sflp_i(r, n)$  is the complete sequence  $\langle r_i(0), r_i(1), \dots, r_i(n-1) \rangle$ .

Does perfect recall for agent  $i$  mean that  $K_i\varphi \rightarrow \Box K_i\varphi$  is valid? Not in general, and mainly not because  $\varphi$  might refer to the current time. Knowing that ‘today is Wednesday’ does and should not imply that you always know that ‘today is Wednesday’. Likewise, ignorance need not persist over time, and hence neither should the knowledge about it: even with perfect recall,  $K_i\neg K_ip \rightarrow \Box K_i\neg K_ip$  should not hold: it would be equivalent to  $\neg K_ip \rightarrow \Box \neg K_ip$  (note that in  $\mathbf{S5}_m$ ,  $\neg K_i\varphi$  is equivalent to  $K_i\neg K_i\varphi$ ). Indeed,  $\mathcal{I}, r, n \models K_i\neg K_i\varphi$  ‘only’ means that  $i$  knows to be ignorant in  $r$  at time  $n$ , this need not persist over time. Perfect recall would only require that at every point  $(r, n+k)$ , the agent knows that he *did not know*  $\varphi$  ‘when the time was  $n$ ’. A modal logic for belief revision in which one distinguishes operators  $B_0$  for ‘initial’ belief (holding before the revision) and  $B_1$  for ‘new’ beliefs (kept after the revision) is explored in [15]. The modal formulation of perfect recall of ignorance then becomes  $\neg B_0\varphi \rightarrow B_1\neg B_0\varphi$ . When we cannot distinguish between what *was* known and what *is* known, one might want to characterise the *stable* formulas for which  $K_i\varphi \rightarrow \Box K_i\varphi$  holds in systems with perfect recall (see [21] and also discussions of ‘only knowing’ [91] and of (Un-)Successful Updates in Section 7). We will return to the property of perfect recall in Section 6: to capture perfect recall in synchronous systems we need the following property:

$$\mathbf{PR} \quad K_i\bigcirc\varphi \rightarrow \bigcirc K_i\varphi \quad (i = 1 \dots m)$$

We now give some technical results concerning the classes of systems discussed, in Theorem 21. To do so, let  $\mathbf{S5}_m \oplus \mathbf{LTL}$  be the axioms for individual knowledge from Table 2, together with those for linear time from Table 3. This system assumes no interaction between knowledge and time. Also, let  $\mathbf{S5}_m^C \oplus \mathbf{LTL}$  add the axioms for common knowledge (r.h.s. of Table 2) to this. Let  $\mathcal{C}_m$  be the class of all interpreted systems for  $m$  agents,  $\mathcal{C}_m^{sync}$  those that are synchronous,  $\mathcal{C}_m^{pr}$  those that satisfy perfect recall, and let  $\mathcal{C}_m^{sync, pr}$  be those synchronous interpreted systems that satisfy perfect recall.

**THEOREM 21.** *All the following results are from [21, Chapter 8], except for the first part of item 3, which is from [93]. Unless stated otherwise, assume  $m \geq 2$ .*

1.  $\mathcal{C}_m$  has  $\mathbf{S5}_m \oplus \mathbf{LTL}$  as a sound and complete axiomatisation. The complexity of the validity problem for this class is PSPACE-complete. Adding common knowledge,  $\mathbf{S5}_m^C \oplus \mathbf{LTL}$  completely axiomatises  $\mathcal{C}_m$ , but the validity moves to being EXPTIME-complete.
2.  $\mathcal{C}_m^{\text{sync}}$ : synchrony does not add anything in terms of axiomatisation or the validity problem: they are exactly as for  $\mathcal{C}_m$ .
3.  $\mathcal{C}_m^{\text{pr}}$  is completely axiomatised by  $\mathbf{S5}_m \oplus \mathbf{LTL} + \{\mathbf{KT}\}$ , where  $\mathbf{KT}$  is defined as  $(K_i\varphi_1 \wedge \bigcirc(K_i\varphi_2 \wedge \neg K_i\varphi_3)) \rightarrow \hat{K}_i[K_i\varphi_1 \mathcal{U}(K_i\varphi_2 \mathcal{U} \neg\varphi_3)]$  (for a discussion, we refer to [93]). For  $m = 1$ , validity in  $\mathcal{C}_m^{\text{pr}}$  is doubly-exponential time complete, otherwise non-elementary time complete. Adding common knowledge to the language makes the validity problem  $\Pi_1^1$ -complete. Hence, when common knowledge is present there is no finite axiomatisation for  $\mathcal{C}_m^{\text{pr}}$ ; indeed, in this case there is not even a recursively enumerable set of axioms that is complete for validity in  $\mathcal{C}_m^{\text{pr}}$ .
4.  $\mathcal{C}_m^{\text{sync,pr}}$  is completely axiomatised by  $\mathbf{S5}_m \oplus \mathbf{LTL} + \{\mathbf{PR}\}$ . The complexity of validity is non-elementary time complete. Adding common knowledge, we again get a complexity of  $\Pi_1^1$  for the validity problem, and a negative result concerning finite axiomatisability for  $\mathcal{C}_m^{\text{sync,pr}}$ .

Our discussion of interpreted systems can only be limited. Rather than linear time, one may consider branching time logic, and apart from synchrony and perfect recall, one may consider properties with or without assuming a unique *initial state*, and with or without the principle of *no learning*—the ‘converse of perfect recall’. Only these parameters all together yield 96 different logics: for a comprehensive overview of the linear case we refer to [31], and for the branching time case, to [97]. Moreover, where this section’s exposition is mainly organised along the ideas in [21], there have been several other but related approaches to knowledge and time, or even knowledge and computation, of which we here only mention the distributed processes approach of [60]. The recent paper [89] provides a general picture of different logics for knowledge and time, by giving a survey of decidability and undecidability results for several logics.

Although in general, the *model checking* problem is computationally easier than that of validity checking, for logics of knowledge and time, in particular those with perfect recall, the complexity of both tasks is often the same (see [94]). Work still progresses, both in the theoretical and the practical realm. We already mentioned an approach that ‘reduces’ epistemic temporal properties to temporal ones in order to use a ‘standard’ model checker. But model checkers that explicitly deal with an epistemic language are now rapidly emerging. Systems to model-check knowledge and time include the system MCK ([22]), DEMO ([104]) and the system MCMAS ([46]).

Recently, there has been a broad interest in model checking dynamics of knowledge in specific scenarios, like in [95] (‘the dining philosophers’), and in [100] (‘the sum and product problem’). Apart from model checking *epistemic properties*, it is also interesting to address the *realisability problem* (does there exist a *protocol* such that a given property is satisfied) and the *synthesis problem* (generate a protocol that satisfies a given constraint, if it exists). Space prohibits us to go into the details, we refer to [96].

## 6 GAMES WITH IMPERFECT INFORMATION

Now that we have seen how Kripke models are perfectly fit to represent games (Section 3) and imperfect information (Sections 4 and 5), let us spend some words on representing and reasoning about the combination of the two. In games with imperfect information, on which we will focus in this section, players *do* know what the rules of the games are, and who they play against, but they do not necessarily know ‘where they are in the game’. In the game models of Section 3, this can be conveniently represented by using an indistinguishability relation for every player, as explained in Section 4. Game theorists call the members of each partition of such an  $S5$ -equivalence relation usually *information sets*. A game of perfect information would then just be the special case in which every information set contains exactly one node.

As a simple example of an imperfect information game, let us consider game  $H$  of Figure 9. We assume that we have the standard knowledge assumptions of  $S5$ , which semantically mean that the indistinguishability relation in that figure is an equivalence relation: however, we do not represent reflexive arrows, so that the only uncertainty in the game is represented by the dotted line labelled with player  $A$ . So, what is modelled in  $H$  is that player  $E$  makes a first move ( $l$  or  $r$ ), and after that,  $A$  has to move, without knowing  $E$ ’s decision. In particular, we have  $H, x \models K_A(\langle R \rangle p \vee \langle L \rangle p) \wedge \neg K_A \langle R \rangle p \wedge \neg K_A \langle L \rangle p$ , in words:  $A$  knows he can guarantee  $p$ , but he does not know how! Recall that since a player is supposed to base his decision on the information at hand, we only consider *uniform strategies*, i.e., strategies  $\sigma$  which satisfy the following condition:

$$\forall s, t \in P^{-1}[\{i\}](R_i s \Rightarrow \sigma(s) = \sigma(t)) \quad (10)$$

Without this constraint on strategies, we would have that player  $A$  can enforce the outcome  $p$  in game  $H$  of Figure 9:  $H, \rho \models [l, r] \langle L \cup R \rangle p$ , which is counterintuitive: in order to achieve  $p$ , player  $A$  must play a different move in two situations that he cannot distinguish. Concluding:  $A$  has a strategy to ensure  $p$  in  $H$ , although we cannot expect him to play it, because the strategy is not uniform. But there is more to say about this. Suppose that in game  $H$  player  $E$  makes his move, and *then* we ask ourselves whether  $A$  has a uniform strategy to win. Surprisingly, he has! If  $E$  would play  $l$ , then  $A$  can use the uniform strategy  $\sigma_1$ , with  $\sigma_1(x) = \sigma_1(y) = R$ , and were  $E$  to play  $r$ , player  $A$  can fall back on the uniform strategy  $\sigma_2(x) = \sigma_2(y) = L$ . So, rather than just requiring uniform strategies for players to be used, we need an additional feature to distinguish winning situations from others. The notion that we are after seems to be closely related to the notions of *knowing-de dicto* and *knowing-de re*. The former expresses that player  $i$  knows that he has a winning strategy (if  $Ac_i$  represents  $i$ ’s set of actions, this would mean  $K_i \bigvee_{a \in Ac_i} \langle a \rangle \mathbf{win}_i$ ), whereas the latter expresses he knows how to achieve it:  $\bigvee_{a \in Ac_i} K_i \langle a \rangle \mathbf{win}_i$ . For a further discussion on knowing-*de re* and knowing-*de dicto* in the context of extensive games (applying this to full strategies, rather than actions), see also [39].

The important difference between these two notions of knowledge, and its consequences for a theory of action, was already made in 1990 in [51] (and it goes back to the question what it means that ‘ $A$  knows who  $B$  is’, [36] and the general problem of ‘quantifying in’ into a knowledge formula, [69]). In the context of reasoning about knowledge and action [51] has been very influential, as it is still in the area of decision making in multi-agent systems. In order to cope with examples like ‘in order to open a safe, you have to know

its key', [51] demonstrates the value of a possible world semantics, under the assumptions that terms are *rigid designators*. Although the language used in [51] is that of first order logic, one easily recognises properties like *perfect recall* and *no learning* (see Section 5 and 6.1) in [51]'s *noninformative action*.

[51] argues that the assumption of perfect information has been the prominent one in planning in Artificial Intelligence up to the nineties. What is missing, in the first place, is an analysis of *epistemic pre-conditions* before executing a plan: does the agent have the information necessary to carry out (the next step of) the plan? This question has been taken up by computer scientists by coming up with the notion of *knowledge programs*, (cf. [21, Chapter 7]) in which objective test conditions in a program are replaced by epistemic conditions. Apart from specifying epistemic pre-conditions, [51] argues to include *epistemic post-conditions*, in order to facilitate the agent to reason about how to acquire the information that is lacking to execute a sequential plan successfully. All these notions are crucial in the context of game playing, as well.

Where we thus far 'only' imposed the *S5* principles on imperfect information games, [83] mentions several properties that one could require 'on top' of this, some even related to incomplete information games, like  $\mathbf{turn}_i \rightarrow C\mathbf{turn}_i$  (it is common knowledge whose turn it is) and  $\langle a \rangle \top \rightarrow C\langle a \rangle \top$  (it is common knowledge which moves can be played, in any node).

If information sets were only used to impose players to stick to uniform strategies (10) one only needs to specify knowledge-accessibility for player  $i$  at  $i$ 's decision nodes. But, although this seems indeed to be common practice in game theory, the full machinery of *S5* allows for much finer structure. And there are many more assumptions in game theory that seem dominant (but maybe easily relaxed), like the one that made us draw a *horizontal* line as indistinguishability in Figure 9, suggesting that players know how long the game has been played for. In general, we can require that agents don't know what has happened, or whose turn it is. Rather than systematically describe all the options, in the next section we focus on one specific property, and show how a modal analysis may be of help.

### 6.1 Case study: Perfect Recall

The principle of perfect recall in a game captures that players have some memory about what happened. Formally, in a dynamic logic setting, it is expressed as (see [83] for further details):

$$\begin{aligned} (i) \quad & (\mathbf{turn}_i \wedge K_i[a]\varphi) \rightarrow [a]K_i\varphi & \text{and} \\ (ii) \quad & (\neg\mathbf{turn}_i \wedge K_i[\cup_{b \in B}]\varphi) \rightarrow [\cup_{b \in B}]K_i\varphi \end{aligned} \tag{11}$$

where  $a$  is any action of player  $i$ , and  $B$  is the union of all actions of the other players. In words: if player  $i$  knows that doing  $a$  will lead to  $\varphi$ , then after having done  $a$ , he knows  $\varphi$ . Note that clause (i) of (11) is not realistic for *specific* moves of the opponent: in game  $H$  of Figure 9 for instance, we have  $H, \rho \models K_A[l]\langle R \rangle p$ , but not  $H, \rho \models [l]K_A\langle R \rangle p$ . Also, it assumes that players are aware when others make their move (cf. synchrony in Section 5).

Semantically, the subformula  $K_i[a]\varphi \rightarrow [a]K_i\varphi$  of (11) corresponds to the following (which is even more apparent from the dual of (11), i.e.  $\langle a \rangle \hat{K}_i\varphi \rightarrow \hat{K}_i\langle a \rangle \varphi$ ):



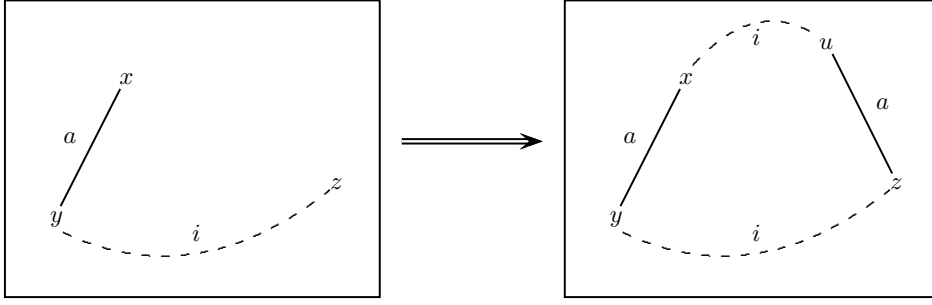


Figure 12. The commuting condition pictured. The arrow stands for implication

$$\forall xyz((R_a xy \ \& \ R_i yz) \rightarrow \exists u(R_i xu \ \& \ R_a uz)) \quad (12)$$

This ‘commutation property’ is also depicted in Figure 12, and guarantees that ignorance, or better indistinguishability, cannot be generated spontaneously: if a player  $i$  cannot distinguish, after doing action  $a$ , between state  $z$  and  $y$ , then both states must be the result of performing  $a$  in indistinguishable states.

For illustrative purposes, let us now look how [14] generalises these notions (over more than just actions), to characterise *von Neumann games*. Let  $s \prec t$  in a game tree denote that there is a path (labelled with choices) from  $s$  to  $t$ . Next, let  $\ell(x)$  denote the number of predecessors of node  $x$  according to  $\prec$ . Then, an extensive game is called von Neumann if

$$\forall xy(R_i xy \ \& \ x, y \in P^{-1}[\{i\}] \Rightarrow \ell(x) = \ell(y)) \quad (13)$$

This implies that in a von Neumann game, a player who has to move knows how many moves have already been played. A game satisfies *Memory of Past Knowledge* (MPK) if

$$\forall xyz((x \prec y \ \& \ R_i yz) \rightarrow \exists u(R_i xu \ \& \ u \prec z)) \quad (14)$$

Note that MPK is weaker than perfect recall in that it abstracts from the specific action taken. In [14] it is shown that perfect recall is equivalent to the conjunction of Memory of Past Knowledge and Memory of Past Actions (in which a player remembers which actions he has taken in the past, not necessarily in which order).

**THEOREM 22.** *Let  $G$  be an extensive form game. Then, the following are equivalent:*

1.  $G$  satisfies MPK
2.  $\forall i, \forall x, y(R_i xy \Rightarrow \ell(x) = \ell(y))$

Note that condition 2 says that it is in fact common knowledge that  $G$  is a von Neumann game. The direction  $(1) \Rightarrow 2$  is proven in [14], the other direction in [9]. The modal characterisation of MPK is given in [14] by using temporal operators. The Past operator  $P$  is for instance interpreted as follows:  $G, s \models P\varphi$  iff for all  $t$  for which  $t \prec s$ , we have  $G, t \models \varphi$ . Similarly, the operator  $\Box$  refers to the future (for more on temporal modal logic, we refer the reader to Chapter 11 of this handbook). Then, MPK is characterised by

$$PK_i \varphi \rightarrow K_i PK_i \varphi \quad (15)$$

As a scheme, (15) is equivalent to  $K_i \varphi \rightarrow \Box K_i PK_i \varphi$ .

## 6.2 Outlook

The upshot of the exercise in the previous section is not so much the specific correspondence results, but rather to demonstrate the elegance and suitability of the modal machinery to reason about notions such as perfect recall. The latter notion also has received considerable attention in the computer science literature, where, in the context of synchronous systems, it is defined as  $K_i \circ \varphi \rightarrow \circ K_i \varphi$ , see Section 5. (Perfect recall is called ‘no-forgetting’ in the seminal paper [29]). The other direction of non-forgetting, i.e.,  $\circ K_i \varphi \rightarrow K_i \circ \varphi$  is called ‘no learning’, and comes with a similar commutation diagram as that in Figure 12.

Unfortunately, such commuting diagrams that enforce regularities on the underlying models have in general an adverse effect on the complexity of checking satisfiability for such logics. Having a grid-like structure in models for a logic enables one to encode Tiling Problems in them, which then can be used to demonstrate that in the worst case satisfiability becomes undecidable (see also Chapters 3 on complexity and 7 on decision problems). In fact, [29] shows that only assuming no-learning (in a context of at least two knowers and allowing common knowledge), the validity problem is highly undecidable. In words of [89]: “Trees are safe” and “Grids are Dangerous”. We refer to that paper for a survey of several (un-)decidability results for temporal epistemic logics. For a restricted set of related results, we refer to Theorem 21 in this Chapter and its succeeding paragraphs.

## 7 DYNAMIC EPISTEMIC LOGIC

The framework for epistemic logic as presented in Section 4 elegantly allows for reasoning about knowledge (and, in particular, higher order knowledge: knowledge about (other’s) knowledge), but as such it does not allow to deal with the *dynamics of epistemics*, in which one can express how certain knowledge changes due to the performance of certain actions, which by itself can be known or not. The notion of run in an interpreted system (Section 5) explicitly allows for such dynamics. In this section, we look at dynamic epistemic logic where the actions themselves are epistemic, like a revision due to a public announcement or a secret message. The famous paper [2] put the *change of information*, or belief revision, as a topic on the philosophical and logical agenda (cf. Chapter 18). This AGM tradition typically studies how a belief or knowledge set should be modified, given some new evidence. Well-studied examples of such modification are *expansion*, *contraction* and *revision*, which are of type  $2^{\mathcal{L}} \times \mathcal{L} \rightarrow 2^{\mathcal{L}}$ , i.e., they transform a belief set  $K$  given new evidence  $\varphi$  into a new belief set  $K'$ , where the belief sets are subsets of the propositional language. The publication of [2] generated a large stream of publications in belief revision, investigating the notion of epistemic entrenchment, the revision of (finite) belief bases, the differences between belief revision and belief updates, and the problem of iterated belief change (for more on belief revision, refer to Chapter 18 in this handbook).

However, in all these approaches the dynamics are studied on a level *above the informational level*: the operators for modification are not part of the object language, and they are defined on (sets of) propositional formulas in  $\mathcal{L}$ . Hence, it is impossible to reason about change of agents’ knowledge and ignorance within the framework, let alone about the change of other agents’ information. This section describes approaches where the changing epistemic attitudes find their way into the object language.

The notion of a run in an interpreted system, together with the availability of temporal operators in the language (Section 5) facilitates reasoning about the dynamics of an agent's knowledge. A run is typically generated by a protocol, which usually represents a standard computer program. The pioneering work in [51] also studies the relation between actions and knowledge: there the emphasis is on epistemic preconditions that are needed to perform certain actions in the world, such as knowing a key-combination in order to open a safe. From the point of view of expressivity, one can say that the work on interpreted systems enables one to reason about the (change of) knowledge over time, and adding actions to the language, one can also reason about the change of knowledge brought about by performing certain plans. This enables one to express properties like *perfect recall* and *no learning* discussed in previous sections.

This section sketches approaches that not only “dynamise the epistemics”, but also “epistemise the dynamics”: the actions that (groups of) agents perform are epistemic actions. Different agents may have different information about which action is taking place, including higher-order information. Perfect recall would then rather look like  $K_i[\alpha]\varphi \rightarrow [K_i\alpha]K_i\varphi$ : “if player  $i$  knows that when  $j$  chooses ‘right’ this offers  $i$  a possible win, only after  $i$  knows that  $j$  does move ‘right’,  $i$  is aware of his profitable situation”! The rather recent tradition often referred to as *Dynamic Epistemic Logic*, treats all of knowledge, higher-order knowledge, and its dynamics on the same level. Following a contribution of 1997 [23], a stream of publications appeared around the year 2000 ([48, 98, 84]) and a general theory only now and partially emerges. In retrospect, it appeared that an original contribution of [68] from 1989 was an unnoticed ancestor of this stream of publications. This section is too short to discuss all those approaches, we will, for homogeneity, mainly follow [102, 101], and [7]. We start by considering a special case of updates.

### 7.1 Public Announcements

Public announcements are a simple and straightforward, but still interesting epistemic action: the idea behind a public announcement of  $\varphi$  is that all players are updated on  $\varphi$ , and they all know this, and they all know that they know this, etc. Given a group of players  $N$ , the language  $\mathcal{L}_N^{pa}$  for public announcements adds a modality  $[\chi]\psi$  on top of the epistemic language with operators  $K_i (i \in N)$ . If the common knowledge operator  $C$  is also allowed, we refer to the language as  $\mathcal{L}_N^{pa}(C)$ . The interpretation of  $[\chi]\psi$  reads: “after truthful public announcement of  $\chi$ , it holds that  $\psi$ ”. Note that both  $\chi$  and  $\psi$  are typical members of  $\mathcal{L}_N^{pa}$  or  $\mathcal{L}_N^{pa}(C)$ : announcements can be nested.

The semantics of  $[\chi]\psi$  is rather straightforward: it is true in  $(M, s)$  if, given that  $\chi$  is true in  $(M, s)$ ,  $\psi$  is true in  $s$  if we ‘throw away’ all the states in which  $\chi$  is false. To achieve this, we define  $M|_\chi$  as that submodel of  $M$  that consists of all points in which  $\chi$  is true. More formally, given  $M = \langle W, R_1, R_2, \dots, R_m, \pi \rangle$ , the model  $M|_\chi = \langle W', R'_1, R'_2, \dots, R'_m, \pi' \rangle$  has as its domain all the  $\chi$  states:  $W' = \{w \in W \mid (M, w) \models \chi\}$ , and the primed relations and valuation in  $M'$  are the restrictions of the corresponding relations and valuation in  $M$  to  $W'$ . Then, we define

$$M, s \models [\chi]\psi \text{ iff } (M, s \models \chi \Rightarrow M_\chi, s \models \psi). \quad (16)$$

EXAMPLE 23. (Example 18 ctd.) In the miniature card game hexa (Example 18), suppose that in (hexa, rwb), player 1 publicly announces that he does not possess card  $w$ , i.e.,

$\varphi = \neg w_1$ . Then, the resulting model is  $\text{hexa}|_{\varphi} = \text{hexa}_1$  of Figure 10: all the deals in which 1 *does* have the white card are removed. Note that we have:  $\text{hexa}, rwb \models [\neg w_1]K_3r_1$ , and even  $\text{hexa}, rwb \models [\neg w_1]K_3(r_1 \wedge w_2 \wedge b_3)$ , saying that after 1's announcement, player 3 knows the exact deal. Note that this is not true for player 2:  $\text{hexa}, rwb \models [\neg w_1]\neg K_2r_1$ , since, after the announcement  $\neg w_1$ , player 2 still considers it possible that 1 has  $b$ . Player 2 knew already the truth of the announcement. Still, he 'learns' from it:

$$\text{hexa}, rwb \models K_2\neg K_3(r_1 \wedge w_2 \wedge b_3) \wedge [\neg w_1]K_2(K_3(r_1 \wedge w_2 \wedge b_3) \vee K_3(b_1 \wedge w_2 \wedge r_3))$$

This expresses that initially, in  $rwb$ , 2 knows that 3 does not know the current deal (described by  $r_1 \wedge w_2 \wedge b_3$ ), but after 1's announcement  $\neg w_1$ , 2 knows that 3 knows the deal. Note that 1 does not learn the same as 2: player 1 cannot be sure that 3 learns the deal from the announcement  $\neg w_1$ , since, according to 1, it might be the case that 3 holds  $w$ , in which case 3 would not learn the deal from the announcement.

Public announcements can be made iteratively: the model  $\text{hexa}_2$  is obtained from  $\text{hexa}_1$ , by letting 3 make the public announcement "I know the deal!". More formally, let  $\text{knowsdeal}(i)$  be  $\bigvee_{c,d,e \in \{r,w,b\}} K_i(c_1 \wedge d_2 \wedge e_3)$ . Then, 1 learns the deal after 3 announces that he learned, but 2 does not (let  $\delta$  be the actual deal  $(r_1 \wedge w_2 \wedge b_3)$ ):

$$\text{hexa}, rwb \models [\neg w_1][\text{knowsdeal}(3)](K_1\delta \wedge \neg K_2\delta)$$

This can be formally verified by inspecting Figure 10, but is also intuitively correct: if 1, holding  $r$ , announces that he does not possess  $w$ , then he knows that this is either informative for 2 (in case 3 has  $w$ , i.e., in  $rbw$ ) or for 3 (in  $rwb$ ). Since 3 subsequently announces he learned the deal, 1 finds out the real situation is  $rwb$ . Similarly, 2 does not learn the deal from this "dialogue", he conceives it still possible that the real deal is  $bwr$ . However, as we saw above, 2 still learns something (i.e., about the knowledge of others: after the first announcement 2 learns that 3 knows the deal and after the second 2 learns that also 1 knows the deal).  $\dashv$

As for an axiomatisation of public announcements, the logic  $\mathbf{S5}_N^{pa}$  is obtained by adding the left-hand side of Table 4 to  $\mathbf{S5}_m$ . The logic  $\mathbf{S5}_N^{pa}(C)$ , which also incorporates common knowledge, is axiomatised by the union of Tables 2 and 4.

axioms and rules for $\mathbf{S5}_N^{pa}$		additional rule for $\mathbf{S5}_N^{pa}(C)$	
A11	$([\chi]\varphi \wedge [\chi](\varphi \rightarrow \psi)) \rightarrow [\chi]\psi$	R5	From $\vdash \varphi \rightarrow [\chi]\psi$ and $\vdash \varphi \wedge \chi \rightarrow E\varphi$ infer $\vdash \varphi \rightarrow [\chi]C\psi$
A12	$[\chi]p \leftrightarrow (\chi \rightarrow p)$		
A13	$[\chi]\neg\psi \leftrightarrow (\chi \rightarrow \neg[\chi]\psi)$		
A14	$[\chi]K_i\psi \leftrightarrow (\chi \rightarrow K_i[\chi]\psi)$		
R4	$\vdash \varphi \Rightarrow \vdash [\chi]\varphi$		

Table 4. Public announcements without (left) and with common knowledge

Axiom A11 and rule R4 characterise  $[\chi]$  as a normal modal operator. The other axioms have the general form  $[\chi]\varphi \leftrightarrow (\chi \rightarrow \varphi')$ ; the second appearance of  $\chi$  indicates that only behaviour of *successful updates* is specified. Note that the general form is equivalent to  $(\chi \wedge ([\chi]\varphi \leftrightarrow \varphi')) \vee (\neg\chi \wedge [\chi]\varphi)$ , which, by the fact that  $\vdash \neg\chi \rightarrow [\chi]\varphi$ , is equivalent to  $(\chi \wedge ([\chi]\varphi \leftrightarrow \varphi')) \vee \neg\chi$ .

Keeping this in mind, axiom A12 (also called *atomic persistence*) assures that atomic facts (and, hence, objective properties, not involving any knowledge) are not affected by public announcement: they do not change the world. According to A13, public announcements are partial functions: under the condition that the announced formula is true, its announcement induces a unique outcome. Finally, axiom A14 relates individual knowledge to public announcements: from right to left it is a variant of the earlier mentioned *perfect recall*, the other direction is a conditionalised *no learning* property.

The straightforward generalisation of A14 to a logic with common knowledge would read  $[\chi]C\psi \leftrightarrow (\chi \rightarrow C[\chi]\psi)$ . However, such a principle is not valid. Consider the models  $M$  and  $M'$  of Figure 13 (taken from [101]). First of all, let 10 denote a world in which  $p$  is true, and  $q$  is false, and similarly for 11 and 01. Then, in model  $M$  of Figure 13, we have  $M, 11 \models [p]Cq$ , since in the updated model  $M' = M|_p$ , we have  $M', 11 \models Cq$  (in  $M'$ , world 11 is only  $a$ - and  $b$ - accessible to itself). At the same time, we have  $M, 11 \not\models (p \rightarrow C[p]q)$ , in particular  $M, 11 \not\models C[p]q$ . This is so since in 11 there is a world  $R_C$ -accessible (to wit, 10), in which a public announcement of  $p$  would lead us to the disconnected part 10 of  $M'$ , where we have  $M', 10 \not\models q$ , so that  $M, 10 \not\models [p]q$ , which justifies the claim  $M, 11 \not\models C[p]q$ .

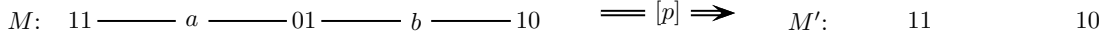


Figure 13. A counter model for  $[p]Cq \leftrightarrow (p \rightarrow C[p]q)$

In order to obtain common knowledge through a public announcement, rule R5 from Table 4 must be used. The soundness of this rule is typically proven using induction on the  $R_C$ -path to  $\psi$  in an updated model  $M|_\chi$ .

**THEOREM 24** ([68, 8]). *The logics  $\mathbf{S5}_N^{pa}$  (without common knowledge) and  $\mathbf{S5}_N^{pa}(C)$  (with common knowledge) as defined in Table 4 are sound and complete with respect to the semantics with key condition (16) on top of the epistemic semantics as given in Section 4.*

In fact, dynamic epistemic logic  $\mathbf{S5}_N^{pa}$  can be reduced to its static counterpart  $\mathbf{S5}_{|N|}$ , by employing the following translation  $T$ :

$$\begin{aligned} T([\chi]p) &= T(\chi) \rightarrow p \\ T([\chi](\varphi \wedge \psi)) &= T([\chi]\varphi) \wedge T([\chi]\psi) \\ T([\chi]\neg\psi) &= T(\chi) \rightarrow \neg T([\chi]\psi) \\ T([\chi]K_i\psi) &= T(\chi) \rightarrow K_i T([\chi]\psi) \end{aligned}$$

The equivalence between  $\varphi$  and  $T(\varphi)$  follows immediately from the axioms of Table 4, and it is also easy to see that  $T(\varphi)$  has no occurrences of the  $[\chi]$  operator: they are all replaced by implications  $\chi \rightarrow p$  for certain atoms  $p$ . This feature can be used to obtain completeness of  $\mathbf{S5}_N^{pa}$  ([68]), but for the case of  $\mathbf{S5}_N^{pa}(C)$ , the completeness proof is much more involved ([8]).

### Unsuccessful Updates

The intuition of a public announcement  $[\chi]$  is that it produces common knowledge of the announced fact  $\chi$ . Remarkably enough, it is not always the case that  $[\chi]C\chi$ . As an example, take the model  $M$  of Figure 13, where in 11 the atom  $p$  holds, but  $a$  is ignorant about it. The public announcement of this very fact (i.e.,  $p \wedge \neg K_a p$ , which could be uttered by player  $b$ , since he knows it) however, leaves  $a$  with a difficult, if not impossible task to update his knowledge; it is hard to see how to simultaneously incorporate  $p$  and  $\neg K_a p$  into his knowledge.

**DEFINITION 25** ((Un-)successful Formulas and Updates). A formula  $\chi$  is *successful* if  $\models [\chi]\chi$ . Otherwise, it is *unsuccessful*. Moreover,  $\chi$  is a *successful update* in  $M, s$  if  $M, s \models (\chi \wedge [\chi]\chi)$ , it is an *unsuccessful update* if  $M, s \models (\chi \wedge [\chi]\neg\chi)$ . An update with  $\chi$  is *publicly successful* in  $M, s$  if  $M, s \models \chi \wedge [\chi]C\chi$ .  $\dashv$

Which formulas are unsuccessful, and which are successful? This question was raised in [82], and some first answers are given in [86] and [99]. Typically, only formulas involving ignorance can be unsuccessful. Hence, propositional formulas, involving no epistemic operators are always successful. Secondly common knowledge formulas are successful, by merits of the validity  $\vdash [C\varphi]C\varphi$ . The paper [99] identifies a fragment of the language,  $\mathcal{L}_N^{u0}$  that is preserved under ‘deleting states’:

$$\varphi \in \mathcal{L}_N^{u0} ::= p \mid \neg p \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid K_i \varphi \mid C\varphi \mid [\neg\varphi]\psi$$

The fragment  $\mathcal{L}_N^{u0}$  is preserved under submodels, from which it follows that for any  $\varphi \in \mathcal{L}_N^{u0}$  and any  $\psi$ ,  $\vdash \varphi \rightarrow [\psi]\varphi$ . As a consequence, the language  $\mathcal{L}_N^{u0}$  is successful. After presenting these partial results, and before giving an example of (un-)successful updates, we mention the following fact about updates:

**THEOREM 26** ([86]). *In every model, every public announcement is equivalent to a successful one.*

**EXAMPLE 27.** (Example 19 ctd., [99]) Consider model **twomud** from Figure 11. Let us abbreviate  $(m_a \vee m_b \vee m_c)$  to **muddy**. Then model  $(\text{twomud}_1, 110)$  is the model that one obtains when publicly announcing **muddy** in  $(\text{twomud}, 110)$ , i.e., after (5) is announced. One easily checks that **muddy** is a publicly successful update in this state:  $\text{twomud}, 110 \models [\text{muddy}]C\text{muddy}$  (note that, since **muddy** is a member of the submodel-preserving language  $\mathcal{L}_N^{u0}$ , it is even a successful *formula*).

Note that  $\text{twomud}, 100 \models [\text{muddy}]K_a m_a$ : if  $a$  is the only muddy child, he knows about his muddiness after the announcement (5) that there is at least one muddy child. Let  $\text{knowmuddy} = \bigvee_{i \in \{a, b, c\}} (K_i m_i \vee K_i \neg m_i)$  (at least one child knows about its muddy state). Now, although we have  $\text{twomud}_1, 110 \models \neg \text{knowmuddy}$ , we have  $\text{twomud}_1, 110 \models [\neg \text{knowmuddy}]\text{knowmuddy}$ . In other words, when the father makes his announcement (5) for the second time, we interpret this as an announcement of  $\neg \text{knowmuddy}$  (since father makes his remark for the second time, this is a public announcement that no child stepped forward after the first utterance of (5), or, in other words, no child knows yet about its muddiness). Since **knowmuddy** is true in  $\text{twomud}_1$  in the states 001, 010 and 100, these states are removed after the announcement  $\neg \text{knowmuddy}$ , giving us model  $\text{twomud}_2$  of Figure 11.

To further explain the story, one easily verifies  $\text{twomud}_2, 110 \models C\text{atleasttwomuddy}$ , with **atleasttwomuddy** having its obvious interpretation  $\bigvee_{i, j \in \{a, b, c\}}^{i \neq j} (m_i \wedge m_j)$ . But if this

is common knowledge in  $\text{twomud}_2, 110$ , in particular children  $a$  and  $b$  know this: since they only see one other muddy child, they conclude that they are muddy themselves and hence step forward. We have seen that  $\text{twomud}, 110 \models \text{muddy} \wedge [\text{muddy}]C\text{muddy}$ , and also that  $\text{twomud}, 110 \models [\text{muddy}] [\neg\text{knowmuddy}]C\text{knowmuddy}$ . In other words,  $\neg\text{knowmuddy}$  is an unsuccessful update in  $\text{twomud}_1, 110$ . Note that this is indeed a local notion: the same announcement  $\neg\text{knowmuddy}$  would have been successful in  $(\text{twomud}_1, 111)$ .  $\neg$

## 7.2 General Updates

Public announcements play an important role in games: putting a card on the table, rolling a die, and moving a pawn on the chess board can all be considered as examples. However, in many situations much more subtle communication takes place than a public announcement. Consider the card game hexa in which player 1 shows player 2 his card. Obviously, this is informative for 2: he even learns the actual deal. But, although 3 does not see 1's card, he certainly obtains new information, viz. that 2 learns the deal. And 1 and 2 also get to know that 3 learns this!

The following – possibly simplest – example in the setting of multi-agent systems (two agents or players, one atom) attempts to demonstrate that the notions of higher-order information and epistemic actions are indeed non-trivial and may be subtle.

*Anne and Bert are in a bar, sitting at a table. A messenger comes in and delivers a letter that is addressed to Anne. The letter contains either an invitation for a night out in Amsterdam, or an obligation to give a lecture instead. Anne and Bert commonly know that these are the only alternatives.*

This situation can be modelled as follows: There is one atom  $p$ , describing ‘the letter invites Anne for a night out in Amsterdam’, so that  $\neg p$  stands for her lecture obligation. There are two agents 1 (Anne) and 2 (Bert). Whatever happens in each of the following action scenarios, is publicly known (to Anne and Bert). Also, assume that in fact  $p$  is true.

SCENARIO 28 (tell). Anne reads the letter aloud.

SCENARIO 29 (read). Bert sees that Anne reads the letter.

SCENARIO 30 (mayread). Bert orders a drink at the bar so that Anne may have read the letter.

SCENARIO 31 (bothmayread). Bert orders a drink at the bar while Anne goes to the bathroom. Both may have read the letter.

After execution of the first scenario (which is in fact a public announcement), it is common knowledge that  $p$ : in the resulting epistemic state  $Cp$  holds. This is not the case in the second scenario, but still, some common knowledge is obtained there:  $C(K_1p \vee K_1\neg p)$ : it is commonly known that Anne knows the content of the letter, irrespective of it being  $p$  or  $\neg p$ . Does this higher-order information change in Scenario 30? Yes, in this case Bert does not even know if Anne knows  $p$  or knows  $\neg p$ :  $\neg K_2(K_1p \vee K_1\neg p)$ . In Scenario 31 something similar is happening, that may best be described by saying that the agents concurrently learn that the other may have learnt  $p$  or  $\neg p$ . Note that in this case, both agents may have learnt  $p$ , so that  $p$  is generally known:  $E_{12}p$ , but they are in that case unaware of each other's knowledge,  $\neg C_{12}p$ , and *that* is commonly known.

Scenarios 30 and 31 are interesting, since semantically, they indicate that one cannot simply rely on the strategy of deleting states. The scenarios not only provide the agents with certainty, but also some doubts arise. After Scenario 30 for example, Bert must find an alternative state possible, in which Anne knows the contents of the letter, but also one in which Anne does not know. This is in a nutshell the main challenge in the semantics of these general updates.

### Language

To a standard multi-agent epistemic language with common knowledge for a set  $N$  of agents and a set  $P$  of atoms, we add dynamic modal operators for programs that are called knowledge actions or just actions. Actions may change the knowledge of the agents involved. The formulas  $\mathcal{L}_N$ , the actions  $\mathcal{L}_N^{\text{act}}$ , and the group  $gr$  of an action are defined by simultaneous induction:

DEFINITION 32 (Formulas and actions). The formulas  $\mathcal{L}_N^{gu}(P)$  are defined by

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid K_n\varphi \mid C_B\varphi \mid [\alpha]\psi$$

where  $p \in P$ ,  $n \in N$ ,  $B \subseteq N$ ,  $\alpha \in \mathcal{L}_N^{\text{act}}(P)$ , and  $\psi \in \mathcal{L}_N^{gu}(P)$ . The actions  $\mathcal{L}_N^{\text{act}}(P)$  are defined by

$$\alpha ::= ?\varphi \mid L_B\beta \mid (\alpha ! \alpha) \mid (\alpha \dot{\mid} \alpha) \mid (\alpha ; \beta') \mid (\alpha \cup \alpha) \mid (\alpha \cap \alpha)$$

where  $\varphi \in \mathcal{L}_N^{gu}(P)$ ,  $B \subseteq N$ ,  $\beta \in \mathcal{L}_N^{\text{act}}(P)$ , and  $\beta' \in \mathcal{L}_N^{\text{act}}(P)$ , and where the group  $gr(\alpha)$  of an action  $\alpha \in \mathcal{L}_N^{\text{act}}(P)$  is defined as:  $gr(? \varphi) := \emptyset$ ,  $gr(L_B\alpha) := B$ , and  $gr(\alpha \bullet \alpha') := gr(\alpha) \cap gr(\alpha')$  for  $\bullet = !, \cap, \cup, ;$ .

The program constructor  $L_B$  is called *learning*. Action  $? \varphi$  is a *test*,  $(\alpha ; \alpha')$  is *sequential execution*,  $(\alpha \cup \alpha')$  is *nondeterministic choice*,  $(\alpha ! \alpha')$  is called *(left) local choice* and  $(\alpha \dot{\mid} \alpha')$  is called *(right) local choice*, and  $(\alpha \cap \alpha')$  is *concurrent execution*. The construct  $L_B ? \varphi$  is pronounced as ‘ $B$  learn that  $\varphi$ ’. Local choice  $\alpha ! \alpha'$  may, somewhat inaccurately, be seen as ‘from  $\alpha$  and  $\alpha'$ , choose the first.’ Local choice  $\alpha \dot{\mid} \alpha'$  may be seen as ‘from  $\alpha$  and  $\alpha'$ , choose the second.’ The interpretation of local choice ‘!’ and ‘ $\dot{\mid}$ ’ depends on the context of learning that binds it: in  $L_B(\alpha ! \alpha')$ , everybody in  $B$  but not in learning operators occurring in  $\alpha, \alpha'$ , is unaware of the specific choice for  $\alpha$ . That choice is therefore ‘local’.

EXAMPLE 33. The description in  $\mathcal{L}_{12}^{\text{act}}(\{p\})$  of the actions in the introduction are:

$$\begin{array}{ll} \text{tell} & L_{12}?p \cup L_{12}\neg p \\ \text{read} & L_{12}(L_1?p \cup L_1\neg p) \\ \text{mayread} & L_{12}(L_1?p \cup L_1\neg p \cup ?\top) \\ \text{bothmayread} & L_{12}((L_1?p \cap L_2?p) \cup (L_1\neg p \cap L_2\neg p) \\ & \cup L_1?p \cup L_1\neg p \cup L_2?p \cup L_2\neg p \cup ?\top) \end{array}$$

For example, the description of *read* (Anne reads the letter) reads as follows: ‘Anne and Bert learn that either Anne learns that she is invited for a night out in Amsterdam or that Anne learns that she has to give a lecture instead.’

By replacing all occurrences of ‘!’ and ‘ $\dot{\mid}$ ’ in an action  $\alpha$  by ‘ $\cup$ ’, except when under the scope of  $?$ , we get the *type*  $t(\alpha)$  of that action. By replacing all occurrences of ‘ $\cup$ ’



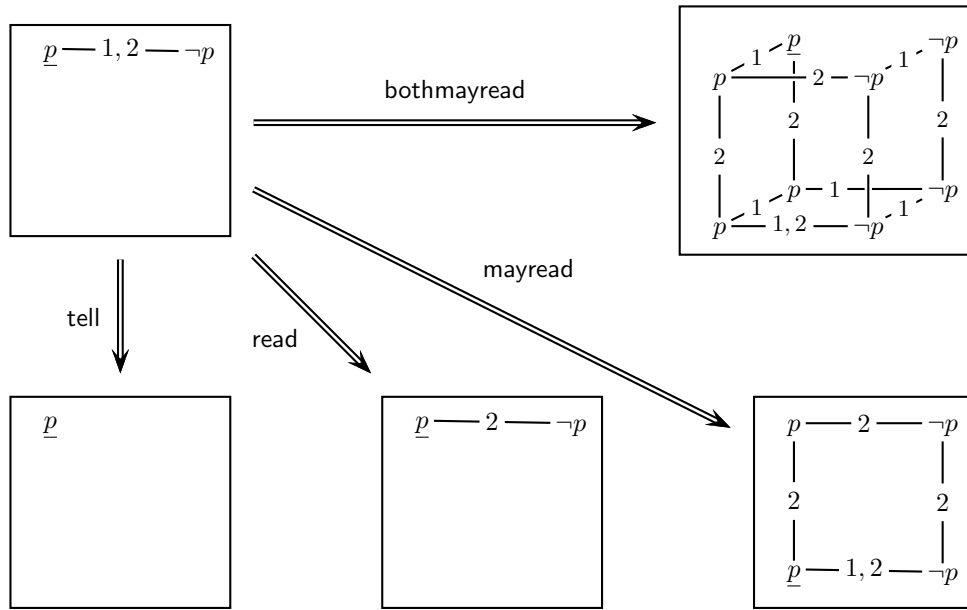


Figure 14. Epistemic states resulting from the execution of actions described in the four action scenarios. The top left figure represents  $(Let, u)$ , in which it is common knowledge that both 1 and 2 are ignorant about  $p$ . For **mayread** and **bothmayread** only one of more executions is shown: namely the one in which actually nothing happens, and the one in which both 1 and 2 find out that  $p$ , respectively.

in an action  $\alpha$  by either ‘!’ or ‘;’, except when under the scope of  $?$ , we get the set of *instances*  $I(\alpha)$  of that action. Informally we write:  $I(\alpha) := \{\alpha[\cup/!, ;]\}$ . If  $t(\alpha) = t(\beta)$  we say that  $\alpha$  and  $\beta$  are the same type of action. Furthermore, if  $\alpha$  and  $\beta$  are identical modulo swapping of occurrences of ‘!’ for ‘;’ or vice versa, we say that  $\alpha, \beta$  are *comparable* actions. The idea here is that in the scope of an  $L_B$  operator, the agents of  $B$  know which action is executed, but the agents not in  $B$  consider all actions of the same type possible. Instead of  $\alpha ! \alpha'$  we also write  $!\alpha \cup \alpha'$ . This expresses more clearly that given choice between  $\alpha$  and  $\alpha'$ , the agents involved in those actions choose  $\alpha$ , whereas that choice remains invisible to the agents that learn about these alternatives but are not involved. Similarly, instead of  $\alpha ; \alpha'$  we write  $\alpha \cup !\alpha'$ .

EXAMPLE 34. The action **read** where Bert sees that Anne reads the letter is different from the instance of that action where *Anne is actually invited for a night out* and Bert sees that Anne reads the letter. The latter is described as  $L_{12}(!L_1?p \cup L_1?\neg p)$ : of the two alternatives  $L_1?p$  and  $L_1?\neg p$ , the first is chosen, but agent 2 is unaware of that choice. The description **read** is its *type*. The other instance of **read** is  $L_{12}(L_1?p \cup !L_1?\neg p)$ . Actions  $L_{12}(!L_1?p \cup L_1?\neg p)$  and  $L_{12}(L_1?p \cup !L_1?\neg p)$  are *comparable* to each other.  $\dashv$

### Semantics and Axioms

Concerning the semantics of  $\mathcal{L}_N^{gu}(P)$  (on epistemic models), we refer to Chapter 12 for the treatment of the dynamic operators, and focus here on the learning operator. Although our object language is that of [102], we focus on the semantics as explained in [7], which we coin *action model semantics*. The appealing idea in the action model semantics is that both the uncertainty about the state of the world, and that of the action taking place, are represented in two independent Kripke models. The result of performing an epistemic action in an epistemic state is then computed as a ‘cross-product’. We give some more explanation by way of an example: see also Figure 15.

Model  $N$  in this figure is the model *Let*, but now we have given names  $s$  and  $t$  to the states in it. The triangular shaped model  $N$  is the *action model* that represents the knowledge and ignorance when the instance  $L_{12}(L_1?p \cup L_1?\neg p \cup !\top)$  of **mayread** is carried out. The points **a**, **b**, **c** of the model  $N$  are also called actions, and the formulas accompanying the name of the actions are called *pre-conditions*: the condition that has to be fulfilled in order for the action to take place. Since we are in the realm of *truthful information transfer*, in order to perform an action that reveals  $p$ , the pre-condition **p** must be satisfied, and we write  $\text{pre}(\mathbf{b}) = \mathbf{p}$ . For the case of nothing happening, only the precondition  $\top$  need be true. Summarising, action **b** represents the action that agent 1 reads  $p$  in the letter, action **c** is the action when  $\neg p$  is read, and **a** is for nothing happening. As with ‘static’ epistemic models, we omit reflexive arrows, so that  $N$  indeed represents that  $p$  or  $\neg p$  is learned by 1, or that nothing happens: moreover, it is commonly known between 1 and 2 that 1 knows which action takes place, while for 2 they all look the same.

Now let  $M, w = \langle W, R_1, R_2, \dots, R_m, \pi \rangle, w$  be an epistemic static state, and  $M, w$  an action in a finite action model. We want to describe what  $M, w \oplus M, w = \langle W', R'_1, R'_2, \dots, R'_m, \pi' \rangle, w'$  looks like — the result of ‘performing’ the action represented by  $M, w$  in  $M, w$ . Every action from  $M, w$  that is executable in any state  $v \in W$  gives rise to a new state in  $W'$ : we let  $W' = \{(v, v) \mid v \in W, M, v \models \text{pre}(v)\}$ . Since epistemic actions do not change any objective fact in the world, we stipulate  $\pi'(v, v) = \pi(v)$ . Finally, when are

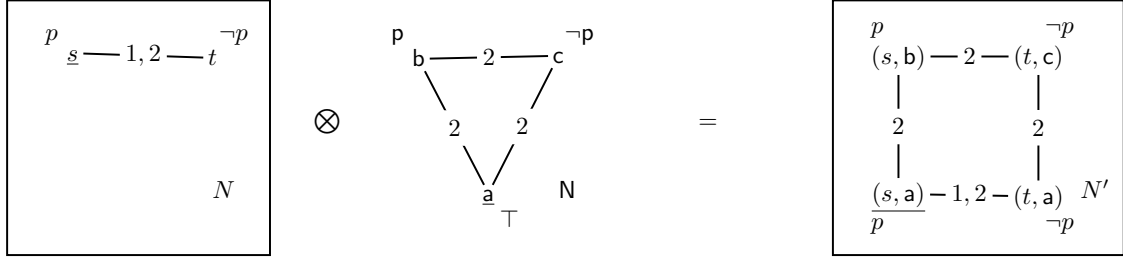


Figure 15. Multiplying the epistemic state  $N$  with the action model  $(N, a)$  representing the action instance  $L_{12}(L_1?p \cup L_1?\neg p \cup !\top)$  of **mayread**

two states  $(v, v)$  and  $(u, u)$  indistinguishable for agent  $i$ ? Well, he should be both unable to distinguish the originating states  $(R_i uv)$ , and unable to know what is happening  $(R_i uv)$ . Finally, the new state  $w'$  is of course  $(w, w)$ . Note that this construction indeed gives  $N, s \oplus N, a = N', (s, a)$ , in our example of Figure 15. Finally, let the action  $\alpha$  be represented by the action model state  $M, w$ . Then the truth definition under the action model semantics reads that  $M, w \models [\alpha]\varphi$  iff  $M, w \models \text{pre}(w)$  implies  $(M, w) \oplus (M, w) \models \varphi$ . In our example:  $N, s \models [L_{12}(L_1?p \cup L_1?\neg p \cup !\top)]\varphi$  iff  $N', (s, a) \models \varphi$ .

Note that the accessibility relation in the resulting model is defined as

$$R_i(u, u)(v, v) \Leftrightarrow R_i uv \ \& \ R_i uv \quad (17)$$

As a consequence, perfect recall does not hold for  $[\alpha]$ : Let  $\alpha$  be  $L_{12}(L_1?p \cup L_1?\neg p \cup !\top)$ . We then have  $N, s \models K_2[\alpha]\neg(K_1 p \vee K_1 \neg p)$  (2 knows that if nothing happens, 1 will not find out whether  $p$ ), but not  $N, s \models [\alpha]K_2\neg(K_1 p \vee K_1 \neg p)$ . We *do* have in general the following weaker form of perfect recall, however. Let  $M, w$  be a static epistemic state, and  $\alpha$  an action, represented by some action state  $M, w$ . Let  $A$  be the set of actions that agent  $i$  cannot distinguish from  $M, w$ . Then we have

$$M, w \models \bigwedge_{\beta \in A} K_i[\beta]\varphi \rightarrow [\alpha]K_i\varphi \quad (18)$$

In words, in order for agent  $i$  to ‘remember’ what holds after performance of an action  $\alpha$ , he should already now in advance that it will hold after *every epistemically possible execution* of that action. In the card example of **hexa**: if player 1 shows player 2 his card (which is red), then player 3 ‘only’ knows that 2 learned that 1 holds red or white, because he cannot distinguish the action in which 1 shows red from the action in which 1 shows white. The perfect recall version (18) is a consequence of the ‘ $\Rightarrow$ ’-direction of (17), the other direction gives the following generalized and conditionalised version of ‘no learning’:  $[\alpha]K_i\varphi \rightarrow (\text{pre}(\alpha) \rightarrow \bigwedge_{\beta \in A} K_i[\beta]\varphi)$ . This implies that, everything that is known after a *specific* execution of an action, was already known to hold after any *indistinguishable* execution of that action.<sup>1</sup>

Concerning axiomatisations for dynamic epistemic logic, we provide some axioms in Table 5. These have to be added on top of **S5<sub>m</sub>** and the usual axioms for the dynamic operators. Let us call the resulting system **DEL(S5)<sub>m</sub>**

<sup>1</sup>We have oversimplified the treatment of [7], in particular we have not discussed what it means that an action  $\alpha$  is represented by an action state  $M, w$ . For further discussion, see [7], or [101], where both semantics discussed here are dealt with.

some axioms of $\mathbf{DEL}(\mathbf{S5})_m$		some rules of $\mathbf{DEL}(\mathbf{S5})_m$	
A15	$\langle L_B \alpha \rangle \top \leftrightarrow \mathbf{pre}(L_B \alpha)$	R6	$\vdash \varphi \rightarrow \psi \Rightarrow \vdash [\alpha] \varphi \rightarrow [\alpha] \psi$
A16	$[\alpha ! \alpha'] \varphi \leftrightarrow [\alpha] \varphi$	R7	If: for all $\beta$ with $\alpha \sim_B \beta$
A17	$[\alpha] p \leftrightarrow (\mathbf{pre}(\alpha) \rightarrow p)$		there is a $\chi_\beta$ such that
A18	$[\alpha] \varphi \leftrightarrow \bigwedge_{\beta \in I(\alpha)} [\beta] \varphi$		(1) $\vdash \chi_\beta \rightarrow [\beta] \varphi$ , and
A19	$\bigwedge_{\beta \sim_i \alpha} K_i [\beta] \varphi \rightarrow [\alpha] K_i \varphi$		(2) $\beta \sim_n \alpha'$ implies
A20	$[\alpha] K_i \varphi \rightarrow$ $(\mathbf{pre}(\alpha) \rightarrow \bigwedge_{\beta \sim_i \alpha} K_i [\beta] \varphi)$		$\vdash (\chi_\beta \wedge \mathbf{pre}(\beta)) \rightarrow E_B \chi_{\alpha'}$
			Then: $\chi_\alpha \rightarrow [\alpha] C_B \varphi$

Table 5. Epistemic axioms and rules:  $i \leq m$ 

We elaborate shortly on the axioms: for further details we refer to [7, 102]. For any action  $\alpha$ , a formula  $\mathbf{pre}(\alpha)$  can be defined which is true exactly when  $\alpha$  can be executed. Axiom A15, also called ‘Learning’, says that this is well-defined for learning actions, and A17 (‘Atomic permanence’) says that, like public announcement, general updates do not change the objective statements, given that the update is executable. Axiom A16 (‘Local choice’) determines the meaning of  $!$ . Axiom A18 (‘Action instances’) formalises that the effect of an action is a combined effect of all its instances. One might have expected a distribution axiom for  $[\alpha]$ , but this is not sound. Such an axiom is unsound in any dynamic logic with concurrency (see Chapter 12), for the same reason: the interpretation of actions are relations between epistemic states and sets of epistemic states. The modality  $[\alpha]$  corresponds to a  $\forall\exists$  quantifier for which distribution does not hold. We do have a weaker form of distribution in the form of the action facilitation rule R7. This is all we need in the completeness proof.

Note that axioms A19 and A20 are variants of the earlier discussed principles of ‘recall’ and ‘no learning’, respectively. They give what one could call a ‘compositional analysis’ of pre- and post-conditions of epistemic events. Axiom A19 expresses that, in order to know, after  $\alpha$  has happened, that  $\varphi$ , one has to know in advance that, no matter which action happens that looks like  $\alpha$ , property  $\varphi$  will result. In a simple card game example: if I know after you show Ann a card that she has to know the full deal of cards among us, I should know in advance that Ann knows the deal after every card that I imagine you showing Ann. Often a contrapositive of such an axiom is appealing, in this case it reads  $\langle \alpha \rangle \hat{K}_i \varphi \rightarrow \bigvee_{\beta \sim_i \alpha} \hat{K}_i \langle \beta \rangle \varphi$ : if there is an execution of  $\alpha$  after which I still consider  $\varphi$  a possibility, then for some action that looks the same to me as  $\alpha$ , I imagine it possible that there is an execution which leads to  $\varphi$ . If you only have two cards, and you show them in a sequence to her ( $\alpha$ ), but I only see you show her a card twice, I still think it conceivable that she does not know both of your cards ( $\varphi$ ), since I take into the consideration that you showed her twice the same card ( $\beta$ ), after which she would not know both your cards. We leave elaboration about A20 to the reader.

Even the soundness of inference rule R7 is not easy to grasp. It mimics an induction rule where the  $\chi$  formulas are used as induction hypotheses. It uses a notion of indistinguishability of actions:  $\alpha \sim_B \beta$  means that group  $B$  cannot distinguish the execution of  $\alpha$  from that of  $\beta$ . Also, the completeness proof of this logic is not easy, since in the canonical model one has to prove that  $[\varphi] C_B \psi$  is in a maximal consistent set  $\Gamma$  iff for every path that runs from  $\Gamma$  along steps from one of the agents in  $B$  and in which  $\varphi$  is

true, also  $[\varphi]\psi$  holds, and the right hand side of this iff has no counterpart in the object language. This motivated [90] to introduce a *relativised common knowledge operator*  $C_B(\varphi, \psi)$  which exactly captures the right hand side of the mentioned equation. Both the inference rule  $R7$  and the completeness proof of the logic based on this notion of common knowledge has a much more natural appearance.

## 8 EPISTEMIC FOUNDATIONS OF SOLUTION CONCEPTS

We now turn to the characterisation of solution concepts in games, formalised using epistemic constructs. To set the scene, let us consider the game in extensive form, depicted on the left hand side of Figure 16. It supposes we have two players,  $A$  and  $B$ , and  $A$  has to decide in the nodes labelled  $a, e, i$  and  $u$ , whereas  $B$  decides in  $b$  and  $d$ . The leaves of the tree are labelled with payoffs, the one in the left-most leaf for instance denoting that  $A$  would receive 1, and  $B$  would get 6.

A natural question now is: “suppose you are agent  $A$ . What would your decision be in the top node  $a$ ?” The obvious backward induction procedure determines  $A$ ’s ‘best’ move starting from the leaves. Suppose the game would end up in node  $u$ . Since  $A$  is *rational*, he prefers an outcome of 4 over 1, and hence he would move ‘left’ in  $u$ ; this is illustrated using the thick lines in the game on the right hand side of Figure 16. Now,  $B$  is *rational* as well, and he moreover *knows that  $A$  is rational*, so, when reaching node  $d$ , player  $B$  *knows* he has in fact a choice between a payoff of 4 (going ‘left’ in  $d$ ) and 3 (going right, and *knowing what  $A$  will do in  $u$* ). We do the same reasoning over nodes  $e$  and  $i$ , and end up with the choices with a thick line in the figure:  $A$  would go ‘right’ and ‘left’, respectively. Again, since  $B$  is *rational and knows that  $A$  is rational*, his payoffs in node  $b$  are 4 and 3, respectively, and he will choose ‘left’. Continuing in this fashion, note that  $A$ ’s choice for going left in  $a$  is based on (i) the fact that he is *rational*; (ii) the fact that he *knows that  $B$  is rational* and (iii) the fact that *he knows that  $B$  knows that he ( $A$ ) is rational*. In short, knowledge of each other’s rationality, and indeed, *common knowledge* thereof, seems to play a crucial role in rational strategic decision making.

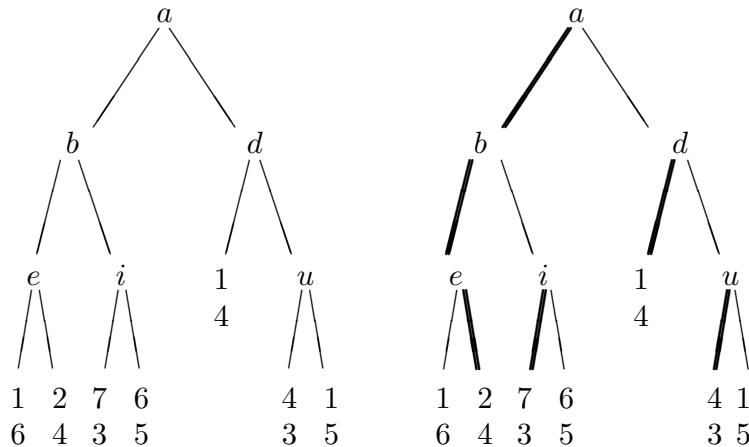


Figure 16. A Game in Extensive Form

The fact that epistemic notions are important in order to analyse certain solution

concepts in games, like common knowledge of rationality being crucial for backward induction, has been recognised for a long time, even though for certain game-theoretic solution concepts, the epistemic foundations are not always easy to determine. It seems most progress has been made in the case of strategic game forms.

### 8.1 Epistemic Foundations for Strategic Games

In ([18]) it is argued that a general form of epistemic characterisation results comes in a format in which one predicts the decision of each agent, given certain assumptions about each player's utility and rationality, and (iterated) knowledge thereof. The following provides an example, taken from [88]: we will mainly restrict ourselves to two player games in this section.

The proof of Theorem 7, and the backward induction algorithm applied above to recursively determine a subgame-perfect equilibrium in an extensive form game has its counterpart in strategic games. The left hand side of Table 6 represents a strategic game for two players  $r$  (choosing a row) and  $c$  (selecting a column). The entries  $x, y$  represent payoffs for player  $r$  and  $c$ , respectively. In this game, the (unique) Nash equilibrium can be achieved by iteratively removing strictly dominated strategies (see Section 2.1): since  $c$ 's strategy  $c_3$  is strictly dominated, by rationality he will not play it, so that we can remove its corresponding column from the game. Player  $r$  is rational and knows about  $c$ 's rationality, and in the new game with only 2 columns that he needs to consider *he* has a dominated strategy  $r_3$  that can be removed. Using  $\text{rat}_i$  to express that player  $i$  is rational, removal of  $r_3$  is granted by the fact that  $\text{rat}_r \wedge K_r \text{rat}_c$ . Continuing in this matter, column  $c_2$  and row  $r_2$  can subsequently be removed, leaving us with the unique Nash equilibrium  $\langle r_1, c_1 \rangle$ . It seems that in our reasoning the last two steps are motivated by the fact that

$$\text{rat}_c \wedge K_c \text{rat}_r \wedge K_c K_r \text{rat}_c \text{ and } \text{rat}_r \wedge K_r \text{rat}_c \wedge K_r K_c \text{rat}_r \wedge K_r K_c K_r \text{rat}_c \quad (19)$$

respectively. Before relating such a condition to [18]'s general format of characterisation results and his syntactic approach, we make a little detour.

$r \backslash c$	$c_1$	$c_2$	$c_3$	$r \backslash c$	$c_1$	$c_2$	$c_3$	
$r_1$	2,3	2,2	1,1	$r_1$	$\text{br}_r, \text{br}_c$	—	—	
$r_2$	0,2	4,0	1,0	$r_2$	$\text{br}_c$	$\text{br}_r$	—	
$r_3$	0,1	1,4	2,0	$r_3$	—	$\text{br}_c$	$\text{br}_r$	

Table 6. A strategic game  $H$  for players  $r$  and  $c$ , distribution of best responses and the model  $M_H$

#### A Semantic Approach

A semantic approach to clarify the nature of epistemic characterisation of solution concepts is given in [74], but here we follow a more recent approach ([88]). We do this since

the latter is closer to the Kripke models as used in this chapter, and it moreover gives a nice example of how dynamic epistemic logic (Section 7) can clarify the subtleties that are at stake here (for instance, is iterated knowledge as in (19) indeed *needed*?). Endowing finite two-player games in strategic form with an epistemic flavour uses the observation that each player knows his strategy, but not the other's. Hence, if we take the *strategy profiles*  $\sigma$  in a game  $G$  as the possible worlds, the epistemic indistinguishability relation that presents itself is then defined by  $R_i \sigma \delta$  iff  $\sigma_i = \delta_i$ . This is in fact the definition that is used in the prominent *distributed* or *interpreted systems* approach to epistemic logic (see Section 5), in which global states in the overall system have a local component for each agent, each exactly knowing this local state. Thus, the *full model* over the game  $H$  of Table 6 is the left model in Figure 17; where the global states  $\langle \sigma_r, \sigma_c \rangle$  are represented by their unique payoff for that profile.

As we have seen above, an algorithm to find a solution concept in a game  $G$  may transfer a model into a smaller one. Let us coin submodels of full models *general game models*. Doing so, we stay in the realm of **S5**<sub>2</sub>, since as [81] observes, every **S5**<sub>2</sub>-model is bisimilar to a general game model. Now, is common knowledge of rationality *needed* to justify the elimination of dominated strategies algorithm? First of all, referring to common knowledge in full games seems to be some overkill: in every full model for two players, every pair of strategy profiles  $\sigma = \langle \sigma_r, \sigma_c \rangle$  and  $\tau = \langle \tau_r, \tau_c \rangle$  is connected through a third  $\delta = \langle \sigma_r, \tau_c \rangle$  (and, indeed, a fourth  $\gamma = \langle \tau_r, \sigma_c \rangle$ ), which immediately gives us  $K_r K_c \varphi \rightarrow K_c K_r \varphi$  as a valid scheme on such models. This is reminiscent of the property that in interpreted systems, common knowledge in every run (roughly, every sequence of global states in the full model) is constant (see Section 5). In full models for two agents, we have  $K_r K_c \varphi \rightarrow C \varphi$  (its contra-positive follows quickly from the semantic insight above: if  $\neg C \varphi$  holds at  $\sigma$  then  $\varphi$  is false at some  $\tau$ , which is in two steps connected through  $\delta$ , giving  $\hat{K}_r \hat{K}_c \neg \varphi$ —recall that  $\hat{K}$  is the dual of  $K$ ) so that there is a natural bound on the needed nesting of epistemic operators, and the notion of common knowledge is not really needed.

Secondly, we have to become a bit more precise of what rationality exactly amounts to in the current setting. The notion of a *best response* for agent  $i$  in a model  $M$  at state  $\sigma$  can be formalised as follows, where  $A_i$  is the set of actions available in the current game  $G$  to agent  $i$ . Let  $M$  be the (full or general) game model generated by  $G$  and, given a profile  $\sigma$ ,  $M, \sigma \models \pi(\cdot) \succeq_i \pi(i \mapsto a)$  means that the payoff, given profile  $\sigma$  is at least as big for  $i$  as for the profile that is like  $\sigma$ , but in which  $i$  plays  $a$  instead.

$$M, \sigma \models \text{br}_i \text{ iff } M, \sigma \models \bigwedge_{a \in A_i} (\pi(i \mapsto a) \preceq_i \pi(\cdot)) \quad (20)$$

Note that a profile  $\sigma$  being in a Nash Equilibrium (**NE**) in a model  $M$ , can be described as follows:  $M, \sigma \models \text{NE} \leftrightarrow \bigwedge_{i \in N} \text{br}_i$ . In the general model  $M_H$  in the right hand side of Table 6, **NE** is true in the profile  $\langle r_1, c_1 \rangle$ . Note that we have  $M_H \models (\neg K_r \text{br}_r \wedge \neg K_c \text{br}_c)$ : neither player knows that he plays a best response, let alone that such a property is common knowledge.

Once a property  $\text{br}_i$  is true in  $\sigma$  in a full model  $M$ , it remains true in  $\sigma$  and any smaller general model  $M'$ : this is because of the universal nature of  $\text{br}$ . One could alternatively define a relative notion of best response  $\text{br}_i^*$  as in (20), but with quantification restricted to all actions  $A_i(M)$  in the model  $M$ . Doing so, a profile  $\sigma$  can *become* a best response for a player  $i$ , just because a better option for  $i$  has just been eliminated, when moving

to  $M'$ .

According to [88], a static analysis of strategic games using epistemics does not seem the right way to go: a general model for a game has players with common knowledge of rationality ( $CRat$ ), iff  $Rat$  is true in all worlds, i.e., profiles, and such games seem not so interesting, under plausible notions of rationality. For instance, if we take rationality to mean that everybody plays a best response, then we would have that  $CRat$  holds iff the model consists only of Nash equilibria. And, giving a modal logic argument, from the validity of  $C(Rat \rightarrow NE)$  one derives  $CRat \rightarrow CNE$ , so that, assuming common knowledge of rationality, there is no need for an algorithm to eliminate worlds, since we already have  $CNE$ : it is already common knowledge that a Nash equilibrium is played. Moreover,  $CRat$  is not a plausible assumption as was illustrated in the game  $H$  of Table 6: not even any individual player knows that he is rational! The dynamic epistemic logic of Section 7 seems more appropriate to deal with assumptions about rationality.

Rationality should have an epistemic component. Rather than saying that all players play their best response, it should cover something like 'every player plays the best, given his knowledge'. We take the following notion of weak rationality  $wr$  from [88], using our notation  $\hat{K}_i\varphi$  for  $\exists\tau(R_i\sigma\tau \ \&\varphi \text{ is true at } \tau)$ .

$$M, \sigma \models wr_i \text{ iff } M, \sigma \models \bigwedge_{a \in Ac_i, a \neq \sigma_i} \hat{K}_i(\pi(\cdot) \succeq_i \pi(i \mapsto a)) \quad (21)$$

In words: a player  $i$  is weakly rational at  $\sigma$ , if for every alternative action for  $\sigma_i$ , he can imagine a profile  $\sigma'$ , for which  $\sigma_i$  (which is, by the definition of the accessibility relation,  $\sigma'_i$ ) would be at least as good for  $i$  as playing  $a$ . Alternatively:  $\sigma$  is a weakly rational profile for  $i$ , if for any alternative action,  $i$  does not know that  $\sigma_i$  is worse. It may be instructive to consider the dual reading of  $wr$ , which is  $\neg \bigvee_{a \in Ac_i(M), a \neq \sigma_i} \hat{K}_i(\pi(\cdot) < \pi(i \mapsto a))$ :  $\sigma$  is weakly rational for  $i$  if there is no action  $a$  for  $i$  for which  $i$  knows that it would give him a better payoff. Yet phrased differently:  $\sigma$  is not weakly rational for a player, if it represents a dominated strategy. Going back to the model  $M_H$  of Figure 17,  $wr_c$  fails exactly in the third column, since there, player  $c$  *does* know to have a strictly better move than playing  $c_3$ .

**THEOREM 35** ([88]). *Every finite general model has worlds where  $wr_i$  holds for both players  $i$ .*

Being defined as lack of knowledge, and given the fact that we are in the realm of  $\mathbf{S5}_m$ , we have  $wr_i \rightarrow K_i wr_i$  as a validity, giving rationality its desired epistemic component. Moreover, we can expect players to announce that they are weakly rational, since they would know it. The next theorem uses the public announcements of Section 7.

**THEOREM 36** ([88]). *Let  $\sigma$  be a strategy profile,  $G$  a game in strategic form, and  $M(G)$  its associated full game model. Then the following are equivalent:*

- (i) *Profile  $\sigma$  survives when doing iterated removal of dominated strategies;*
- (ii) *Repeated announcement of  $wr_i$  stabilises at a substate  $(N, \sigma)$ , for which the domain of  $N$  is exactly the set of states that survive when doing iterated removal of dominated strategies.*

In Figure 17, the left-most model is the model  $M_H$  from Table 6. The other models are obtained by public announcements of  $wr_c, wr_r, wr_c$  and  $wr_r$ , respectively. As a local



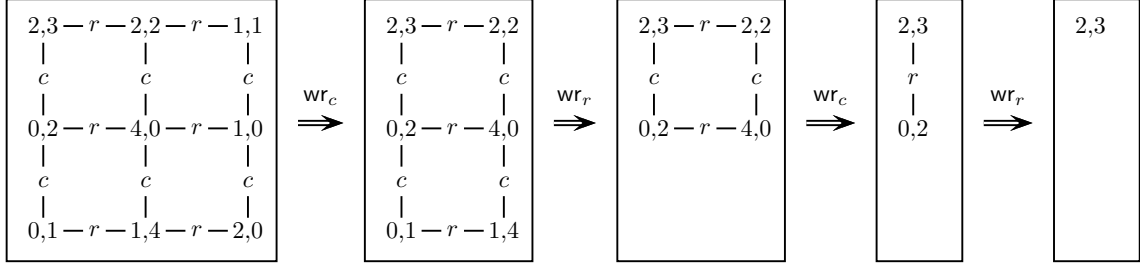


Figure 17. Epistemic model  $M_H$  for the game  $H$  (left), and after announcements of rationality.

update, this sequence can be executed in  $\sigma = \langle r_1, c_1 \rangle$ , but one can also conceive it as an operation on the whole model, cf. the interpretation of the learning operators in Section 7.2. Thus, in our model  $M_H$ , we have  $M_H, \langle r_1, c_1 \rangle \models [\text{wr}_c][\text{wr}_r][\text{wr}_c][\text{wr}_r]CNash$ : if the players iteratively announce that they are weakly rational, the process of dominated strategy elimination leads them to a solution that is commonly known to be Nash.

One can give a similar analysis of *strong rationality*. A profile  $\sigma$  satisfies this property for a player  $i$  if  $i$  considers it possible that  $\sigma_i$  is as good as any other of  $i$ 's actions.

$$M, \sigma \models \text{sr}_i \text{ iff } M, \sigma \models \hat{K}_i \bigwedge_{a \in A_i, a \neq \sigma_i} (\pi(\cdot) \succeq_i \pi(i \mapsto a)) \quad (22)$$

If we abbreviate SR to be  $\text{sr}_r \wedge \text{sr}_c$ , we can again perform announcements of SR. In general, such announcements of  $\text{wr}$ ,  $\text{sr}$  and SR give rise to different behaviour.

**THEOREM 37 ([88]).** *On full models, repeated announcements of SR lead to its common knowledge.*

We have assumed that games are finite. For infinite games, the reasoning put forward in this section would be naturally dealt with in some kind of fixed-point logic. We have for instance the following theorem:

**THEOREM 38 ([88]).** *The stable set of worlds for repeated announcement of SR is defined in the full game model by the greatest fixed point formula*

$$\nu p \cdot (\hat{K}_c(\text{br}_c \wedge p) \wedge \hat{K}_r(\text{br}_r \wedge p))$$

### A Syntactic Approach

One of the main contributions of [18] is that it gives a unifying modal framework to present and relate several epistemic characterisation results of solution concepts. The claim is that such results, for a two-player strategic form game, are usually expressed in a form  $\varphi(\text{rat}_1, \text{rat}_2, \mathbf{u}_1, \mathbf{u}_2) \rightarrow \text{actions}$ . This is demonstrated by (19), albeit that there the assumptions about the utilities (the  $\mathbf{u}$ -propositions) are kept implicit in the model. It goes without saying that for instance player  $c$  can only apply his knowledge about  $r$ 's rationality if he also knows  $r$ 's utilities. A syntactic approach forces us to make such assumptions explicit.

Let us here briefly explain how [18] formalises a characterisation result for strategic form games. First of all, we need a language with knowledge operators  $K_i$  and probabilistic belief operators  $\mathbf{P}_i$ , where the intended interpretation of  $\mathbf{P}_i(\varphi) = r$  (with  $r$  a

rational number) is obvious. Basic propositions are  $i_1, i_2, \dots$  expressing that player  $i$  plays his first, second, ... strategy. The expression  $u_i(k, l) = r_{i,k,l}$  denotes that the utility for player  $i$ , when the strategy profile  $(k, l)$  is played, equals the number  $r$ .

The axioms needed are then  $A1, A2$  and  $A3$  for knowledge (see Table 2), axioms for dealing with inequalities of terms referring to probabilities of events, like  $c > 0 \Rightarrow (\sum_k q_k \mathbf{P}_i(\varphi) \geq r \Leftrightarrow \sum_k c q_k \mathbf{P}_i(\varphi)_k \geq cr)$ . Add the Kolmogorov axioms for  $\mathbf{P}_i$ , which say that  $\mathbf{P}_i(\top) = 1$ ,  $\mathbf{P}_i(\perp) = 0$ ,  $\mathbf{P}_i(\varphi) \geq 0$ ,  $\mathbf{P}_i(\varphi) = \mathbf{P}_i(\varphi \wedge \psi) + \mathbf{P}_i(\varphi \wedge \neg\psi)$  and  $\mathbf{P}_i(\varphi) = \mathbf{P}_i(\psi)$  whenever  $\varphi \leftrightarrow \psi$  is a propositional tautology. The connection axioms relate knowledge and probabilistic belief:  $\mathbf{P}_i(\varphi) = 1$  is the same as  $K_i\varphi$ , and for every  $i$ -probability sentence  $\varphi$  (i.e., a Boolean combination of statements of the form  $\mathbf{P}_i(\cdot) = \cdot$ ), we have  $\varphi \rightarrow K_i\varphi$ . The inference rules are  $R1$  and  $R2$  of Table 2.

Then there are axioms specifically for strategic game forms, called *axioms for game playing situations*. These are, respectively, that every player plays at least one strategy ( $\bigwedge_i \bigvee_m i_m$ , where  $m$  ranges over the strategies available to player  $i$ ), and not any other ( $\bigwedge_i \bigwedge_{m \neq n} \neg(i_m \wedge i_n)$ ). Moreover, every player knows his chosen strategy ( $\bigwedge_i \bigwedge_m K_i i_m \leftrightarrow i_m$ ), and likewise for his utilities ( $u_i(k, l) = r \rightarrow K_i u_i(k, l) = r$ ). The scheme that captures rationality of player  $i$  is called  $\text{meu}_i$ , which is defined as the following implication:

$$K_i((\bigwedge_{k,l} u_i(k, l) = r_{i,k,l}) \wedge \bigwedge_l (\mathbf{P}_i(j_l = p_l)) \wedge i_m) \rightarrow \bigwedge_k \sum_l p_l \cdot r_{i,m,l} \geq \sum_l p_l \cdot r_{i,k,l}$$

expressing that each player  $i$  aims at his Maximal Expected Utility: if  $i$  knows all his possible utilities in the game, and the probabilities with which his opponent  $j$  chooses his strategies, and  $i$  opts for his strategy  $m$ , then, for every alternative choice  $k$  for  $i$ , the expected utility for  $i$ , using  $i$ 's expectations of  $j$ 's behaviour, will never be bigger.

**THEOREM 39 ([18]).** *Let  $\Gamma$  be a 2-person normal form game. Assume that the following three conditions hold (where player 1 plays  $m$  and 2 plays  $n$ ).*

- (i) *All players are rational*  $\bigwedge_i \text{meu}_i$
- (ii) *All players know their own utility function*  $\bigwedge_i K_i \bigwedge_{k,l} u_i(k, l) = r_{i,k,l}$
- (iii) *All players know each player's actual choice*  $K_1 2_n \wedge K_2 1_m$

*Then, the played action profile constitutes a Nash equilibrium, i.e., we have  $\bigwedge_k r_{1,m,n} \geq r_{1,k,n} \wedge \bigwedge_l r_{2,m,n} \geq r_{2,m,l}$ .*

Note that the theorem above does not refer to common knowledge at all. As [18] points out, Theorem 39 is in fact well known in game theory (since 1995 [6], and even 1982, [73]). But once again, the added value of [18]'s analysis is that it can relate those approaches, and that framework for instance enables to point at weak spots in proofs of theorems similar to the one discussed here.

## 8.2 Epistemic Foundations for Extensive Games

The analysis in [18] is particularly interesting in the realm of extensive games, especially by pinpointing the difference between two interpretations of them: the *one-shot interpretation* on the one hand, and the *many-moment interpretation* on the other. The first interpretation is the one propagated by the key publication in game theory ([54]) and renders extensive games 'the same' as games in normal form: players act only once. Metaphorically speaking, under the one-shot interpretation, players can be thought of as making up their mind before the game really starts, and then all submit their chosen

strategy in a closed envelope to a referee. The outcome of the game is then completely determined, even without any player really ‘performing’ a move. In the many moment interpretation, a player only has to make a decision for the decision node he thinks he is at (and he thinks is his).

Let us briefly revisit the game on the left hand side of Figure 16. In the one-shot interpretation, as part of player 2’s strategy, 2 makes a decision in node  $d$ , and rationality imposes him to choose ‘left’, and the decision  $\langle d, left \rangle$  is part of the strategy that he will put in his envelop. (And the fact that 1 can predict this, makes the whole idea of backward induction work). In the many-moment interpretation however, if player 2 ever finds himself making a choice in node  $d$ , he obviously has to re-think his assumptions about the situation, and in particular, about player 1’s rationality. It need not come as a surprise that this second interpretation has led several researchers to analyse this using *belief revision* or *counterfactuals* (see also Chapters 18 and 21 of this handbook), most notably by [75, 76].

Especially the analysis of common knowledge (or, for that matter, belief) is much harder under the second analysis, because “... (i) at one single decision moment only the beliefs of one player are relevant ... and (ii) because we have to decide whether common beliefs involve past beliefs, or future beliefs, or both” ([18, Chapter 4]). For a more philosophical view regarding the two interpretations, we refer to [18], here we indicate only how a formal analysis clarifies the difference.

For a proof system, we now take the axioms  $A1, A2$  and  $A6 - A10$  for  $K_i, E$  and  $C$  (note that since  $A3$  is not assumed, we will call the attitude ‘belief’ but still write  $K_i$ ), and the axioms for games from Section 8.1. On top of this, we add the following atoms to the language. Let  $i_k^x$  mean that player  $i$  chooses according to his  $k$ -th strategy in the subgame generated by node  $x$ , and  $u_i^x(k, l) = r$  denotes that the utility for  $i$ , when the strategies  $k$  and  $l$  are played in the subgame generated by  $x$ , is  $r$ . An axiom  $\text{KnUtSub}$  says that these utilities are known by player  $i$ .

The principles of rationality now needed are  $\text{nrat}_i$  (“on-path rationality”) and  $\text{frat}_i$  (“off-path rationality”), with the following axioms:

$$\begin{array}{ll} \text{NRat}_{bas} & \text{nrat}_i^x \rightarrow \text{nsd}_i(\mathcal{A}_i, \mathcal{A}_j) \\ \text{NRat}_{ind} & (\text{nrat}_i^x \wedge K_i X_i \wedge K_i X_j) \rightarrow \text{nsd}_i(X_i, X_j) \\ \text{FRat} & \text{frat}_i \leftrightarrow \bigwedge_{\rho \preceq x} \text{nrat}_i^x \end{array}$$

The last axiom says that off-path rationality means on-path rationality at every subgame (generated by an arbitrary node  $x$  reachable from  $\rho$ , the root of the game). The statement  $\text{nsd}_i(X_i, Y_j)$  means that  $i$  plays a strategy that is not strictly dominated in the game generated by  $i$ ’s strategies  $X_i \subseteq \mathcal{A}_i$  (the latter being  $i$ ’s set of actions) and  $j$ ’s strategies  $Y_j \subseteq \mathcal{A}_j$ . On-path rationality is expected to model that “a player chooses a (full) strategy that given his beliefs prescribes optimal actions along the path that he expects will ensue” ([18, Chapter 4]). The base case for  $\text{nrat}_i$  then expresses that a full strategy that is optimal in a subgame, is not dominated. Similarly, the induction axiom says that a strategy that is optimal given the expectations of player  $i$  about how the game is played, is not strictly dominated in the game in which all other paths are eliminated.

**THEOREM 40** ([18]). *Let  $\Gamma$  be a finite two-person extensive form game, with given utilities  $\bigwedge_{i,k,l} u_i(k, l) = r_{i,k,l}$ , for which the following hold*

- (i) The players are rational  $\bigwedge_i \text{frat}_i$
- (ii) (i) is common belief  $C(\bigwedge_i \text{frat}_i)$
- (iii) the utilities are commonly believed  $C(\bigwedge_{i,k,l} u_i(k,l) = r_{i,k,l})$

Then the players play the strategy profile that is generated using backward induction.

What about the many-moment interpretation? First of all, [18] adds some specific axioms, saying that every player knows always (that is, everywhere in the game) all payoffs, for any strategy profile, and that a player knows where he is in the game (Kn-Where:  $K_i^x \bigwedge_j \bigvee_{y \prec x, j_i \in D} j_k^y$ , where  $D$  collects all full strategies that are consistent with reaching  $x$ ). The axiom KnStratM ( $i_k \leftrightarrow \bigwedge_{\rho \preceq x} K_i^x i_k$ ) is rather pushing the limits of the many-moment interpretation: it says that player  $i$ , when playing strategy  $i_k$ , knows this wherever he ends up in the game.

The rationality principles  $RRat$  are a bit more involved, now:

$$\begin{aligned} \text{rrat}_i^x \leftrightarrow & ((K_i^x \bigwedge_{k,l} u_i^x(k,l) = r_{i,k,l} \wedge \bigwedge_k \mathbf{P}_i^x(i_k^x) = p_k \wedge \bigwedge_l \mathbf{P}_j^x(j_l^x) = p_l \wedge i_m(x)) \\ & \rightarrow \bigwedge_k EU_i(m, x, \mathbf{P}_i^x) \geq EU_i(k, x, \mathbf{P}_i^x)) \end{aligned}$$

The property  $RRat$  above states that it holds for player  $i$  at node  $x$  iff the following implication holds. The antecedent states that  $i$  knows at  $x$  the utilities for every strategy profile, and he has some probabilistic beliefs about the strategies that both he and his opponent  $j$  will choose at  $x$ . Moreover,  $i$  in fact chooses the action prescribed by his  $m$ -th strategy at  $x$ . If this is fulfilled, then the chosen strategy  $m$  should give at least the utility obtained using any other option  $k$ . Here,  $EU_i(m, x, \mathbf{P}_i^x)$  is the expected utility for  $i$  in the node  $y$  that he reaches immediately from  $x$  when using strategy  $m$ .

We now show how a negative result can be proven ‘during progress in a game’, using the two-player centepede example. In [70]’s variant of centepede, (see Figure 18), there are three decision nodes  $\rho$ ,  $y$  and  $z$ . Of the two players, player 2 only moves in  $y$ . The payoffs for the players are represented as pairs  $(u_1, u_2)$  in the leaves of the tree, where  $u_i$  represents player  $i$ ’s payoff. We leave it to the reader to check that backward induction leads player 1 to play down  $D_1$  at the root  $\rho$ . [18] formalises [70]’s claim about this game that says that in node  $y$ , there cannot be common knowledge of rationality.

To make the proof work, we need two persistence properties. The first applies to strategies, and says that  $i$ ’s prediction about  $j$ ’s choices should not change if nothing unexpected happens: if the nodes  $x, y$  and  $z$  appear in a path on the tree (in that order), then  $\mathbf{P}_i^x(j_j^z)$  and  $\mathbf{P}_i^y(j_j^z)$  should coincide. The other preservation property applies to rationality, and says that  $i$  will not change his rationality assumptions about  $j$  during a period that  $j$  did not make a move:  $K_i^x \text{rrat}_j^y \leftrightarrow K_i^x \text{rrat}_j^z$  if  $x, y$  and  $z$  appear in a path on a tree (in that order, possibly  $x = y$ ), and  $z$  is the first node from  $y$  where  $j$  moves.

The proof in [18] of the claim  $C^y \text{rrat}^y \rightarrow \perp$  runs as follows. First of all, we show (a)  $C^y \text{rrat}^y \rightarrow K_2^y K_1^\rho d_1$  (if the principle of  $\text{rrat}$  at  $y$  is common knowledge at  $y$ , then 2 knows at  $y$  that 1 knows at  $\rho$  that 2 plays  $d_1$ ). To prove it, we prove (i)  $C^y \text{rrat}^y \rightarrow K_1^\rho d_1$  and then apply necessitation for  $K_2^y$ . Using the preservation of strategies principle, it is sufficient to prove (ii)  $C^y \text{rrat}^y \rightarrow K_1^y d_1$ . Applying  $K_1^y$ -necessitation to the rationality axiom gives us  $K_1^y \text{rrat}_2^y \wedge K_1^y K_2^y D_2 \rightarrow K_1^y d_1$ , so that, to prove (ii), we are done if we show (iii)  $C^y \text{rrat}^y \rightarrow (K_1^y \text{rrat}_2^y \wedge K_1^y K_2^y D_2)$ . The first conjunct is immediate, for the second, use the persistence of rationality (for  $K_2^y$ ) and necessitation (for  $K_1^y$ ) to show  $K_1^y K_2^y \text{rrat}_1^y \rightarrow K_1^y K_2^y \text{rrat}_1^z$ . Applying rationality to this gives the second conjunct.

Furthermore, from the persistence of rationality, it is easy to show (b)  $C^y \text{rrat}^y \rightarrow$

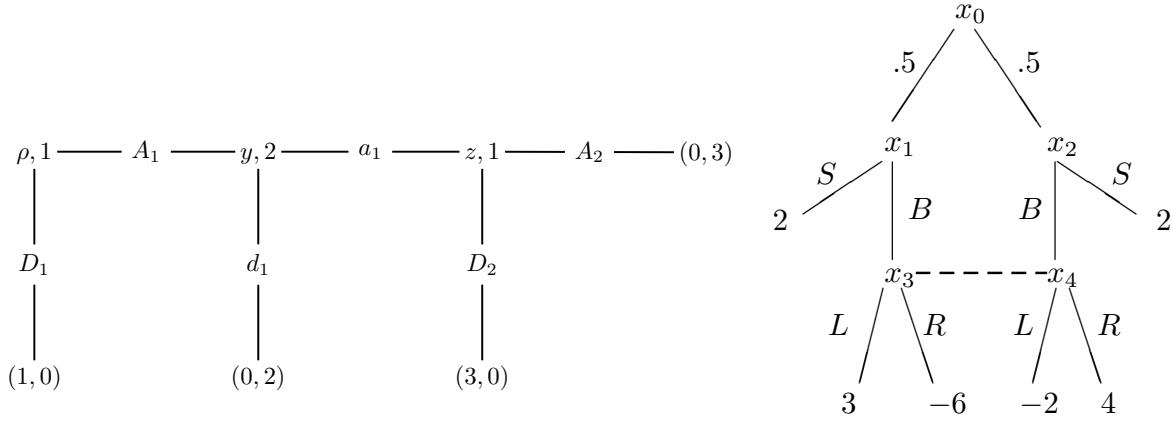


Figure 18. Centepede (left) and a one-person game with perfect recall (right)

$K_2^y \text{rrat}_1^\rho$ . Also, rationality and necessitation give  $(c)(K_2^y K_1^\rho d_1 \wedge K_2^y \text{rrat}_1^\rho) \rightarrow K_2^y \neg A_1$ . Combining (a), (b) and (c) immediately gives us  $C^y \text{rrat}^y \rightarrow K_2^y \neg A_1$ . However, the  $D$  axiom ( $\neg K_2^y \perp$ ) together with the derivability of  $K_2^y A_1$  from the KnWhere axiom (see above) then establishes our claim  $C^y \text{rrat}^y \rightarrow \perp$ .

Which would again emphasise the intuition uttered by many game theorists that in the analysis with which we started this section, backward induction must be based on some form of counterfactual reasoning.

### 8.3 On the Representation of Games with Imperfect Information

The manuscript [28] gives a neat discussion on issues that arise when modelling a game with imperfect information. We already argued that the extensive form of a game gives a more natural account of the temporal aspects of a game than its strategic form. For imperfect information games, we can add equivalence relations to denote the player's information sets. However, such information sets can for instance not capture knowledge or beliefs that a player has over the strategies of the other players, or about their rationality. There are of course other ways of representing uncertainty in games, like for instance what [28] calls a *state space* representation, in which one for instance associates a strategy profile to every state (cf. the representation used in Section 8.1 and Figure 17). But in those representations, one loses the ability again to represent the temporal information. In fact, [28]'s main point is a plea to use *interpreted systems* (see Section 5) to model games of imperfect information, because they make explicit where the knowledge comes from.

Let us present [28]'s argument representing games as systems using the game that it borrows from [67], represented as the game on the right of Figure 18. In  $x_0$ , nature chooses randomly either  $x_1$  or  $x_2$ , and then the only player can chose either  $S$  or  $B$ . He cannot distinguish  $x_3$  from  $x_4$ , where he has the option  $L$  or  $R$ . Using uniform strategies only, let the strategy  $\sigma = \langle B, S, R \rangle$  denote that our player plays  $B$  in  $x_1$ ,  $S$  in  $x_2$  and  $R$  in the information set  $\{x_3, x_4\}$ . Similarly, let  $\sigma'$  be  $\langle B, S, L \rangle$ . One easily verifies that the payoff 3 when playing  $\sigma$  is the maximal expected utility: no other strategy is as good. One could easily implement such a strategy using an interpreted system where the local state  $\ell$  of the player would indicate whether he is in  $\{x_1\}$ , in  $\{x_2\}$  or in  $\{x_3, x_4\}$ .

Now [67] argues that if node  $x_1$  is reached, the agent is better off by changing from strategy  $\sigma$  to  $\sigma'$ . And, as [28] argues, this is right, if the agent is able to remember that he switched strategies. Since the agent cannot distinguish  $x_3$  from  $x_4$ , he should use a uniform strategy and do the same in both. However, if the agent would have perfect recall, he would distinguish the nodes, and by allowing him to remember that he switched strategies he can simulate perfect recall: if he ends up in  $\{x_3, x_4\}$ , he must realize that he came through  $x_1$  and is now playing  $\sigma'$ . Under this assumption it is not clear anymore what it means to have a dotted line between  $x_3$  and  $x_4$ , since the states become distinguishable. Using interpreted systems it is very natural to encode in the local state of the agent that he switched strategies, by recording the strategies he has been playing until now, for instance.

## 9 GAME LOGIC

Computer science has developed many different logics for reasoning about the behaviour of computer programs or algorithms. Propositional Dynamic Logic (PDL) (see [33, 34] and also Chapter 12 of this handbook) is a well-studied example which contains expressions  $[\pi]\varphi$  stating that all terminating executions of program  $\pi$  will end in a state satisfying  $\varphi$ . What distinguishes PDL from simple multi-modal logic is that  $\pi$  can be a complex program such as  $p_1; p_2$ , the sequential execution of first  $p_1$  and then  $p_2$ . PDL formalises properties of programs in its axioms such as  $[p_1; p_2]\varphi \leftrightarrow [p_1][p_2]\varphi$  which completely characterises the sequential composition operator.

Nondeterministic programs may be viewed as particularly simple games, namely 1-player games. Examples of algorithms involving more than one player or agent are cake-cutting algorithms, voting procedures and auctions. Also structurally, games are very similar to programs. One game may be played after another, a player may choose to play a game repeatedly, and so on. Hence, one may expect reasoning about games to be similar to reasoning about programs, and consequently game logics should resemble program logics.

Game Logic (GL), introduced in [58, 59], is a generalisation of PDL for reasoning about determined 2-player games, allowing us to describe algorithms like the cake-cutting algorithm and to reason about their correctness. GL extends PDL by generalising its semantics and adding a new operator to the language. The meta-theoretic study of PDL has given us valuable insights, e.g., into the complexity of reasoning about programs and the expressive power of various programming constructs. By comparing GL to PDL, we can get an idea of how reasoning about games differs from reasoning about programs. In this section, we can only introduce the syntax and semantics of Game Logic (Subsection 9.1) and discuss some of its central meta-theoretic properties (Section 9.2). Further topics of research are briefly mentioned in Section 9.3. The interested reader is referred to a recent survey article on the subject [65] with more detailed references.

### 9.1 Syntax and Semantics

The games of Game Logic involve two players, player 1 (Angel) and player 2 (Demon). Just like PDL, the language of GL consists of two sorts, propositions and (in the case of GL) games. Given a set of atomic games  $\Gamma_0$  and a set of atomic propositions  $\Phi_0$ , games  $\gamma$  and propositions  $\varphi$  can have the following syntactic forms, yielding the set of GL-games

$\Gamma$  and the set of GL-propositions/formulas  $\Phi$ :

$$\begin{aligned}\gamma &:= g \mid \varphi? \mid \gamma_1; \gamma_2 \mid \gamma_1 \cup \gamma_2 \mid \gamma^* \mid \gamma^d \\ \varphi &:= \perp \mid p \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \langle \gamma \rangle \varphi\end{aligned}$$

where  $p \in \Phi_0$  and  $g \in \Gamma_0$ . As usual, we define  $[\gamma]\varphi := \neg\langle \gamma \rangle \neg\varphi$ . The formula  $\langle \gamma \rangle \varphi$  expresses that Angel has a strategy in game  $\gamma$  which ensures that the game ends in a state satisfying  $\varphi$ .  $[\gamma]\varphi$  expresses that Angel does not have a  $\neg\varphi$ -strategy, which by determinacy is equivalent to saying that Demon has a  $\varphi$ -strategy. To provide some first intuition regarding the game operations,  $\gamma_1 \cup \gamma_2$  denotes the game where Angel chooses which of the two subgames to continue playing, and the sequential composition  $\gamma_1; \gamma_2$  of two games consists of first playing  $\gamma_1$  and then  $\gamma_2$ . In the iterated game  $\gamma^*$ , Angel can choose how often to play  $\gamma$  (possibly not at all); each time she has played  $\gamma$ , she can decide whether to play it again or not. Playing the dual game  $\gamma^d$  is the same as playing  $\gamma$  with the players' roles reversed, i.e., any choice made by Angel in  $\gamma$  will be made by Demon in  $\gamma^d$  and vice versa. The test game  $\varphi?$  consists of checking whether a proposition  $\varphi$  holds at that position. This construction can be used to define conditional games such as  $(p?; \gamma_1) \cup (\neg p?; \gamma_2)$ : Suppose for instance that  $p$  holds at the present state of the game, then Angel will naturally choose the left side (if she chooses the right side, she loses at once), and  $\gamma_1$  will be played. Otherwise Angel will choose right and  $\gamma_2$  will be played.

Thanks to the dual operator, demonic analogues of these game operations can be defined. Demonic choice between  $\gamma_1$  and  $\gamma_2$  is denoted as  $\gamma_1 \cap \gamma_2$  which abbreviates  $(\gamma_1^d \cup \gamma_2^d)^d$ . Demonic iteration of  $\gamma$  is denoted as  $\gamma^\times$  which abbreviates  $((\gamma^d)^*)^d$ .

A further note on iteration: In  $\gamma^*$ , it is essential that Angel can decide as the game proceeds whether to continue playing another round of  $\gamma$  or not. The game where Angel has to decide up front how often to play  $\gamma$  is a different game and in general more difficult for Angel to win than  $\gamma^*$ . For programs (i.e., in PDL), these two notions of iteration coincide, but for games (i.e., in GL), they do not.

The formal semantics of Game Logic utilises the following game models which generalise the neighbourhood models or minimal models used in the semantics of non-normal modal logics [17]. A *game model*  $\mathcal{M} = ((S, \{E_g \mid g \in \Gamma_0\}), V)$ , consists of a set of states  $S$ , a valuation  $V : \Phi_0 \rightarrow \mathcal{P}(S)$  for the propositional letters and a collection of *neighbourhood functions*  $E_g : S \rightarrow \mathcal{P}(\mathcal{P}(S))$  which are monotonic, i.e.  $X \in E_g(s)$  and  $X \subseteq X'$  imply  $X' \in E_g(s)$ .

The intuitive idea is that  $X \in E_g(s)$  (alternative notation:  $sE_g X$ ) holds whenever Angel has a strategy in game  $g$  to achieve  $X$ . Intuitively, neighbourhood functions are reduced monotonic effectivity functions (see Section 2.3) in that they only represent the effectivity of a single player. Since we are dealing with determined games only, it is sufficient to represent Angel's effectivity. Then Demon is effective for  $X$  if and only if Angel is not effective for its complement  $\overline{X}$ . Furthermore, neighbourhood functions do not satisfy the two conditions we put on effectivity functions, i.e., due to the presence of the test operator  $?$ , we allow that  $\emptyset \in E_g(s)$  and  $S \notin E_g(s)$ . For Angel, these conditions correspond to heaven (a game where Angel can achieve anything at all) and hell (a game where Angel can achieve nothing whatsoever).

By simultaneous induction, we define truth in a game model on the one hand and the neighbourhood functions for non-atomic games on the other hand. The function  $E_\gamma : S \rightarrow \mathcal{P}(\mathcal{P}(S))$  is defined inductively for non-atomic games  $\gamma$  (where  $E_\gamma(Y) = \{s \in$

$S|sE_\gamma Y\}$ ), and the truth of a formula  $\varphi$  in a model  $\mathcal{M} = ((S, \{E_g | g \in \Gamma_0\}), V)$  at a state  $s$  (denoted as  $\mathcal{M}, s \models \varphi$ ) is defined by induction on  $\varphi$ . We define

$$\begin{array}{llll}
\mathcal{M}, s \not\models \perp & & E_{\alpha;\beta}(Y) & = E_\alpha(E_\beta(Y)) \\
\mathcal{M}, s \models p & \text{iff } p \in \Phi_0 \text{ and } s \in V(p) & E_{\alpha \cup \beta}(Y) & = E_\alpha(Y) \cup E_\beta(Y) \\
\mathcal{M}, s \models \neg \varphi & \text{iff } \mathcal{M}, s \not\models \varphi & E_{\varphi?}(Y) & = \overline{\varphi^\mathcal{M} \cap Y} \\
\mathcal{M}, s \models \varphi \vee \psi & \text{iff } \mathcal{M}, s \models \varphi \text{ or } \mathcal{M}, s \models \psi & E_{\alpha^d}(Y) & = \overline{E_\alpha(\overline{Y})} \\
\mathcal{M}, s \models \langle \gamma \rangle \varphi & \text{iff } sE_\gamma \varphi^\mathcal{M} & E_{\alpha^*}(Y) & = \bigcap \{X \subseteq S \mid Y \cup E_\alpha(X) \subseteq X\}
\end{array}$$

where  $\varphi^\mathcal{M} = \{s \in S \mid \mathcal{M}, s \models \varphi\}$ . For iteration, our definition yields a least fixpoint, i.e., the smallest set  $X \subseteq S$  such that  $Y \cup E_\alpha(X) = X$ .

## 9.2 Metatheory

### Axiomatisation and Expressiveness

PDL has a very natural complete axiomatisation, and given the similarity between programs and games, one can hope that adding an axiom for the dual operator is all that is needed to obtain a complete axiomatisation of GL. A small problem is presented by the induction principle

$$(\varphi \wedge [\gamma^*](\varphi \rightarrow [\gamma]\varphi)) \rightarrow [\gamma^*]\varphi$$

which is valid in PDL but invalid in GL. While this induction principle usually forms part of the axiomatic basis of PDL, alternative axiomatisations exist which instead of the induction axiom use a fixpoint inference rule (given below). It turns out that this rule is indeed sound, yielding the following axiomatic system.

Let **GL** be the smallest set of formulas which contains all propositional tautologies together with all instances of the axiom schemas of Figure 19, and which is closed under the rules of Modus Ponens, Monotonicity and Fixpoint, also presented in Figure 19.

Axioms:	$\langle \alpha \cup \beta \rangle \varphi \leftrightarrow \langle \alpha \rangle \varphi \vee \langle \beta \rangle \varphi$	
	$\langle \alpha; \beta \rangle \varphi \leftrightarrow \langle \alpha \rangle \langle \beta \rangle \varphi$	
	$\langle \psi? \rangle \varphi \leftrightarrow (\psi \wedge \varphi)$	
	$(\varphi \vee \langle \gamma \rangle \langle \gamma^* \rangle \varphi) \rightarrow \langle \gamma^* \rangle \varphi$	
	$\langle \gamma^d \rangle \varphi \leftrightarrow \neg \langle \gamma \rangle \neg \varphi$	
Inference Rules:		
$\frac{\varphi \quad \varphi \rightarrow \psi}{\psi}$	$\frac{\varphi \rightarrow \psi}{\langle \gamma \rangle \varphi \rightarrow \langle \gamma \rangle \psi}$	$\frac{(\varphi \vee \langle \gamma \rangle \psi) \rightarrow \psi}{\langle \gamma^* \rangle \varphi \rightarrow \psi}$

Figure 19. The axioms and inference rules (Modus Ponens, Monotonicity and Fixpoint Rule) of Game Logic.

Intuitively, the axiom for iteration (our fourth axiom) states that  $\langle \gamma^* \rangle \varphi$  is a pre-fixpoint of the operation  $\varphi \vee \langle \gamma \rangle X$ . Conversely, the fixpoint rule states that  $\langle \gamma^* \rangle \varphi$  is the *least* such pre-fixpoint.



**THEOREM 41.** ***GL** is sound with respect to the class of all game models.*

At the time of writing, completeness of **GL** is still open, but some weaker results exist. If  $x$  is an operator of Game Logic such as  $d$  or  $*$ , let  $\mathbf{GL}^{-x}$  denote Game Logic without the operator  $x$ , i.e., restricted to formulas without the operator and without the axioms involving it.

**THEOREM 42** ([59, 62]). *Dual-free Game Logic  $\mathbf{GL}^{-d}$  and iteration-free Game Logic  $\mathbf{GL}^{-*}$  are both sound and complete with respect to the class of all game models.*

Hence, we have axiomatic completeness for  $\mathbf{GL}^{-d}$  as well as  $\mathbf{GL}^{-*}$ , but iteration together with duality remains a problem for axiomatisation. It may then not come as a surprise that it is precisely this combination which gives Game Logic its expressive power. This is most easily demonstrated when considering GL interpreted over Kripke models.

Kripke models can be viewed as special kinds of game models, namely game models where neighbourhood functions have a special property called *disjunctivity*: for every  $g \in \Gamma_0$  and  $V \subseteq \mathcal{P}(S)$  we have  $\bigcup_{X \in V} E_g(X) = E_g(\bigcup_{X \in V} X)$ . Hence, one may also investigate Game Logic when interpreted over Kripke models (i.e., disjunctive game models) only. Dual-free Game Logic over Kripke models is nothing but Propositional Dynamic Logic. Since the absence of infinite  $g$ -branches is not expressible in PDL but can be expressed by the GL-formula  $\langle (g^d)^* \rangle \perp$ , Game Logic over Kripke models is strictly more expressive than PDL.

Over Kripke models, full Game Logic can be embedded into the modal  $\mu$ -calculus (see Chapter 12 and [41]). In fact, the 2-variable fragment of the  $\mu$ -calculus suffices for the embedding, and since it has been shown that the variable hierarchy of the  $\mu$ -calculus is strict [10], Game Logic is a proper fragment of the  $\mu$ -calculus.

### Complexity

The two central complexity measures associated with a logic are the complexity of model checking and the complexity of the satisfiability problem. For the latter, we are interested to know the difficulty of determining whether a formula  $\varphi$  is satisfiable, measured in the length of the formula  $|\varphi|$ .

Using a translation of Game Logic formulas into modal  $\mu$ -calculus formulas, we can reduce Game Logic satisfiability to  $\mu$ -calculus satisfiability. Since Game Logic and the standard modal  $\mu$ -calculus are interpreted over different models, the models have to be translated as well. The result obtained via this procedure is the following:

**THEOREM 43** ([59, 62]). *The complexity of the satisfiability problem for Game Logic is in EXPTIME.*

Turning now toward model checking, given a game model  $\mathcal{M}$  and a Game Logic formula  $\varphi$ , we want to determine the set of states  $s$  for which  $\mathcal{M}, s \models \varphi$  holds. The complexity of model checking is usually measured in terms of the size of the formula and the size of the model. Given a game model  $\mathcal{M} = (S, \{E_g | g \in \Gamma_0\}, V)$ , we define its size  $|\mathcal{M}|$  as

$$|\mathcal{M}| = |S| + \sum_{\{s | s \in S\}} \sum_{\{g | g \in \Gamma_0\}} \sum_{\{X | s E_g X\}} |X|$$

Note that in practice one will want to represent game models more succinctly by only representing the non-monotonic core of  $E_g$ , i.e., we will disregard all those triples  $(s, g, X)$  for

which there is some  $Y \subseteq X$  such that  $sE_g Y$ . While in some cases (e.g., in case the game model corresponds to a Kripke model) such a representation can yield a dramatically more efficient representation, in general this is not the case, and hence the complexity result below cannot be essentially improved by disregarding supersets.

In the modal  $\mu$ -calculus, the complexity of current model-checking algorithms depends on the alternation depth of a formula, i.e., on the nestings of least and greatest fixpoint operators in the formula. For GL, the situation is similar, since angelic iteration corresponds to a least fixpoint and demonic iteration to a greatest fixpoint. Hence, the maximal number of nested demonic and angelic iterations will determine the model-checking complexity of the formula. As an example, the alternation depth of a formula  $\varphi$ , denoted as  $ad(\varphi)$  will be higher for  $\langle(g^*)^\times\rangle p$  than for  $\langle(g^*)^*\rangle p$ .

**THEOREM 44 ([62]).** *Given a Game Logic formula  $\varphi$  and a finite game model  $\mathcal{M}$ , model checking can be done in time  $O(|\mathcal{M}|^{ad(\varphi)+1} \times |\varphi|)$ .*

### 9.3 Other Topics

The notion of bisimulation has been the central notion of process equivalence for modal and dynamic logic (see Chapter 1, 5 and [12]). As for modal logic, modal formulas are invariant for bisimulation, i.e., bisimilar states satisfy the same modal formulas, and for finite models, the converse holds as well. Furthermore, it has been shown that the bisimilar fragment of first-order logic is precisely modal logic [80].

Bisimulation can be generalised from Kripke models to game models. Two states  $s$  and  $s'$  are bisimilar in case (i) they satisfy the same atomic properties, (ii) if player 1 has an  $X$ -strategy in game  $g$  from  $s$ , she also has an  $X'$ -strategy in  $g$  from  $s'$ , where every state in  $X'$  must have a bisimilar state in  $X$ , and (iii) analogously for strategies from  $s'$ . Intuitively, bisimilar states cannot be distinguished by either their atomic properties or by playing atomic games. It can be shown that this notion of bisimulation generalises bisimulation as it is normally defined for Kripke models. Similarly, one can show that bisimilar game models satisfy the same GL formulas, and one can even partially characterise the game operations of Game Logic in terms of bisimulation [61].

The operations of Game Logic have also been studied from an algebraic perspective [87]. We call two game expressions  $\gamma_1$  and  $\gamma_2$  *equivalent* provided that  $E_{\gamma_1} = E_{\gamma_2}$  holds for all game models. Put differently,  $\gamma_1$  and  $\gamma_2$  are equivalent iff  $\langle\gamma_1\rangle p \leftrightarrow \langle\gamma_2\rangle p$  is valid for a  $p$  which occurs neither in  $\gamma_1$  nor in  $\gamma_2$ . When  $\gamma_1$  and  $\gamma_2$  are equivalent, we say that  $\gamma_1 = \gamma_2$  is a *valid game identity*.

As an example, if a choice of a player between  $x$  and  $y$  is followed by game  $z$  in any case, then the player might as well choose between  $x; z$  and  $y; z$  directly. Hence,  $(x \cup y); z = x; z \cup y; z$  is a valid game identity. The right-distributive law  $x; (y \cup z) = x; y \cup x; z$  on the other hand is not valid. In the first game, player 1 can postpone her choice until after game  $x$  has been played. She may have a winning strategy which depends on how  $x$  is played, and hence such a strategy will not necessarily be winning in the second game, where she has to choose before  $x$  is played.

Game algebra further illustrates the link between games and processes that we already discussed in section 3.3. Basic game algebra studies the game operations of sequential composition, choice (demonic and angelic) and duality. The test-operator is excluded since it would take us out of the purely algebraic framework. The central result obtained for basic game algebra is a complete axiomatisation of the set of valid game identities

[25, 105]. So far, the complete axiomatisation has not been extended to a version of game algebra which includes iteration.

## 10 COALITION LOGIC

Modal logic describes transition systems at a very abstract level. The transition relation does not specify what or who is involved in making the transition, it only models all the possible evolutions of the system. Game forms on the other hand explicitly represent how different agents can contribute to the system's evolution by modeling the agents' strategic powers.

The semantic models of Coalition Logic [63, 64] make use of strategic games to describe the agents' abilities to influence system transitions. Using  $\alpha$ -effectivity functions (see Section 2.3), we can formalise an agent's ability to bring about  $\varphi$ . More generally, the expression  $[C]\varphi$  states that the coalition  $C$ , a subset of agents/players, can bring about  $\varphi$ . After presenting the syntax and semantics of Coalition Logic, Section 10.2 presents an axiomatisation of coalitional ability in extensive games of almost perfect information. Complexity results concerning the satisfiability problem will be discussed in Section 10.3.

### 10.1 Syntax and Semantics

Assuming a finite nonempty set of agents or players  $N$ , we define the syntax of Coalition Logic as follows. Given a set of atomic propositions  $\Phi_0$ , a formula  $\varphi$  can have the following syntactic form:

$$\varphi := \perp \mid p \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid [C]\varphi$$

where  $p \in \Phi_0$  and  $C \subseteq N$ . We define  $\top$ ,  $\wedge$ ,  $\rightarrow$  and  $\leftrightarrow$  as usual. In case  $C = \{i\}$ , we write  $[i]\varphi$  instead of  $[\{i\}]\varphi$ .

A *coalition frame* is a pair  $\mathcal{F} = (S, G)$  where  $S$  is a nonempty set of states (the universe) and  $G$  assigns to every state  $s \in S$  a strategic game form  $G(s) = (N, (\Sigma_i)_{i \in N}, o, S)$ . At state  $s$ , the game form  $G(s)$  represents the possible transitions based on the strategic choices of the players. A *coalition model* is a pair  $\mathcal{M} = (\mathcal{F}, V)$  where  $\mathcal{F}$  is a coalition frame and  $V : \Phi_0 \rightarrow \mathcal{P}(S)$  is the usual valuation function for the propositional letters. Given such a model, truth of a formula in a model at a state is defined as follows:

$$\begin{aligned} \mathcal{M}, s &\not\models \perp \\ \mathcal{M}, s &\models p && \text{iff } p \in \Phi_0 \text{ and } s \in V(p) \\ \mathcal{M}, s &\models \neg\varphi && \text{iff } \mathcal{M}, s \not\models \varphi \\ \mathcal{M}, s &\models \varphi \vee \psi && \text{iff } \mathcal{M}, s \models \varphi \text{ or } \mathcal{M}, s \models \psi \\ \mathcal{M}, s &\models [C]\varphi && \text{iff } \varphi^{\mathcal{M}} \in E_{G(s)}^{\alpha}(C) \end{aligned}$$

where  $\varphi^{\mathcal{M}} = \{s \in S \mid \mathcal{M}, s \models \varphi\}$ . Hence, a formula  $[C]\varphi$  holds at a state  $s$  iff coalition  $C$  is  $\alpha$ -effective for  $\varphi^{\mathcal{M}}$  in  $G(s)$ .

Coalition frames are essentially extensive game forms of *almost perfect* information. The only source of imperfect information is that players make choices simultaneously. After the choices are made at state  $s$ , a new state  $t$  results and the choices become common knowledge, before new (and possibly different) choices can be made. Note that coalition frames are game graphs rather than trees (though they can be unravelled into trees), and they contain no terminal states since every state is associated to a game.

It is possible to isolate (semantically and axiomatically) the class of coalition frames corresponding to extensive game forms of *perfect* information, see [63] for details.

Note that by Theorem 12, we can equivalently view a coalition frame  $\mathcal{F}$  as a pair  $(S, E)$ , where

$$E : S \rightarrow (\mathcal{P}(N) \rightarrow \mathcal{P}(\mathcal{P}(S)))$$

assigns to every state  $s \in S$  a monotonic,  $N$ -maximal and superadditive effectivity function  $E(s)$  (see Section 2.3). Using this formulation, we can then simply define  $\mathcal{M}, s \models [C]\varphi$  iff  $\varphi^{\mathcal{M}} \in E(s)(C)$ . From a logical point of view, this second formulation directly in terms of effectivity functions is preferable. It simplifies meta-theoretic reasoning, e.g. by immediately suggesting certain axioms of coalitional power, and it also demonstrates that coalition models are essentially neighbourhood models, providing a neighbourhood relation for every coalition of players. Neighbourhood models have been the standard semantic tool to investigate non-normal modal logics (see, e.g. [17]), and techniques used to provide complete axiomatisations for such logics can also be adapted to Coalition Logic.

The two extreme coalitions  $\emptyset$  and  $N$  are of special interest.  $[N]p$  expresses that some possible next state satisfies  $p$ , whereas  $[\emptyset]p$  holds if no agent needs to do anything for  $p$  to hold in the next state. Hence,  $[N]p$  corresponds to  $\Diamond p$  in standard modal logic whereas  $[\emptyset]p$  corresponds to  $\Box p$ . If  $|N| = 1$ , e.g.  $N = \{1\}$ , coalition models are just serial Kripke models, i.e., Kripke models where every state has at least one successor. In this case,  $[\emptyset]\varphi$  coincides with  $\Box\varphi$  and  $[1]\varphi$  with  $\Diamond\varphi$ .

## 10.2 Axiomatics

Let  $\mathbf{CL}_N$  denote the smallest set of formulas which contains all propositional tautologies together with all instances of the axiom schemas listed in Figure 20, and which is closed under the rules of Modus Ponens and Equivalence given below:

$$\frac{\varphi \quad \varphi \rightarrow \psi}{\psi} \qquad \frac{\varphi \leftrightarrow \psi}{[C]\varphi \leftrightarrow [C]\psi}$$

Note that axioms  $(\perp)$  and  $(\top)$  correspond to the two basic assumptions we made for effectivity functions in Definition 9. The remaining three axioms express the conditions of Theorem 12,  $N$ -maximality, monotonicity and superadditivity.

( $\perp$ )	$\neg[C]\perp$
( $\top$ )	$[C]\top$
(N)	$\neg[\emptyset]\neg\varphi \rightarrow [N]\varphi$
(M)	$[C](\varphi \wedge \psi) \rightarrow [C]\psi$
(S)	$([C_1]\varphi_1 \wedge [C_2]\varphi_2) \rightarrow [C_1 \cup C_2](\varphi_1 \wedge \varphi_2)$ where $C_1 \cap C_2 = \emptyset$

Figure 20. The axiom schemas of Coalition Logic

**THEOREM 45 ([63]).**  $\mathbf{CL}_N$  is sound and complete with respect to the class of all coalition models.

**KD** is the normal modal logic for reasoning about serial Kripke models. In the formulation closest to Coalition Logic, **KD** is the set of formulas containing all propositional tautologies, closed under the rules of Modus Ponens and Equivalence (for  $\Box$  only), and containing the axioms of Figure 21.

$\Diamond\varphi \leftrightarrow \neg\Box\neg\varphi$	$\Box(\varphi \wedge \psi) \leftrightarrow (\Box\varphi \wedge \Box\psi)$
$\Box\top$	$\Diamond\top$

Figure 21. Axioms of **KD**

The following result states that **KD** is precisely single-agent Coalition Logic. The result is the axiomatic analogue to our earlier observation that if  $|N| = 1$ , coalition models are simply serial Kripke models.

**THEOREM 46** ([63]). *Identifying  $[\emptyset]\varphi$  with  $\Box\varphi$  and  $[1]\varphi$  with  $\Diamond\varphi$ , we have  $\mathbf{KD} = \mathbf{CL}_{\{1\}}$ .*

The logic  $\mathbf{CL}_N$  is the most general and hence weakest coalition logic which has been investigated. The only assumption made is that at every state, the coalitional power distribution arises from a situation which can be modeled as a strategic game. Additional axioms can be added for characterising special kinds of strategic interaction. For example, in order to characterise extensive game forms of perfect information, one adds the axiom

$$[N]\varphi \rightarrow \bigvee_{i \in N} [i]\varphi,$$

expressing that everything which can be achieved at all can be achieved already by some individual. This axiom will enforce that at every state there is a single agent who can determine the next state independent of the other agents. Note that the converse implication can be derived in  $\mathbf{CL}_N$ .

Nash-consistent Coalition Logic [32] is another example of a logic stronger than  $\mathbf{CL}_N$ . By adding a further axiom to  $\mathbf{CL}_N$ , one can characterise the class of Nash-consistent coalition models, i.e., coalition models where the strategic game form associated to every state must have a Nash equilibrium under every possible preference profile. Nash-consistent models can be viewed as stable systems, in the sense that no matter what the agents' preferences are, there is a stable strategy profile, a profile for which individual deviation is irrational.

### 10.3 Complexity

We assume in our discussion of complexity that  $|N| > 1$ . As was mentioned, the two extreme coalitions  $\emptyset$  and  $N$  allow one to capture necessity and possibility. For this reason, the normal modal logic **KD** forms a fragment of Coalition Logic, thereby establishing a PSPACE lower bound for the basic Coalition Logic over general coalition models. As it turns out, this bound is tight, i.e., we have the following result.

**THEOREM 47** ([63]). *The complexity of the satisfiability problem for Coalition Logic is PSPACE-complete.*

Via the satisfiability problem, we can compare the complexity of reasoning about games of various different kinds. For instance, it turns out that restricting the class

of coalition models to perfect information models, the satisfiability problem remains PSPACE-complete. Hence, given a coalitional specification, finding an extensive game (form) of almost perfect information satisfying the specification is not harder nor simpler than finding an extensive game of perfect information.

Besides comparing reasoning over different classes of games, we can compare reasoning about groups to reasoning about individuals. Let the *individual fragment of Coalition Logic* be the set of formulas of Coalition Logic where all modalities only involve singleton coalitions, i.e.

$$\varphi := \perp \mid p \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid [i]\varphi$$

where  $p \in \Phi_0$  and  $i \in N$ . The individual fragment is strictly less expressive than full Coalition Logic, since the formula  $[C]p$  is in general not equivalent to any formula involving only singleton coalitions. More precisely, there is no formula  $\varphi$  of the individual fragment such that  $\varphi^{\mathcal{M}} = [C]p^{\mathcal{M}}$  for every coalition model  $\mathcal{M}$ .

**THEOREM 48 ([64]).** *The complexity of the satisfiability problem for the individual fragment of Coalition Logic is NP-complete.*

Hence, reasoning about individuals is simpler than reasoning about coalitions if and only if  $\text{NP} \neq \text{PSPACE}$ . For perfect information models, the complexity of the satisfiability problem for the individual fragment is not simpler, it remains PSPACE-complete.

## 11 ALTERNATING-TIME TEMPORAL LOGIC

Coalition Logic allows one to express strategic properties of multi-agent systems, where these systems are essentially modeled as extensive games of almost perfect information. The basic modal expression  $[C]\varphi$  states that coalition  $C$  has a joint strategy for ensuring  $\varphi$  in the next state. What is lacking are more expressive temporal operators which allow us to describe, e.g., that coalition  $C$  has a strategy for achieving  $\varphi$  some time in the future. In other words, we are looking for the strategic coalitional analogue of a rich temporal logic like CTL (see Chapter 11 and [19]). Alternating-time temporal logic (ATL) [3] is precisely this temporal extension of Coalition Logic.

As usual, we start by presenting the syntax and semantics of ATL in Section 11.1. After discussing a modelling example in Subsection 11.2, we discuss axiomatisation and complexity (Subsection 11.3) and end with some extensions of ATL in Subsection 11.4.

### 11.1 Syntax and Semantics

Given a set of atomic propositions  $\Pi$ , an ATL formula  $\varphi$  can have the following syntactic form:

$$\varphi := \perp \mid p \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \langle\langle C \rangle\rangle \bigcirc \varphi \mid \langle\langle C \rangle\rangle \Box \varphi \mid \langle\langle C \rangle\rangle \varphi_1 \mathcal{U} \varphi_2$$

where  $p \in \Pi$  and  $C \subseteq N = \{1, \dots, k\}$ , the set of all agents. We define  $\top$ ,  $\wedge$ ,  $\rightarrow$  and  $\leftrightarrow$  as usual. The formula  $\langle\langle C \rangle\rangle \bigcirc \varphi$  expresses that coalition  $C$  has a joint strategy for achieving  $\varphi$  at the next state. Thus,  $\langle\langle C \rangle\rangle \bigcirc \varphi$  corresponds to  $[C]\varphi$  in Coalition Logic.  $\langle\langle C \rangle\rangle \Box \varphi$  expresses that coalition  $C$  can cooperate to maintain  $\varphi$  forever (always in the future), and  $\langle\langle C \rangle\rangle \varphi_1 \mathcal{U} \varphi_2$  expresses that  $C$  can maintain  $\varphi_1$  until  $\varphi_2$  holds. In the standard way, we use  $\Diamond \varphi$  to abbreviate  $\top \mathcal{U} \varphi$ .

In the same way in which CTL can be extended to CTL\*, the language of ATL can be generalised to ATL\*. We simultaneously define the set of ATL\* *state formulas*  $\varphi$  and the set of ATL\* *path formulas*  $\psi$  as follows:

$$\begin{aligned}\varphi &:= \perp \mid p \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \langle\langle C \rangle\rangle\psi \\ \psi &:= \varphi \mid \neg\psi \mid \psi_1 \vee \psi_2 \mid \bigcirc\psi \mid \psi_1 \mathcal{U} \psi_2\end{aligned}$$

where  $p \in \Pi$  and  $C \subseteq N$ . Note that in ATL\*,  $\langle\langle C \rangle\rangle \Box \psi$  is expressible as  $\langle\langle C \rangle\rangle \neg(\bigwedge \mathcal{U} \neg\psi)$  which is not an ATL formula. In general, ATL is a proper fragment of ATL\*, containing only formulas where every temporal operator is immediately preceded by a cooperation modality. For  $|N| = 1$ ,  $\text{ATL} = \text{CTL}$  and  $\text{ATL}^* = \text{CTL}^*$ . Given that CTL is less expressive than CTL\*, it also follows that ATL is less expressive than ATL\*.

The semantics of Alternating-time Temporal Logic uses concurrent game structures, essentially the coalition models we discussed for Coalition Logic. A *concurrent game structure* is a tuple  $\mathcal{S} = (k, Q, \Pi, \pi, d, \delta)$ , where  $k$  is the number of players ( $N = \{1, \dots, k\}$ ),<sup>2</sup>  $Q$  is the set of states (usually assumed to be finite),  $\Pi$  is the set of atomic propositions, and  $\pi : Q \rightarrow \mathcal{P}(\Pi)$  is the valuation function. For every agent  $i \in N$  and every state  $q \in Q$ ,  $d_i(q) \geq 1$  gives the number of actions available to player  $i$  at state  $q$ . Hence, at state  $q$ , a *move vector*  $(j_1, \dots, j_k) \in D(q) = \{1, \dots, d_1(q)\} \times \dots \times \{1, \dots, d_k(q)\}$  corresponds to a joint action at state  $q$ . Finally, for each such move vector,  $\delta(q, j_1, \dots, j_k) \in Q$  is the transition function.

Like coalition models in Coalition Logic, different types of concurrent game structures correspond to natural classes of games. *Turn-based synchronous* game structures are extensive game forms of perfect information where only a single player can choose at each state. In *synchronous* game structures, the state space is the Cartesian product of the players' local state spaces. *Turn-based asynchronous* game structures involve a scheduler who determines the player who can choose the next state. Furthermore, fairness constraints can be added to these structures.

For two states  $q, q' \in Q$  and an agent  $i \in \Sigma$ , we say that state  $q'$  is a *successor* of  $q$  if there exists a move vector  $(j_1, \dots, j_k) \in D(q)$  such that  $\delta(q, j_1, \dots, j_k) = q'$ . A *computation* of  $\mathcal{S}$  is an infinite sequence of states  $\lambda = q_0, q_1, \dots$  such that for all  $u > 0$ , the state  $q_u$  is a successor of  $q_{u-1}$ . A computation  $\lambda$  starting in state  $q$  is referred to as a *q-computation*; if  $u \geq 0$ , then we denote by  $\lambda[u]$  the  $u$ 'th state in  $\lambda$ ; similarly, we denote by  $\lambda[0, u]$  and  $\lambda[u, \infty]$  the finite prefix  $q_0, \dots, q_u$  and the infinite suffix  $q_u, q_{u+1}, \dots$  of  $\lambda$  respectively.

A *strategy*  $f_i$  for an agent  $i \in N$  is a total function  $f_i$  mapping every finite nonempty sequence of states  $\lambda$  to a natural number such that if the last state of  $\lambda$  is  $q$ ,  $1 \leq f_i(\lambda) \leq d_i(q)$ . Given a set  $C \subseteq N$  of agents, and an indexed set of strategies  $F_C = \{f_i \mid i \in C\}$ , one for each agent  $i \in C$ , we define  $\text{out}(q, F_C)$  to be the set of possible computations that may occur if every agent  $a \in C$  follows the corresponding strategy  $f_a$ , starting when the system is in state  $q \in Q$ . Formally,  $\lambda = q_0, q_1, \dots \in \text{out}(q, F_C)$  iff  $q = q_0$  and for all  $m \geq 0$ , there exists a move vector  $(j_1, \dots, j_k) \in D(q_m)$  such that  $\delta(q_m, j_1, \dots, j_k) = q_{m+1}$  and for all  $i \in C$ ,  $j_i = f_i(\lambda[0, m])$ . The semantics of ATL\* can now be defined as follows.

<sup>2</sup>given  $|N| = k$ , we will sometimes use the equivalent representation  $\mathcal{S} = (N, Q, \Pi, \pi, d, \delta)$

For state formulas we define

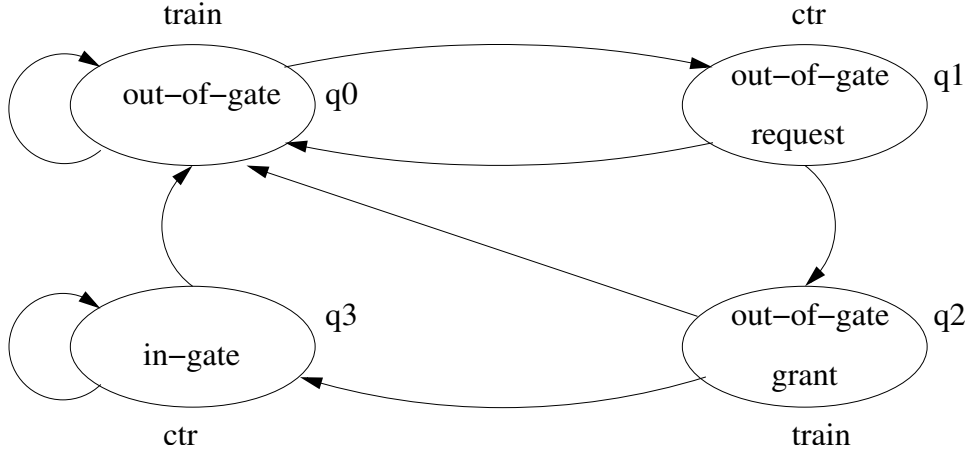
$$\begin{aligned}
\mathcal{S}, q &\not\models \perp \\
\mathcal{S}, q &\models p && \text{iff } p \in \Pi \text{ and } p \in \pi(q) \\
\mathcal{S}, q &\models \neg\varphi && \text{iff } \mathcal{S}, q \not\models \varphi \\
\mathcal{S}, q &\models \varphi_1 \vee \varphi_2 && \text{iff } \mathcal{S}, q \models \varphi_1 \text{ or } \mathcal{S}, q \models \varphi_2 \\
\mathcal{S}, q &\models \langle\langle C \rangle\rangle\psi && \text{iff } \exists F_C \forall \lambda \in \text{out}(q, F_C) : \mathcal{S}, \lambda \models \psi,
\end{aligned}$$

and for path formulas we define

$$\begin{aligned}
\mathcal{S}, \lambda &\models \varphi && \text{iff } \mathcal{S}, \lambda[0] \models \varphi, \text{ where } \varphi \text{ is a state formula} \\
\mathcal{S}, \lambda &\models \neg\psi && \text{iff } \mathcal{S}, \lambda \not\models \psi \\
\mathcal{S}, \lambda &\models \psi_1 \vee \psi_2 && \text{iff } \mathcal{S}, \lambda \models \psi_1 \text{ or } \mathcal{S}, \lambda \models \psi_2 \\
\mathcal{S}, \lambda &\models \bigcirc\psi && \text{iff } \mathcal{S}, \lambda[1, \infty] \models \psi \\
\mathcal{S}, \lambda &\models \psi_1 \mathcal{U} \psi_2 && \text{iff } \exists m \geq 0 : (\mathcal{S}, \lambda[m, \infty] \models \psi_2 \text{ and } \forall l (0 \leq l < m \Rightarrow \mathcal{S}, \lambda[l, \infty] \models \psi_1))
\end{aligned}$$

## 11.2 An Example

The following example from [3] presents a turn-based synchronous game structure modeling a simple train system involving a train and a controller.



Formally, the concurrent game structure  $\mathcal{S} = (k, Q, \Pi, \pi, d, \delta)$  consists of  $\Pi = \{\text{out-of-gate}, \text{in-gate}, \text{request}, \text{grant}\}$ ,  $N = \{\text{train}, \text{ctr}\}$  and  $Q = \{q_0, q_1, q_2, q_3\}$ , with valuation function as given (e.g.  $\pi(q_0) = \{\text{out-of-gate}\}$ ). The concurrent game structure is turn-based synchronous, so if we take the train as player 1 and the controller as player 2, we have, e.g.,  $d_1(q_0) = 2$  and  $d_2(q_0) = 1$ . The transition function at state  $q_0$  is then described by  $\delta(q_0, 1, 1) = q_0$  and  $\delta(q_0, 2, 1) = q_1$ .

ATL can be utilised to describe properties of this model. For instance, the formula  $\langle\langle \emptyset \rangle\rangle \Box (\text{in-gate} \rightarrow \langle\langle \text{ctr} \rangle\rangle \bigcirc \text{out-of-gate})$  expresses that whenever the train is in the gate, the controller can force it out immediately. Similarly,  $\langle\langle \emptyset \rangle\rangle \Box (\text{out-of-gate} \rightarrow \langle\langle \text{ctr}, \text{train} \rangle\rangle \Diamond \text{in-gate})$  states that whenever the train is out of the gate, it can cooperate with the controller to enter eventually. As a final example,

$$\langle\langle \emptyset \rangle\rangle \Box (\text{out-of-gate} \rightarrow \langle\langle \text{train} \rangle\rangle \Diamond (\text{request} \wedge \langle\langle \text{ctr} \rangle\rangle \Diamond \text{grant} \wedge \langle\langle \text{ctr} \rangle\rangle \Box \neg \text{grant}))$$

expresses that whenever the train is out of the gate, it can eventually send a request to enter, and the controller can then either grant it eventually or not. All these formulas are valid in the given game structure.



### 11.3 Axiomatics and Complexity

The axiomatisation of ATL given below is an extension of the axiomatisation given for Coalition Logic. The next-time  $\bigcirc$  operator is characterised by the axioms of Coalition Logic, and the long-term temporal operators  $\Box$  and  $\mathcal{U}$  are captured using two fixpoint axioms each.

Formally, **ATL** is the smallest set of formulas which contains all propositional tautologies together with all instances of the axiom schemas listed in Figure 22, and which is closed under the inference rules of Modus Ponens, Monotonicity and Necessitation given below:

$$\frac{\varphi \quad \varphi \rightarrow \psi}{\psi} \qquad \frac{\varphi \rightarrow \psi}{\langle\langle C \rangle\rangle \bigcirc \varphi \rightarrow \langle\langle C \rangle\rangle \bigcirc \psi} \qquad \frac{\varphi}{\langle\langle \emptyset \rangle\rangle \Box \varphi}$$

For the case of  $\Box$ ,  $FP \Box$  states that  $\langle\langle C \rangle\rangle \Box \varphi$  is a fixpoint of the operator  $F(X) = \varphi \wedge \langle\langle C \rangle\rangle \bigcirc X$ , and  $GFP \Box$  states that it is the greatest fixpoint of  $F(X)$ . Analogously for the least fixpoint with  $\mathcal{U}$ .

( $\perp$ )	$\neg \langle\langle C \rangle\rangle \bigcirc \perp$
( $\top$ )	$\langle\langle C \rangle\rangle \bigcirc \top$
( <b>N</b> )	$(\neg \langle\langle \emptyset \rangle\rangle \bigcirc \neg \varphi \rightarrow \langle\langle N \rangle\rangle \bigcirc \varphi)$
( <b>S</b> )	$(\langle\langle C_1 \rangle\rangle \bigcirc \varphi_1 \wedge \langle\langle C_2 \rangle\rangle \bigcirc \varphi_2) \rightarrow \langle\langle C_1 \cup C_2 \rangle\rangle \bigcirc (\varphi_1 \wedge \varphi_2) \text{ where } C_1 \cap C_2 = \emptyset$
( $FP \Box$ )	$\langle\langle C \rangle\rangle \Box \varphi \leftrightarrow \varphi \wedge \langle\langle C \rangle\rangle \bigcirc \langle\langle C \rangle\rangle \Box \varphi$
( $GFP \Box$ )	$\langle\langle \emptyset \rangle\rangle \Box (\psi \rightarrow (\varphi \wedge \langle\langle C \rangle\rangle \bigcirc \psi)) \rightarrow \langle\langle \emptyset \rangle\rangle \Box (\psi \rightarrow \langle\langle C \rangle\rangle \Box \varphi)$
( $FP \mathcal{U}$ )	$\langle\langle C \rangle\rangle \varphi_1 \mathcal{U} \varphi_2 \leftrightarrow \varphi_2 \vee (\varphi_1 \wedge \langle\langle C \rangle\rangle \bigcirc \langle\langle C \rangle\rangle \varphi_1 \mathcal{U} \varphi_2)$
( $LFP \mathcal{U}$ )	$\langle\langle \emptyset \rangle\rangle \Box ((\varphi_2 \vee (\varphi_1 \wedge \langle\langle C \rangle\rangle \bigcirc \psi)) \rightarrow \psi) \rightarrow \langle\langle \emptyset \rangle\rangle \Box (\langle\langle C \rangle\rangle \varphi_1 \mathcal{U} \varphi_2 \rightarrow \psi)$

Figure 22. The axiom schemas of ATL

**THEOREM 49** ([26]). **ATL** is sound and complete with respect to the class of all concurrent game structures.

The complexity of model checking ATL formulas has been investigated in [3]. As with Game Logic, given a formula  $\varphi$  and a finite model  $\mathcal{S}$ , we are interested in the complexity of determining the states of  $\mathcal{S}$  where  $\varphi$  holds. The results are for general concurrent game structures.

**THEOREM 50** ([3]). Given an ATL formula  $\varphi$  and a concurrent game structure  $\mathcal{S}$  with  $m$  transitions, model checking can be done in time  $O(m \times |\varphi|)$ . For  $ATL^*$  formulas, model checking is  $2EXPTIME$ -complete.

The complexity of the satisfiability problem has been investigated only more recently. At the time of writing, only the complexity of ATL has been determined. Let us say that an ATL-formula  $\varphi$  is over a set of agents  $N$  if all coalitions mentioned in  $\varphi$  are subsets of  $N$ . Moreover, we say that a concurrent game structure  $\mathcal{S} = (M, Q, \Pi, \pi, d, \delta)$  is over  $N$  if  $M = N$  (see Footnote 2)

**THEOREM 51** ([103, 26]). *Let  $N$  be a finite set of players. Then, the complexity of the satisfiability problem for ATL-formulas over  $N$  with respect to concurrent game structures over  $N$  is EXPTIME-complete.*

To demonstrate that Theorem 51 marks ongoing work in a lively research area, note that the decision procedure of [103, 26] is 2EXPTIME if the set of agents  $N$  is not fixed in advance. This gives rise to the following two related questions:

1. Is the following problem in EXPTIME: Given an ATL-formula  $\varphi$ , is  $\varphi$  satisfiable in an ATL-model over a set of agents containing at least the agents occurring in  $\varphi$ ?
2. Is the following problem in EXPTIME: Given a set of agents  $N$  and an ATL-formula  $\varphi$  over  $N$ , is  $\varphi$  satisfiable in an ATL-model over  $N$ ?

Positive answers to both questions are given in [106].

#### 11.4 Extensions: $\mu$ -calculus and imperfect information

CTL and CTL\* are both subsumed by the very expressive  $\mu$ -calculus. Similarly for ATL and ATL\*, one can develop an analogous *alternating  $\mu$ -calculus* with general fixpoint expressions. Formally, we have a set of propositional variables  $V$ , and formulas

$$\varphi := \perp \mid p \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid X \mid \langle\langle C \rangle\rangle \bigcirc \varphi \mid \mu X. \varphi$$

where  $X \in V$  and in  $\mu X. \varphi$ ,  $X$  occurs free only under an even number of negations in  $\varphi$ . The semantics of the fixpoint operator is defined as  $(\mu X. \varphi)^S = \bigcap \{Q_0 \subseteq Q \mid \varphi[X := Q_0]^S \subseteq Q_0\}$ . Corresponding to the least fixpoint  $\mu X$  there is also a greatest fixpoint  $\nu X$  defined as  $\nu X. \varphi = \neg \mu X. \neg \varphi$ . The alternating  $\mu$ -calculus subsumes both ATL and ATL\*. For ATL, note that  $\langle\langle C \rangle\rangle \Box \varphi = \nu X. \varphi \wedge \langle\langle C \rangle\rangle \bigcirc X$  and  $\langle\langle C \rangle\rangle \varphi \mathcal{U} \psi = \mu X. \psi \vee (\varphi \wedge \langle\langle C \rangle\rangle \bigcirc X)$ . In fact, it can be shown that the alternating  $\mu$ -calculus is strictly more expressive than ATL\*. As in the standard modal  $\mu$ -calculus and in Game Logic, the complexity of model checking depends on the alternation depth of a formula, i.e., the nesting depth of alternating least and greatest fixpoints.

Various attempts have been made to introduce imperfect information into concurrent game structures. Since at the time of writing this problem is still under discussion, we restrict ourselves here to a rough sketch of some of the difficulties. In [3], a set of observable propositions is associated with every player. This extension of ATL introduces quite a few new complexities, syntactically as well as semantically, and this is witnessed by the result that model-checking becomes undecidable. Syntactically, not all ATL expressions make sense anymore, since, e.g.,  $\langle\langle C \rangle\rangle \Box p$  presupposes that the cooperative goal  $p$  is actually observable by the members of  $C$ . Semantically, a player's strategy has to be restricted in such a way that it can only influence propositions observable by that player. The difficulties which arise when trying to extend ATL to incomplete information also become visible in an alternative approach explored in [92]. Here, concurrent game structures are augmented with an epistemic accessibility relation  $\sim_i$  for each player  $i$ . In contrast to the previous approach, the language of ATL is also extended with an epistemic  $K_i$  knowledge operator with its standard definition. The resulting *Alternating-time Temporal Epistemic Logic* (ATEL) can express properties from a variety of domains, e.g. confidentiality properties like  $\langle\langle \{1, 2\} \rangle\rangle (\neg K_3 p \mathcal{U} K_2 p)$ . The semantics of ATL, however,

cannot simply be left unchanged. At a state  $q$ , player 1 may have some strategy  $s_1$  to eventually achieve  $\varphi$ . But if he cannot distinguish  $q$  from  $q'$  where only a different strategy  $s_2$  achieves  $\varphi$ , player 1 does *not* have a strategy at  $q$  (as defined for imperfect information games) for achieving  $\varphi$ . More precisely, player 1 has a strategy only *de dicto*, but not *de re*, and it is the *de re* strategies which game theory is interested in [39]. A further approach is presented in [72].

## 12 CONCLUSION

What can modal logic contribute to the study of strategic interaction? Four answers, not mutually exclusive, suggest themselves to us. First, logic contributes a more abstract and hence more general perspective on games. As we have seen in Sections 3 and 6, games can essentially be viewed as special kinds of Kripke models, with move relations and possibly epistemic uncertainty relations to model players' knowledge. In this way, certain game-theoretic notions such as perfect recall turn out to be special cases of general logical axioms (cf. 6.1) which have been investigated in more generality in modal logic. Similarly, game logic (Section 9) presents games as generalisations of programs, providing a semantics which is general enough to study the difference between 2-player games and 1-player games (i.e., programs). This more general perspective provided by logic also raises new questions. In the case of game logic, for instance, we may ask what operations on games suffice to build all games in a particular class of games. More generally, to the best of our knowledge, operations on games are rarely investigated in game theory, while a computational logic perspective naturally suggests such an investigation.

Second, there is the analysis of players' knowledge and beliefs in games. While much of the game-theoretic analysis of interactive epistemics has been independent of developments in epistemic logic (Section 4), the models employed are essentially the same (information partitions or Kripke models satisfying the S5 axioms), as is the notion of common knowledge. Developments in update logics (Section 7) and belief revision (see Chapter 21 of this handbook) have shifted the logical focus from an analysis of static epistemic situations to epistemic dynamics, without a doubt important in the analysis of games of imperfect information. Furthermore, the epistemic foundation of solution concepts (Section 8) translates solution concepts into the language of the epistemic logician, while the language itself remains hidden from view. More precisely, while epistemic logic rests on the link between the semantic model of knowledge and the formal language used to describe it, game theory has mainly focused on the semantic model exclusively. Only more recently have syntactic approaches to the epistemic foundations of solution concepts been advanced.

The issue of syntax or formal language is, in fact, a third way in which logic can raise new issues in strategic interaction. While many game theorists may find the logician's insistence on syntax cumbersome and unnecessary, it is precisely the interplay between syntax and semantics that the logician is interested in. In general, decision makers use a certain language and conceptual apparatus to reason about the situation at hand. As a consequence, one would suspect that game theoretic models and solution concepts would be language dependent (see also [71]). More specifically, we may be interested whether a particular logical language is rich enough to define a particular solution concept such as the Nash equilibrium (see, e.g., Theorem 15 in Section 3). As mentioned above, we may wonder whether a certain set of game operations suffices to construct all games of a

particular class. Or we may be interested in the computational complexity of reasoning about information update in games. For all these questions, a syntactic approach may help.

Fourth, due to its formal language, logic becomes important when it comes to the specification and verification of multi-agent systems. Logics like Alternating-time Temporal Logic (Section 11) have been devised as a way to specify properties of systems of interacting agents. By analysing the complexity of model checking, we find out how complex certain game-theoretic properties are to verify for a given game or game form. Hence, logic is also useful when applying game theory to complex games played by artificial agents, and hence modal logic can serve as a tool of computation for difficult real or artificial life games.

While we hope to have given some insights into how modal logic may enrich game theory, we should point out again that there are many cases where game theory can be useful for modal logic. Maybe the most interesting example is Independence-Friendly Modal Logic (IFML) [78]. In IFML, we consider formulas like  $\Box\Diamond^*p$ , where the modal diamond  $\Diamond^*$  is independent of the box. Such a formula will be true just in case the diamond successor satisfying  $p$  can be chosen independently from the earlier box successor, in other words, there needs to be a uniform diamond successor satisfying  $p$  for all earlier box successors. As this example suggests, the semantics of IFML can be formulated in terms of imperfect information games. IFML is interesting from a logical point of view since it is more expressive than standard modal logic, due to its ability to express certain weak confluence properties.

The area of modal logic and games is active and in full development. Due to upcoming technologies like on-line auctions and e-voting, researchers apply a new range of tools and techniques to ask and settle ‘standard’ questions of logicians and computer scientists regarding the specification, verification and synthesis of interactive systems or mechanisms. In the ‘classical’ areas of game theory and social choice theory, this logical work generates both interesting results regarding formalisation and computational complexity, and at the same time new and sometimes even philosophical questions about the nature of games, or more generally, interaction.

## ACKNOWLEDGEMENTS

We are thankful to Johan van Benthem, Giacomo Bonanno, Govert van Drimmelen, Valentin Goranko and Michael Wooldridge, who gave valuable suggestions on earlier drafts of this chapter.

## BIBLIOGRAPHY

- [1] J. Abdou and H. Keiding. *Effectivity Functions in Social Choice*. Kluwer, 1991.
- [2] C.E. Alchourrón, P. Gärdenfors, and D. Makinson. On the logic of theory change: partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50:510–530, 1985.
- [3] R. Alur, T. Henzinger, and O. Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49:672–713, 2002.
- [4] R.J. Aumann. Game theory. In J. Eatwell, M. Milgate, and P. Newman, editors, *Game Theory*, The New Palgrave, pages 1–54. Macmillan, 1997.
- [5] R.J. Aumann. Interactive epistemology I: Knowledge. *International Journal of Game Theory*, 28:263–300, 1999.

- [6] R.J. Aumann and A. Brandenburger. Epistemic conditions for Nash equilibrium. *Econometrica*, (63):1161–1180, 1995.
- [7] A. Baltag and L. Moss. Logics for epistemic programs. *Synthese*, 139(2):165–224, 2004. In the section ‘Knowledge, Rationality and Action’.
- [8] A. Baltag, L. Moss, and S. Solecki. The logic of public announcements, common knowledge, and private suspicions. In I. Gilboa, editor, *Proceedings of TARK*, pages 43–56, 1998.
- [9] P. Battigalli and G. Bonanno. Recent results on belief, knowledge and the epistemic foundations of game theory. *Research in Economics*, 53:149–225, 1999.
- [10] D. Berwanger and G. Lenzi. The variable hierarchy of the  $\mu$ -calculus is strict. In *STACS 2005*, pages 97–109, 2005.
- [11] K. Binmore. *Fun and Games – A Text on Game Theory*. W.C. Heath Co., 1992.
- [12] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001.
- [13] G. Bonanno. Modal logic and game theory: Two alternative approaches. *Risk Decision and Policy*, 7(3):309–324, 2002.
- [14] G. Bonanno. A characterisation of von neumann games in terms of memory. *Synthese*, pages 281–295, 2004. In the section ‘Knowledge, Rationality and Action’.
- [15] G. Bonanno. A simple modal logic for belief revision. *Synthese*, pages 193–228, 2005. In the section ‘Knowledge, Rationality and Action’.
- [16] J.R. Büchi. Using determinacy to eliminate quantifiers. In M. Karpinski, editor, *Fundamentals of Computation Theory*, volume 56 of *Lecture Notes in Computational Science*, pages 367–378. Springer, 1977.
- [17] B. Chellas. *Modal Logic - An Introduction*. Cambridge University Press, 1980.
- [18] B. de Bruin. *Explaining Games– On the Logic of Game Theoretic Explanations*. PhD thesis, University of Amsterdam, Amsterdam, 2004. ILLC series DS-2004-03.
- [19] E.A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, pages 995–1072. MIT Press, 1990. Volume B.
- [20] K. Engelhardt, R. van der Meyden, and Y. Moses. Knowledge and the logic of local propositions. In *Proceedings of the 1998 Conference on Theoretical Aspects of Reasoning about Knowledge (TARK98)*, pages 29–41, 1998.
- [21] R. Fagin, J.Y. Halpern, Y. Moses, and M.Y. Vardi. *Reasoning About Knowledge*. The MIT Press, Cambridge, MA, 1995.
- [22] P. Gammie and R. van der Meyden. MCK: Model checking the logic of knowledge. In R. Alur and D. Peled, editors, *Proceedings of the 16th International conference on Computer Aided Verification (CAV 2004)*, pages 479–483. Springer, 2004.
- [23] J.D. Gerbrandy and W. Groeneveld. Reasoning about information change. *Journal of Logic, Language, and Information*, 6:147–169, 1997.
- [24] A. Gibbard. A Pareto-consistent libertarian claim. *Journal of Economic Theory*, 7(4):388–410, 1974.
- [25] V. Goranko. The basic algebra of game equivalences. *Studia Logica*, 75(2):221–238, 2003.
- [26] V. Goranko and G. van Drimmelen. Decidability and complete axiomatization of the alternating-time temporal logic. *Theoretical Computer Science*, 353(1-3):93–117, 2006.
- [27] Raymond Greenlaw, H. James Hoover, and Walter L. Ruzzo. *Limits to Parallel Computation: P-Completeness Theory*. Oxford University Press, Oxford & New York, 1995.
- [28] J. Halpern. A computer scientist looks at game theory. *Games and Economic Behavior*, 45(1):114–131, 2003.
- [29] J. Halpern and M. Vardi. The complexity of reasoning about knowledge and time. In *Proceedings 18th ACM Symposium on Theory of Computing*, pages 304–315, 1986.
- [30] J. Halpern and M. Vardi. The complexity of reasoning about knowledge and time, I: Lower bounds. *Journal of Computer and System Sciences*, 38(1):195–237, 1989.
- [31] J.Y. Halpern, R. van der Meyden, and M.Y. Vardi. Complete axiomatizations for reasoning about knowledge and time. *SIAM Journal on Computing*, 33(3):674–703, 2004.
- [32] H. Hansen and M. Pauly. Axiomatising Nash-consistent coalition logic. In S. Flesca, S. Greco, N. Leone, and G. Ianni, editors, *Logics in Artificial Intelligence*, LNAI 2424, pages 394–406. Springer, 2002.
- [33] D. Harel. Dynamic logic. In D. Gabbay and F. Guenther, editors, *Handbook of Philosophical Logic*, pages 497–604. Kluwer, 1984.
- [34] D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic*. The MIT Press: Cambridge, MA, 2000.
- [35] B.P. Harrenstein, W. van der Hoek, J.-J.Ch. Meyer, and C. Witteveen. A modal characterization of Nash equilibrium. *Fundamenta Informaticae*, 57(2–4):281–321, 2003.
- [36] J. Hintikka. *Knowledge and Belief*. Cornell University Press, 1962. Republished King’s College London Publications, 2005.
- [37] J. Hintikka. *Logic, Language Games and Information*. Oxford University Press, 1973.

- [38] W. van der Hoek and M. Wooldridge. Model checking knowledge and time. In D. Bošnački and S. Leue, editors, *Model Checking Software, Proceedings of SPIN 2002 (LNCS Volume 2318)*, pages 95–111. Springer, 2002.
- [39] W. Jamroga and W. van der Hoek. Agents that know how to play. *Fundamenta Informaticae*, 63(2-3):185–219, 2004.
- [40] E. Kindler. Safety and liveness properties: A survey. *Bulletin of the European Association for Theoretical Computer Science*, 53:268–272, 1994.
- [41] D. Kozen. Results on the propositional  $\mu$ -calculus. *Theoretical Computer Science*, 27:333–354, 1983.
- [42] H.W. Kuhn. Extensive games and the problem of information. In H.W. Kuhn and A.W. Tucker, editors, *Contributions to the Theory of Games, Volume II*, Annals of Mathematics Studies 28, pages 193–216. Princeton University Press, 1953.
- [43] L. Lamport. Proving the correctness of multiprocess programs. *IEEE Transactions on Software Engineering*, SE-3(2), 1977.
- [44] M. Lange and C. Stirling. Model checking games for branching time logics. *Journal of Logic and Computation*, 12(4):623–639, 2002.
- [45] Logic and the foundations of game and decision theory (LOFT). <http://www.econ.ucdavis.edu/faculty/bonanno/loft.html>.
- [46] A. Lomuscio and F. Raimondi. MCMAS: A model checker for multi-agent systems. In *Proceedings of TACAS06*, volume 3920, pages 450–454. Springer Verlag, 2006.
- [47] A. Lomuscio, R. van der Meyden, and M. Ryan. Knowledge in multiagent systems: initial configurations and broadcast. *ACM Transactions on Computational Logic (TOCL)*, 1(2):247–284, 2000.
- [48] A.R. Lomuscio. *Knowledge Sharing among Ideal Agents*. PhD thesis, University of Birmingham, Birmingham, UK, 1999.
- [49] J.-J.Ch. Meyer and W. van der Hoek. *Epistemic Logic for AI and Computer Science*. Cambridge University Press, Cambridge, England, 1995.
- [50] M. Mizutani, Y. Hiraide, and H. Nishino. Computational complexity to verify the instability of effectivity function. *International Journal of Game Theory*, 22:225–239, 1993.
- [51] R.C. Moore. A formal theory of knowledge and action. In J.F. Allen J., Hendler, and A. Tate, editors, *Readings in Planning*, pages 480–519. Morgan Kaufmann Publishers, San Mateo, CA, 1990.
- [52] H. Moulin. *The Strategy of Social Choice*. North-Holland, 1983.
- [53] J. Nash. Non-cooperative games. *Annals of Mathematics*, 54:286–295, 1951.
- [54] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. John Wiley and Sons, 1944.
- [55] D. Niwiński.  $\mu$ -calculus via games. In J.C. Bradfield, editor, *Computer Science Logic*, volume 2471 of *Lecture Notes in Computer Science*, pages 27–43. Springer, 2002.
- [56] M. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, 1994.
- [57] C. Papadimitriou. Algorithms, games, and the Internet. In *33rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 749–753, 2001.
- [58] R. Parikh. Propositional logics of programs: New directions. In M. Karpinski, editor, *Foundations of Computation Theory*, LNCS 158, pages 347–359. Springer, 1983.
- [59] R. Parikh. The logic of games and its applications. In M. Karpinski and J. van Leeuwen, editors, *Topics in the Theory of Computation*, Annals of Discrete Mathematics 24, pages 111–140. Elsevier, 1985.
- [60] R. Parikh and R. Ramanujam. *Distributed Processes and the Logic of Knowledge*, pages 256–268. Number 193 in *Lecture Notes in Computer Science*. Springer, 1985.
- [61] M. Pauly. From programs to games: Invariance and safety for bisimulation. In P. Clote and H. Schwichtenberg, editors, *Computer Science Logic*, LNCS 1862. Springer, 2000.
- [62] M. Pauly. *Logic for Social Software*. PhD thesis, University of Amsterdam, 2001.
- [63] M. Pauly. A modal logic for coalitional power in games. *Journal of Logic and Computation*, 12(1):149–166, 2002.
- [64] M. Pauly. On the complexity of coalitional reasoning. *International Game Theory Review*, 4(3):237–254, 2002.
- [65] M. Pauly and R. Parikh. Game logic - an overview. *Studia Logica*, 75(2):165–182, 2003.
- [66] B. Peleg. Effectivity functions, game forms, games and rights. *Social Choice and Welfare*, 15:67–80, 1998.
- [67] M. Piccione and A. Rubinstein. On the interpretation of decision problems with imperfect recall. *Games and Economic Behaviour*, 20(1):3–24, 1997.
- [68] J.A. Plaza. Logics of public communications. In M.L. Emrich, M.S. Pfeifer, M. Hadzikadic, and Z.W. Ras, editors, *Proceedings of the 4th International Symposium on Methodologies for Intelligent Systems*, pages 201–216, 1989.
- [69] W.V.O. Quine. Quantifiers and propositional attitudes. In L. Linsky, editor, *Reference and Modality*, pages 101–111. Oxford University Press, London, 1971.