A Model for Epistemic Games

Tanja Hötte

14th January 2004

Acknowledgments

In the first place I want to thank Johan van Benthem for his continual advice and for stimulating discussions of the thesis. Without him, it would never have been finished. Many thanks also to Boudwijn the Bruin for his valuable feedback, in particular on early drafts of the thesis. I am indebted to Jill Cirasella, Heike Hötte, Miriam Ossa and especially Rainer Osswald for proofreading, of which of course none is responsible for any remaining errors and fallacies.

Contents

Chapter 1

Introduction

What makes a game a game and thus a source of joy, mind-boggling tasks, tension and loss of time is uncertainty about what will happen while playing. If there is no uncertainty, there is not much use in playing at all, exactly because you know the outcome already. In games with a goal and more than one player, the most prominent uncertainty is about other players' behavior. This kind of uncertainty is dealt with by ascribing players some kind of rational behavior and then calculating what they will most likely do (assuming they really do act rationally).

There is epistemic uncertainty in every game, but of different quality.

There need not just be uncertainty about the future; the past is as good a source for uncertainty. If a move was made in secret or a player did not pay attention, he ends up being not sure which of several options were realized; to speak in terms of game trees: there is more than one state that he considers possible to be the actual state the game is in. Uncertainty of this kind is modeled e.g. in imperfect information games.[1]

There is no guarantee that players know which kind of game they are playing, perfect or imperfect information: a player might be convinced that he is playing a perfect information game – until he is confronted with an imperfect information situation that ruins his strategy. Or he might not be sure whether imperfect information (for him or for his opponents) will occur later in the game.

So far, all uncertainties have been restricted to player behavior - but what if the game itself is unclear? Maybe a player did not listen when the rules were explained, so now he does not know whether the game goes for ten or for sixteen rounds. Or he might have missed some extra moves he is allowed to make. It might even be part of the game that he is not told all his options.

Last but not least there are pure epistemic uncertainties: a player can only guess what other players think about the game. He does not know whether he guesses the other players' intentions correctly or whether they know where they are in which game.

[1] See appendix B for a definition of perfect and imperfect information games, and for example van Benthem [2] their motivation.

And, recursively, he does not know what other players think he thinks.

So how to model all those uncertainties?

Imperfect Information Games

As an illustration for modeling imperfect information, regard the game represented in figure 1.1. The game is represented by a process graph, and labels on non-terminal states indicate whose turn it is at that state. At the leaves, the utility for each player is indicated (in this case, just win or lose, but you can have tuples giving more fine-scaled outcomes as well). If a player cannot distinguish one state from the other, those states are said to be in the same information set. Seen from the point of view of the epistemic relation, this relation is an equivalence relation and information sets are just the equivalence classes. Information sets containing more than one state are indicated by dotted lines, indexed for the player who cannot distinguish between the connected states (in the example, E knows that A has made a move, but she does not know which one). In the perfect information game, without the dotted line, E had a winning strategy. Now, the move that would make her win in one situation would make her lose in the other - she loses her winning strategy to ignorance.

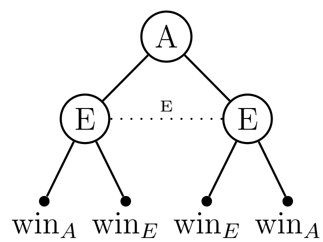

Figure 1.1: An imperfect information game

Let us look at a slightly more complex imperfect information game (figure 1.2) - the only difference to the game in figure 1.1 is that now A and E have more differentiated outcomes. Here, (n, m) means that A gets the value n and E gets the value m. The goal of the players is assumed to be to get the highest utility for themselves – they are taken not to care about what others get. In this game, E can still pursue a sensible strategy: by choosing right she can avoid the worst possible outcome for her (i.e. ending in the leaf labeled (1,1)).

But this is not all that can be said about the epistemic states of the players: Does A know that E will not know his move? Maybe he knows in advance that she will not pay attention and ask him what move he has made, so he can lie to her. In the example, this would lead to two different strategies: In case he thinks E will have perfect information as well, he chooses right, since $(4, 3)$ is the best node for her, too. In case he knows that E has imperfect information and will go for a uniform strategy, he chooses left. And

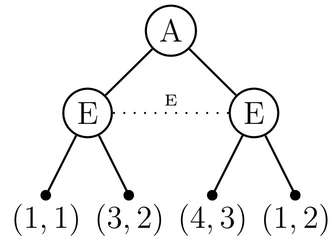

Figure 1.2: An imperfect information game with values

what if A is not sure in which kind of game he is, perfect or imperfect? You cannot indicate information of this kind with information sets.

So, let us look at a more refined way to draw pictures.

Perfect Or Imperfect?

A first step to generalize imperfect information games is to give up using an equivalence relation for the epistemic relation. I will indicate a non-equivalence-relation by a dashed arrow, meaning that in the state where the arrow starts, the player whose label is on the arrow thinks it possible to be in the state the arrow points at. But for clarity's sake I will go on using dotted lines where there is an equivalence relation in order to avoid having to draw a whole bunch of arrows. For the same reason, I just add epistemic lines to states I will talk about in my example.[2] Note that at a state where just one dashed arrow goes out for a player, that player is convinced to be in the state the arrow points at – there is no reflexivity presupposed.

As a different kind of relaxation of restrictions, we can allow for more than one game tree to be in the picture. The root of the original game, the game that is actually played, is indicated by a diamond. Any other game tree represents epistemic alternatives to the real game, i.e. games the players believe they are or might be playing.

The game with imperfect information for E where A is not sure which kind of game he plays can thus be represented as in figure 1.3: at the root, A thinks both the actual game and its perfect information version possible. As argued above, A has a strategy for each of the situations. But they conflict, so he is deprived of a criterion to choose an action.

Uncertainty about the Structure

In the last example, we stuck to a feature of imperfect information games: there is just one tree to regard (even if that tree showed up in two copies as in figure 1.3). But

[2]If you want to see all epistemic information in the game represented in figure 1.3 you have to add reflexive dashed arrows to each E-state and to each leaf, labeled with A and E and exchange the dotted line by two arrows going back and forth, both labeled with E.

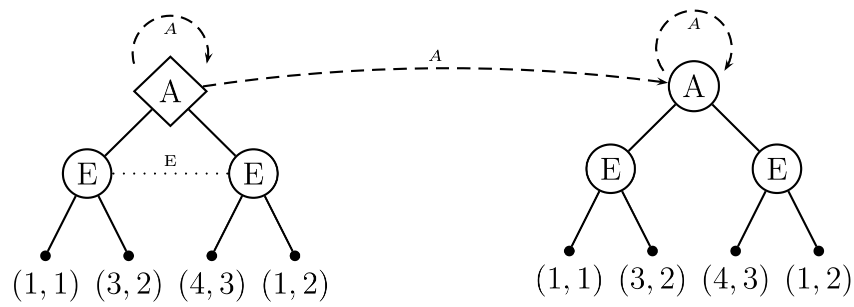

Figure 1.3: Perfect or Imperfect?

why keep this restriction? Players need not know what the whole game looks like in order to play. For example, if a player just sees the next two steps he could be willing to take the risk to go into 'unknown territory' if the options he can see are not very attractive. Or a player could not have payed attention the whole time while the game was explained, so he missed a certain part of the structure. The other way around, the organizer of the game might be lazy and neglect to fully explain all possibilities until the run of the game forces him to.

In figure 1.4 a simple example of this situation is modeled. The dots in the right tree indicate that A has no idea how the game goes on. Or stated the other way around, he thinks possible every game that agrees on the first two levels with what he knows. So the tree with the dotted line stands for not just one tree the player thinks possible but for all trees starting out at this tree does – anything is possible at that dotted part, so for A it is like knowing nothing at all about his 'dotted future'.

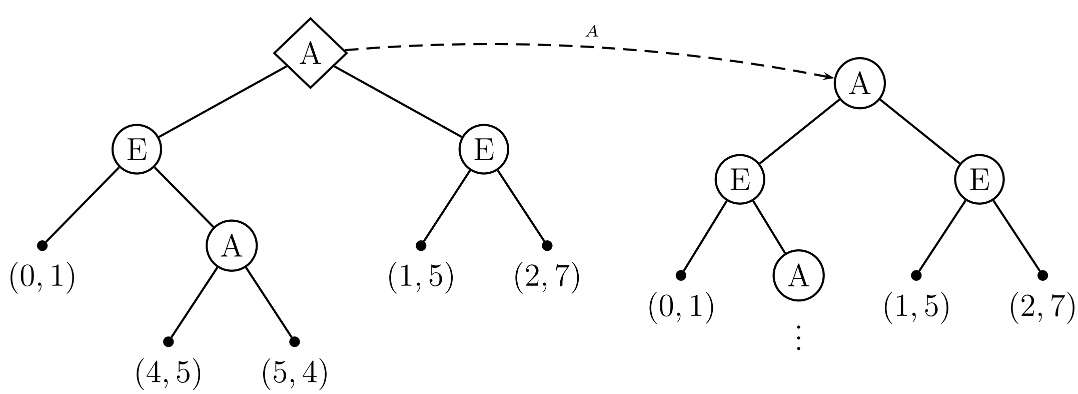

Figure 1.4: A Hidden Jackpot?

In figure 1.4 A's utilities are very low, as far as he sees them, while E gets higher utilities. So A could assume that there are some higher utilities in for him as well and hope that E cooperates in pursuing the branch unknown to him.

Changing Winners

In imperfect information games, winning strategies can get lost (as in the game in figure 1.5; E had a winning strategy for the perfect information game, but none for the imperfect information game). But if the information gets even worse than imperfect, i.e. the epistemic relation is no equivalence relation anymore, and the other player knows about this, it might yield a winning strategy for him. In addition, believing that an action is good for you does not necessarily imply that after carrying out that action you still believe that it was a good idea.

Regard the game in figure 1.5. Here, at the beginning E thinks herself to be in a perfect information game. After A made his move, whatever he did E thinks A went left, so she plays according to her best interest and goes right. Now, if A had really gone left this would make her win. But A knows before his first step that E would believe him to go left, no matter what he really does (maybe because he is cheating and has some trick to make her believe this). So he can choose right and thus make E make *him* win.

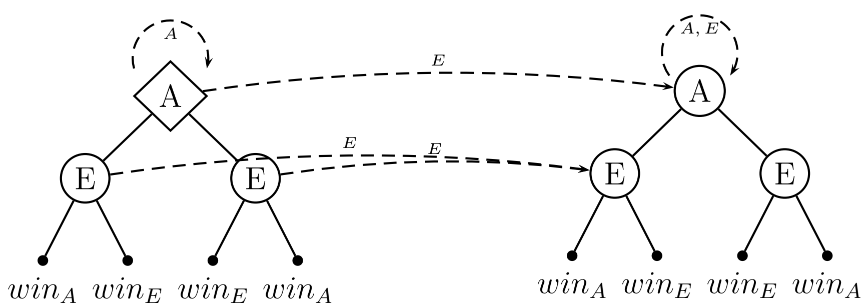

Figure 1.5: A has a winning strategy

Purpose of Thesis

In this thesis, I want to give a framework for reasoning about strategies in games including epistemic information. In games, there is uncertainty about the future behavior of players, the past behavior of players, which state a player is in, the structure of the game itself and the epistemic considerations of other players. As I have argued above, not all of these issues of epistemic information in games can get addressed in the framework of imperfect information games, i.e. the framework we have in the literature so far. The aim of this thesis is to give a larger framework capable of modeling the above examples. In addition, I will explore a bit of the logic used.

Outline of Thesis

In chapter 2 I introduce a way to model the examples in this chapter. I define epistemic game models (section 2.1) and a language to talk about them (section 4). Then I go back to the examples of this chapter and show how they are modeled in my framework. In the last section I discuss some general principles to demonstrate the expressive power of the language.

As a starting point for discussing model theory in chapter 3 I introduce bisimulations (section 3.1). They are used in comparing ordinary modal models and imperfect information games with epistemic game models (section 3.2).

In chapter 4 the logic of epistemic game models is explored. We get completeness for the minimal logic. Furthermore, we give some formulas characterizing frame properties (section 4.2).

There is a useful extension of the basic language, which we present in chapter 5. This language allows us to express more features of epistemic game models, as discussed in section 5.5. A short discussion of even more expressive tree languages follows (section 5.6).

As my models are static, you might wonder what their dynamic counterparts look like. In chapter 6 I present and discuss a dynamic approach of Baltag, Moss and Solecki that gets quite close. After outlining their system in section 6.1 I discuss how their update system works in games (section 6.2).

In chapter 7 I give an outlook on what issues there are in regarding misty games.

Chapter 8 sums up the results and gives some open questions.

Chapter 2

Language and Semantics

2.1 Epistemic Game Model

In this section I want to formalize the sort of diagrams that I used in the introduction to model epistemic games. I assume that a game is played by a set N of players which have to choose from a set A of possible actions. Let U be the set of possible utilities of the game. In the examples presented in the introduction, U either consists of elements of the form win_i, indicating that player i has won, or of sequences of natural numbers that assign a numerical utility to each player. (The former option can be reduced to the latter by using the sequence with a 1 at the ith position and zeros otherwise instead of win_i.)

For defining a model of epistemic games, let us first recall the definition of a tree.[1] (I use the following standard notation: given a binary relation S then S^+ is the transitive closure of S and S^* is the reflexive and transitive closure of S.)

Definition 2.1.1 (Tree) A *tree* $\mathcal{T}$ is a pair (T, S) consisting a set T of *nodes* (or *states*) and a binary relation S on T, the *successor* or *daughter* relation, such that

a) there exists a unique node $r \in T$, called the *root*, such that $\forall t \in T(rS^*t)$,

b) every element of T distinct from r has a unique S-*predecessor*, that is, for every $t \neq r$ there are unique $t' \in T$ such that $t'St$,

c) S is *acyclic*, that is, $\forall t \in T \neg(tS^+t)$.

A tree is *finite* if T is finite. A node or state t is *terminal* (or a *leaf*) if it has no *daughter*, that is, if there is no t' with tSt'. $\lhd$

To model games, I use a relational structure $(T, (S_a)_{a\in\mathsf{A}})$, where the S_a's are the action relations on the set T of states, such that $(T, \bigcup_{a\in\mathsf{A}} S_a)$ is a tree. Such structures

[1] see also [7], p. 6.

are called *tree-like*. Now, a game is just a tree-like action structure which has specified which player's turn it is at each non-terminal state and which gives the utilities at each terminal state. (Note that I do not require the action relations to be disjoint; in other words, in a given state two different actions may lead to the same successor state.) In order to have a framework that allows to assign further kinds of properties to the states of a game, the following definition includes a valuation function for an arbitrary set Φ of propositional variables.

Definition 2.1.2 (Game) A *game* $\mathcal{G}$ is a tuple $(T, (S_a)_{a\in\mathsf{A}}, (C_i)_{i\in\mathsf{N}}, (D_u)_{u\in\mathsf{U}}, V)$ where $(T, (S_a)_{a\in\mathsf{A}})$ is tree-like, $(C_i)_{i\in\mathsf{N}}$ and $(D_u)_{u\in\mathsf{U}}$ are families of subsets of T, and V is a valuation function $V : \Phi \to \mathfrak{P}(T)$ such that the C_i's are pairwise disjoint, $\bigcup_{i\in\mathsf{N}} C_i$ is the set of non-terminal nodes of the tree $(T, \bigcup_{a\in\mathsf{A}} S_a)$, and $\bigcup_{u\in\mathsf{U}} D_u$ is the set of its terminal nodes. $\lhd$

Remark 2.1.3 The intuition behind the foregoing definition is that $t \in C_i$ just in case it is i's turn at node t whereas $t \in D_u$ means that the utility at the (terminal) node t is u. Note that utility values are assigned only at the leaves. I do not exclude the possibility that a leaf can have more than one utility. You could have games where more than one player wins at a certain leaf – this would lead to reasoning about coalitions. Notice also that I do not add a preference relation as in Osborne and Rubinstein, because you can directly calculate one given the utilities. (Maybe you would have to decide if a player prefers an outcome where the other players fare worse or better, but that is a different issue and can be treated separately).

A game as defined above contains no epistemic information.[2] To express such information we take a whole bunch of games and connect them via an epistemic relation B_i for each player i. The idea is that at each state each player i is connected via B_i to the states he believes he could be in.

In order to properly combine games we make use of the somewhat technical notion of a *disjoint union*.[3] Recall the definition of the disjoint union $\biguplus_j T_j$ of a family $(T_j)_{j\in J}$ of sets: The basic idea is to first make disjoint copies of the sets and then take their union. The standard construction is $\biguplus_j T_j := \bigcup_j (T_j \times \{j\})$, that is, the elements of $\biguplus_j T_j$ are of the form (t, j) with $t \in T_j$. Consider the function $\iota_k : T_k \to \biguplus_j T_j$ with $\iota_k(t) = (t, k)$. By construction, ι_k is injective, every element of $\biguplus_j T_j$ is of the form $\iota_k(t)$ for some $k \in J$ (i.e. $\biguplus_j T_j = \bigcup_j \iota_j(T_j)$), and $\iota_j(T_j) \cap \iota_k(T_k) = \emptyset$ for all $j \neq k$.[4]

Suppose each of the T_j's carries a relation R_j of arity m, then one naturally gets an

[2] Unless you choose your actions to be epistemically – like changing your mind.

[3] see also [7], p. 53.

[4] Any set T together with injections $\iota_j : T_j \to T$ $(j \in J)$ satisfying these properties can serve as a representation of the disjoint union of $(T_j)_{j\in J}$. In particular, if the T_j's are already disjoint, we can choose $T := \bigcup_j T_j$ where the ι_j's are the inclusion functions.

m-ary relation on $\biguplus_j T_j$ by taking $\bigcup_j \iota_j(R_j)$.[5] We can apply this construction to games as follows:

Definition 2.1.4 (Disjoint Union) The *disjoint union* $\biguplus_j \mathcal{G}_j$ of a family $(\mathcal{G}_j)_{j\in J}$ of games $\mathcal{G}_j = (T_j, (S_{j,a})_{a\in\mathsf{A}}, (C_{j,i})_{i\in\mathsf{N}}, (D_{j,u})_{u\in\mathsf{U}}, V_j)$ is the tuple

$$(T, (S_a)_{a\in\mathsf{A}}, (C_i)_{i\in\mathsf{N}}, (D_u)_{u\in\mathsf{U}}, V),$$

with $T = \biguplus_j T_j$, $S_a = \bigcup_j \iota_j(S_{j,a})$, $C_i = \bigcup_j \iota_j(C_{j,i})$, $D_u = \bigcup_j \iota_j(D_{j,u})$, and $V : \Phi \to \mathfrak{P}(T)$ defined by $V(p) = \bigcup_j \iota_j(V_j(p))$. $\lhd$

Clearly the disjoint union of games is not a game in turn because $(T, (S_a)_{a\in\mathsf{A}})$ (in general) does not have a root. But, by construction, the disjoint union is a model of the same modal similarity type as the games it is built of.

Now we are prepared to give a formal definition of epistemic games models: An epistemic game model is a family of games, together with an epistemic relation for each player on the disjoint union of these games. One game gets marked as the "real game", this is the game actually played, while the other games represent what games players think they might be playing.

Definition 2.1.5 (Epistemic Game Model) An *epistemic game model* $\mathcal{M}$ is a triple $(\mathcal{G}, (\mathcal{E}_j)_{j\in J}, (B_i)_{i\in\mathsf{N}})$ where $(\mathcal{E}_j)_{j\in J}$ is a family of games (the *epistemic alternatives*), $\mathcal{G}$ is one of the $\mathcal{E}_j$'s, and $(B_i)_{i\in\mathsf{N}}$ is a family of serial[6] binary relations on the state set T of $\biguplus_j \mathcal{E}_j$. $\lhd$

Let $\mathcal{M}^G$ be the class of all epistemic game models. There is a special subclass of epistemic game models, those models where all epistemic alternatives are copies of the original game. These models are quite close to imperfect information games, in that they have complete information (see section 4.2), but they do not necessarily have equivalence relations for their epistemic relations.

Definition 2.1.6 (Simple Epistemic Game Model) An epistemic game model $\mathcal{M} = (\mathcal{G}, (\mathcal{E}_j)_{j\in J}, (B_i)_{i\in\mathsf{N}})$ with $\mathcal{E}_j = \mathcal{G}$ for all $j \in J$ is called *simple*. $\lhd$

Simple Models and Strategies

The main reason for introducing simple game models lies in their structural stability. From the point of view of strategies you can regard them with respect to survival under lack of information. For perfect information games, we can calculate a winning strategy for one of the players by using e.g. Zermelo's algorithm. For imperfect information

[5]with $\iota_j(R_j) = \{(\iota_j(t_1), \ldots, \iota_j(t_m)) \mid (t_1, \ldots, t_m) \in R\}$.

[6]A binary relation R on a set U is *serial* (or *right-unbounded*) iff for every $x \in U$ there is a $y \in U$ such that Rxy.

games, we can calculate a winning strategy in the same way, but only those can be successfully played that are stable under information sets. Those games can be won where the winning player (of the perfect information version) has a uniform strategy; i.e. in each state within an information set the player has to make the same move in order to follow his strategy. Simple epistemic game models have a similar condition: if a player has a *totally uniform* strategy for the perfect information version of the game, he will still win the simple game. A totally uniform strategy is a strategy that tells the player to perform the same move at each state where it is his turn. So this strategy is insensitive with respect to a players misinformation about where he is in the game.

For arbitrary epistemic game models, we cannot find such an easy criterion for 'survival' of a winning strategy. There is no way to go on simplifying the set of surviving strategies. But it might be worth it to think about strategies from a different point of view. In fact, a player can believe the strangest things about a game, as long as he knows what the game looks like when it is his turn to make a decision. In such a situation, he will play as good as in a perfect information game.

2.2 Language

I will define two languages: the basic language $\mathfrak{L}$ and the extended language $\mathfrak{L}^*$ (named like this because the Kleene Star is the most important feature of the extension). The basic language talks about local properties of the model; the extended language has a global flavor.

In this chapter I am concerned with the basic language, for $\mathfrak{L}^*$ see chapter 5.

Definition 2.2.1 (Alphabet) The alphabet of $\mathfrak{L}$ consists of

a) propositional constants: turn_i ($i \in \mathsf{N}$), elements of U, $\top$

b) propositional variables: elements of Φ

c) modal operators: $[a]$ ($a \in \mathsf{A}$), $\Box$, $[i]$ ($i \in \mathsf{N}$).

$\lhd$

Definition 2.2.2 (Formulas) The formulas of $\mathfrak{L}$ are of the form

$$p \mid \top \mid \text{turn}_i \mid u \mid \neg\varphi \mid \varphi \wedge \psi \mid [a]\varphi \mid [i]\varphi \mid \Box\varphi$$

where p ranges over Φ, φ and ψ are formulas of $\mathfrak{L}$, a ranges over A, i ranges over N, and u ranges over U. $\lhd$

Let '$\vee$', '$\rightarrow$', '$\bot$' and the diamonds be defined in the usual way, that is, $\bot := \neg\top$, $\langle a \rangle := \neg[a]\neg$, etc.

Definition 2.2.3 (Truth) Let $\mathcal{M}$ be an epistemic game model $(\mathcal{G}, (\mathcal{E}_j)_{j \in J}, (B_i)_{i \in \mathsf{N}})$ with $\biguplus_j \mathcal{E}_j = (T, (S_a)_{a \in \mathsf{A}}, (C_i)_{i \in \mathsf{N}}, (D_u)_{u \in \mathsf{U}}, V)$. We inductively define the notion of a formula φ being *true* (or *satisfied*) in $\mathcal{M}$ at state $t \in T$ as follows (with $S = \bigcup_{a \in \mathsf{A}} S_a$):

$\mathcal{M}, t \models \top$	always	
$\mathcal{M}, t \models \text{turn}_i$	iff	$t \in C_i$
$\mathcal{M}, t \models u$	iff	$t \in D_u \quad (u \in \mathsf{U})$
$\mathcal{M}, t \models p$	iff	$t \in V(p) \quad (p \in \Phi)$
$\mathcal{M}, t \models \neg\varphi$	iff	$\mathcal{M}, t \not\models \varphi$
$\mathcal{M}, t \models \varphi \wedge \psi$	iff	$\mathcal{M}, t \models \varphi$ and $\mathcal{M}, t \models \psi$
$\mathcal{M}, t \models [a]\varphi$	iff	$\mathcal{M}, t' \models \varphi$ for all $t' \in T$ such that tS_at'
$\mathcal{M}, t \models \Box\varphi$	iff	$\mathcal{M}, t' \models \varphi$ for all $t' \in T$ such that tSt'
$\mathcal{M}, t \models [i]\varphi$	iff	$\mathcal{M}, t' \models \varphi$ for all $t' \in T$ such that tB_it'

$\lhd$

For any formula φ, we say that φ is *satisfiable* if there is an epistemic game model $\mathcal{M}$ a node t such that $\mathcal{M}, t \models \varphi$. A formula φ is *valid on a model* if it is true at every node of that model. If a formula φ is valid in all models then we say it is *valid* and write $\models \varphi$. We can also restrict validity of φ to a specific epistemic alternative $\mathcal{E}_j$: φ is *valid on the game* $\mathcal{E}_j$ if φ is true at every element of $\iota_j(T_j)$.[7]

2.3 Examples Revisited

Let us go back to the examples in the introduction and see how we can formalize them in our language. We have actions, l and r, for 'going left' and 'going right', two players, A and E, and utilities as depicted in the figures.

Perfect or Imperfect?

What we wanted to express in the first picture (see figure 1.3) was that A does not know whether he is in a perfect or an imperfect information game with respect to E's knowledge.

Perfect information for E at a state s means the following: For each φ that holds at s we have that E knows φ at s, i.e. that both $\langle E\rangle\varphi$ and $[E]\varphi$ hold as well. In our example in a game with perfect information $[E][l](4,3) \vee [E][r](3,2)$ hold at both states in which it is E's turn.[8] This proposition holds at both states in the right game in figure 1.3 where it is E's turn. In contrast, in the states in the left game where it is

[7] Note that the injections ι_j are embeddings from $\mathcal{E}_j$ to $\biguplus_j \mathcal{E}_j$.

[8] Note that since both actions are deterministic and possible at each state we discuss, it does not matter whether we write $[l]$ or $\langle l\rangle$.

E's turn the negation of that proposition holds: $\neg[E][l](4,3) \wedge \neg[E][r](3,2)$. So, at the root we have that A considers both scenarios possible:

$$\langle A\rangle\Diamond(\neg[E][l](4,3) \wedge \neg[E][r](3,2)) \wedge \langle A\rangle\Diamond([E][l](4,3) \vee [E][r](3,2)).$$

That is, A thinks it possible that after he made a move E knows how to get to the best outcome for her or after his move she does not know how that. He even has more specific information about E. When A goes right E might know that she could get to $(4,3)$ by going left – or she might not know that (and thus choose to go right instead):

$$\langle A\rangle[r][E][l](4,3) \wedge \langle A\rangle[r]\langle E\rangle\langle l\rangle\neg(4,3).$$

At this point, you can see a possible extension of the language. We cannot talk about E actually carrying out an action, so we cannot express A's reasoning about how the game will end. In short, we cannot formalize how to come up with a strategy.[9]

Uncertainty About the Structure

In figure 1.4 the three dots indicate a multitude of possible games. So, actually our model does not just consist of the two games depicted, but of the left game plus all other games that start out the same way. 'All games' does of course depend on what our model is build of. What you minimally need are actions l and r, two players and utilities $(0,1),(1,5),(2,7),(4,5)$ and $(5,4)$. But there is nothing that tells you the set of thinkable outcomes is less than $I\!N \times I\!N$, similarly with actions. We have fixed the set of players to the one in the real game (i.e. we required that all epistemic alternatives have the same set of players). But it is of course not forbidden to have a player that does not participate in the game (but maybe gets an outcome value).

To sum up: in a misty game (as we call games with 'all options open' (see chapter 7)) the building material available plays an important role. You can think of it as what players consider possible in general. So far, I have not made a distinction between the 'epistemic horizon' of different players. Since mistiness shall formalize 'too many options to reason with at all' I leave it out here.

But for reasoning about what strategy a player should follow you do not need to have the model spelled out in detail. However small or large the epistemic horizon of the game is, mistiness just indicates that it is too much for the player to reason about. He has to leave the misty part blank.

Changing Winners

The crucial observation for A is that he knows that E will believe he has gone left after he went right, and thus she will believe going right herself will make her win, while in

[9]For a language to describe strategies, see [3].

fact it will make A win. This gets formalized as follows:

$$[A]([r][E][r]\text{win}_E \wedge [r]\langle r\rangle\text{win}_A).$$

2.4 General Principles

In this section I discuss some standard epistemic principles in order to demonstrate the expressive power of the basic language.

Let me first point out that I do not aim to model knowledge in epistemic game models. The reason for this is twofold. One reason is that I think of games from the point of view of the agents. Naturally, an agent cannot decide whether his firm belief is true or not, so for him knowledge and having just one belief option 'feels' the same. Whether he truly has knowledge is something that an outside referee has to decide. Second, the standard definition of knowledge as 'knowledge is justified true belief' you cannot model adequately. There are lots of refinements of this definition, but the common feature they share it that they are even less fit for modeling. The problem, of course, is the notion of justification. If you are satisfied with the definition of knowledge as true belief, you can characterize games with knowledge as those games, where the B_i relations are bending back (see chapter 4), i.e. as perfect information games.

Since most of the discussion in epistemic logic deals with knowledge I use the term sometimes in the following, but just in the sense of true belief.

Introspection

The introspection axioms express that you know your own mind. With positive introspection, which is formalized as $[i]\varphi \rightarrow [i][i]\varphi$ you know what you do know, while negative introspection, formalized as $\neg[i]\varphi \rightarrow [i]\neg[i]\varphi$ expresses that you know what you do not know.

The formula for positive introspection is the well-known formula for transitivity. No reflexivity is needed, so you can have positive introspection for false beliefs as well. In figure 2.1 you see an example for this. A is convinced that going right will make him win, and he is convinced that he is convinced of that fact.

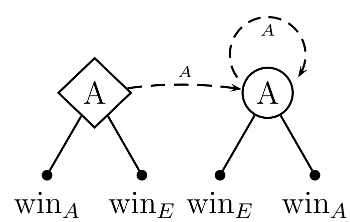

Figure 2.1: Positive Introspection

For negative introspection (valid qua the underlying frame) you need an epistemic relation that is symmetric and transitive, so you get additionally that your real situation is one of your belief alternatives. This means that negative introspection tells you more about the 'real world' than positive introspection does.

Perfect Recall

The formula $[i][a]\varphi \to [a][i]\varphi$ is called the axiom of perfect recall. What the formula means is that when you are certain that after some action φ is bound to be the case, then after that action, you are certain that φ *is* the case. This seems very plausible, but it is not necessarily the case. As an example regard the game in figure 1.5 and assume A has played right. Then E is certain that playing right will make her win – but after she does play right, she has lost and knows that. In an epistemic game model this means that at a state s whenever after performing an action a a player thinks it possible to be at a state t' then in s he thinks it possible to be at a state s' where action a will bring him to state t'.

The inverse of perfect recall, $[a][i]\varphi \to [i][a]\varphi$, is a kind of knowing beforehand. Whenever an action is bound to make a player be certain that φ holds, the player is certain that the action is bound to make φ hold. Again, this fails in the example in figure 1.5. Playing right will make E know that she has lost, while she is certain that playing right will make her win (and know that she has won).

Knowing Your Options

If you want to use epistemic game models for reasoning about strategies with their help, it is useful to regard some principles concerned with rules. If it is a player's turn, he should know it – and he should not believe it to be his turn when it is not. This is expressed by

$$\text{turn}_i \leftrightarrow [i]\text{turn}_i.$$

As well, it is useful that a player whose turn it is knows what actions he is allowed to take. This is expressed by

$$\text{turn}_i \wedge \langle a\rangle\top \leftrightarrow [i]\langle a\rangle\top.$$

Of course you need not require this formula to hold on the models. Letting go of the left to right direction opens the possibility that there are more options for the player than he is aware of. This might be unfortunate for him, but it does not cause problems with respect to playing. If you allow the right to left condition to be violated you can have the situation that a player's most rational strategy tells him to make a move which is not possible in the real game. This makes it difficult to make predictions about the move he would perform in playing the game, since the one predicted is a move he definitely cannot perform at all.

There is a related formula, that says when it is your turn you know what is the case:

$$\text{turn}_i \wedge \varphi \rightarrow [i]\varphi.$$

Models where this formula holds are from the point of view of strategies reduced to perfect information games. Each time you have to make a decision, you know what is the case. Your epistemic status while it is not your turn is uninteresting for reasoning about strategies.

Chapter 3

Some Basic Model Theory

3.1 Bisimulation

This chapter is concerned with the model theory of epistemic game models. We start by introducing an important tool: bisimulation. It will be useful in comparing epistemic game models with ordinary poly-modal models and imperfect information games.

Definition 3.1.1 (Bisimulation) Suppose $\mathcal{M} = (\mathcal{G}, (\mathcal{E}_j)_{j\in J}, (B_i)_{i\in \mathsf{N}})$ and $\mathcal{M}' = (\mathcal{G}', (\mathcal{E}'_k)_{k\in K}, (B'_i)_{i\in \mathsf{N}})$ be epistemic game models with $\biguplus_j \mathcal{E}_j = (T, (S_a)_{a\in \mathsf{A}}, (C_i)_{i\in \mathsf{N}}, (D_u)_{u\in \mathsf{U}}, V)$ and $\biguplus_k \mathcal{E}'_k = (T', (S'_a)_{a\in \mathsf{A}}, (C'_i)_{i\in \mathsf{N}}, (D'_u)_{u\in \mathsf{U}}, V')$. A non-empty binary relation $Z \subseteq T \times T'$ is a *bisimulation* between $\mathcal{M}$ and $\mathcal{M}'$ if the following conditions hold:

1. If tZt' then t and t' satisfy the same propositional constants and variables
2. If tZt' and tS_as then there is an $s' \in T'$ such that sZs' and $t'S'_as'$.
3. If tZt' and $t'S'_as'$ then there is an $s \in T$ such that sZs' and tS_as.
4. If tZt' and tB_is then there is an $s' \in T'$ such that sZs' and $t'B'_is'$.
5. If tZt' and $t'B'_is'$ then there is an $s \in T'$ such that sZs' and tB_is.

If Z is a bisimulation and tZt' then t and t' are called *bisimilar* (notation: $t \leftrightarroweq t'$). We say that $\mathcal{M}$ and $\mathcal{M}'$ are *game bisimilar* if the roots of $\mathcal{G}$ and $\mathcal{G}'$ are bisimilar. $\lhd$

Definition 3.1.1 closely resembles the standard definition of bisimulation used in modal logic, with conditions 2 and 4 as *forth conditions* and conditions 3 and 5 as *back conditions* (see appendix A). We can make this correspondence more explicit by presenting epistemic game models as models of a certain modal similarity type in the ordinary sense of modal logic. The similarity type in question consists of unary modalities for each $a \in \mathsf{A}$ and $i \in \mathsf{N}$, and nullary modalities (i.e. modal constants) for each $i \in \mathsf{N}$ and $u \in \mathsf{U}$. In the following, I will refer to any model of that similarity type briefly as a "general model":

Definition 3.1.2 (General Model) A *general model* is a tuple

$$(W, (S_a)_{a \in \mathsf{A}}, (B_i)_{i \in \mathsf{N}}, (C_i)_{i \in \mathsf{N}}, (D_u)_{u \in \mathsf{U}}, V)$$

where the S_a's and B_i's are binary relations on W, the C_i's and D_u's are unary relations, i.e. subsets of W, and $V : \Phi \to \mathfrak{P}(W)$ is a valuation function. $\triangleleft$

Every epistemic game model $\mathcal{M} = (\mathcal{G}, (\mathcal{E}_j)_{j \in J}, (B_i)_{i \in \mathsf{N}})$ gives rise to an *associated general model* in the following obvious way: Just combine the disjoint union $\biguplus_j \mathcal{E}_j$ with the family $(B_i)_{i \in \mathsf{N}}$ of epistemic relations. Bisimulation between epistemic game models as defined above is then essentially the same as bisimulation between the associated general models. The only thing to notice is the treatment of the modal constants, which correspond to the propositional constants in my definition of epistemic game models. But these two notions are just two sides of the same coin: The forth (and back) condition for unary relations, say for C_i, just says that if tZt' and $t \in C_i$ then there is a t' such that $t' \in C_i'$; in other words, if $\mathcal{M}, t \models \text{turn}_i$ then $\mathcal{M}', t' \models \text{turn}_i$, which is the condition that satisfaction of the propositional constant turn_i is invariant under bisimulation – see condition 1 of definition 3.1.1.

The general model associated with an epistemic game model inherits all the structural properties of the game model. In the next chapter, I will axiomatize some of these properties within the modal language introduced in section 2.2. The tree property of the action relations, however, cannot be enforced within this language. In the completeness proof of my axiomatization I will naturally come across general models that have all the structural properties of a model associated with an epistemic game model expect for the tree-likeness of the epistemic alternatives. I will call such a model a "generalized game model":

Definition 3.1.3 (Generalized Game Model) A *generalized (epistemic) game model* is a general model $(W, (S_a)_{a \in \mathsf{A}}, (B_i)_{i \in \mathsf{N}}, (C_i)_{i \in \mathsf{N}}, (D_u)_{u \in \mathsf{U}}, V)$ such that the B_i's are serial, the C_i's are pairwise disjoint, $\bigcup_{i \in \mathsf{N}} C_i$ is the set of nodes that are non-terminal with respect to $S := \bigcup_{a \in \mathsf{A}} S_a$, and $\bigcup_{u \in \mathsf{U}} D_u$ is the set of S-terminal nodes. $\triangleleft$

3.1.1 Bisimulation and Elementary Equivalence

In this section we examine the connection between bisimulation and elementary equivalence with respect to $\mathfrak{L}$.Two nodes t and t' in models $\mathcal{M}$ and $\mathcal{M}'$, respectively, are called *equivalent* if $\{\varphi | \mathcal{M}, t \models \varphi\} = \{\varphi | \mathcal{M}, t' \models \varphi\}$. Due to the close correspondence between epistemic game models and their associated general models, it is an easy task to transport the standard proofs to the setting of epistemic game models.[1]

[1]cf. [7], section 2.2.

Proposition 3.1.4 *Let* $\mathcal{M} = (\mathcal{G}, (\mathcal{E}_j)_{j\in J}, (B_i)_{i\in \mathsf{N}})$ *and* $\mathcal{M}' = (\mathcal{G}', (\mathcal{E}'_k)_{k\in K}, (B'_i)_{i\in \mathsf{N}})$ *be epistemic game models with* $\biguplus_j \mathcal{E}_j = (T, (S_a)_{a\in \mathsf{A}}, (C_i)_{i\in \mathsf{N}}, (D_u)_{u\in \mathsf{U}}, V)$ *and* $\biguplus_k \mathcal{E}'_k = (T', (S'_a)_{a\in \mathsf{A}}, (C'_i)_{i\in \mathsf{N}}, (D'_u)_{u\in \mathsf{U}}, V')$*. Then, for all* $t \in T$ *and* $t' \in T'$*,*

$$\mathcal{M}, t \leftrightarroweq \mathcal{M}', t' \quad \textit{implies} \quad \mathcal{M}, t \equiv \mathcal{M}', t'.$$

Proof. Suppose $t \in T$, $t' \in T'$, and $\mathcal{M}, t \leftrightarroweq \mathcal{M}', t'$. We show by induction on φ that $\mathcal{M}, t \models \varphi$ iff $\mathcal{M}', t' \models \varphi$.

- If φ is a propositional constant or variable, then $\mathcal{M}, t \models \varphi$ iff $\mathcal{M}', t' \models \varphi$, by condition 1 of definition 3.1.1.
- If $\varphi = \neg\psi$ and the claim holds for ψ, then $\mathcal{M}, t \models \psi$ iff $\mathcal{M}', t' \models \psi$. Consequently $\mathcal{M}, t \models \neg\psi$ iff $\mathcal{M}', t' \models \neg\psi$.
- If $\varphi = \psi \wedge \chi$ and the claim holds for ψ and χ you have $\mathcal{M}, t \models \varphi$ iff $\mathcal{M}, t \models \psi$ and $\mathcal{M}, t \models \chi$ iff $\mathcal{M}', t' \models \psi$ and $\mathcal{M}', t' \models \chi$ iff $\mathcal{M}', t' \models \psi \wedge \chi$.
- Assume $\varphi = \langle a\rangle\psi$ and the claim holds for ψ. $\mathcal{M}, t \models \langle a\rangle\psi$ iff there exists a $s \in T$ such that $tS_a s$ and $\mathcal{M}, s \models \psi$. By condition 3.1.1-2 there is an $s' \in T'$ such that $s \leftrightarroweq s'$ and $t'S'_a s'$. But then, by 3.1.1-1, $\mathcal{M}', s' \models \psi$, so $\mathcal{M}', t' \models \langle a\rangle\psi$.
- Assume $\varphi = \langle i\rangle\psi$ and the claim holds for ψ. $\mathcal{M}, t \models \langle i\rangle\psi$ iff there exists a $s \in T$ such that $tB_i s$ and $\mathcal{M}, s \models \psi$. By condition 3.1.1-4 there is an $s' \in T'$ such that $s \leftrightarroweq s'$ and $t'B'_i s'$. But then, by 3.1.1-1, $\mathcal{M}', s' \models \psi$, so $\mathcal{M}', t' \models \langle i\rangle\psi$.

QED

The other direction of the implication only holds with a further restriction; relations have to be image finite.

Definition 3.1.5 (Image Finite) A binary relation $R \subseteq X \times Y$ is *image finite* if for each $x \in X$ the set $\{y|Rxy\}$ is finite. An epistemic game model is *image finite* if each of its action and belief relations is image finite. $\lhd$

Note that this does not require any relation or the number of relations to be finite.

Proposition 3.1.6 *Let* $\mathcal{M} = (\mathcal{G}, (\mathcal{E}_j)_{j\in J}, (B_i)_{i\in \mathsf{N}})$ *and* $\mathcal{M}' = (\mathcal{G}', (\mathcal{E}'_k)_{k\in K}, (B'_i)_{i\in \mathsf{N}})$ *be image finite epistemic game models with* $\biguplus_j \mathcal{E}_j = (T, (S_a)_{a\in \mathsf{A}}, (C_i)_{i\in \mathsf{N}}, (D_u)_{u\in \mathsf{U}}, V)$ *and* $\biguplus_k \mathcal{E}'_k = (T', (S'_a)_{a\in \mathsf{A}}, (C'_i)_{i\in \mathsf{N}}, (D'_u)_{u\in \mathsf{U}}, V')$*. Then, for all* $t \in T$ *and* $t \in T'$*,*

$$\mathcal{M}, t \leftrightarroweq \mathcal{M}', t \quad \textit{iff} \quad \mathcal{M}, t \equiv \mathcal{M}', t'.$$

Proof. The left to right direction follows from proposition 3.1.4. It is left to show the converse, that is, if $\mathcal{M}, t \equiv \mathcal{M}', t'$ then $\mathcal{M}, t \leftrightarroweq \mathcal{M}', t$. For this, we show that $\equiv$ itself is a bisimulation. So let us go through the conditions in definition 3.1.1:

1. Assume $\mathcal{M}, t \equiv \mathcal{M}', t'$. Then, by definition 2.2.3, t and t' are satisfied by the same propositional constants and variables.

2. Assume $\mathcal{M}, t \equiv \mathcal{M}', t'$ and $tS_a s$. Suppose there is no $s' \in T'$ such that $\mathcal{M}, s \equiv \mathcal{M}', s'$ and $t'S'_a s'$. The set $R := \{s' | t'S'_a s'\}$ is non-empty since $\mathcal{M}, t \models \langle a \rangle \top$ and thus $\mathcal{M}', t' \models \langle a \rangle \top$. Because S'_a is image finite, R is finite, say $R = \{s'_1, \ldots, s'_n\}$. By assumption, there is a formula φ_k for every $s'_k \in R$ such that $\mathcal{M}, s \models \varphi_k$ but $\mathcal{M}', s_k \not\models \varphi_k$. It follows that $\mathcal{M}, t \models \langle a \rangle (\varphi_1 \wedge \cdots \wedge \varphi_n)$ and $\mathcal{M}', t' \not\models \langle a \rangle (\varphi_1 \wedge \cdots \wedge \varphi_n)$, in contradiction to $\mathcal{M}, t \equiv \mathcal{M}', t'$.

3. Assume $\mathcal{M}, t \equiv \mathcal{M}', t'$ and $t'S'_a s'$. Suppose there is no $s \in T$ such that $\mathcal{M}, s \equiv \mathcal{M}', s'$ and $tS_a s$. The set $R := \{s | tS_a s\}$ is non-empty since $\mathcal{M}', t' \models \langle a \rangle \top$ and thus $\mathcal{M}, t \models \langle a \rangle \top$. Since S_a is image finite, R is finite, say $R = \{s_1, \ldots, s_n\}$. By assumption, there is a formula φ_k for every $s_k \in R$ such that $\mathcal{M}', s' \models \varphi_k$ but $\mathcal{M}, s_k \not\models \varphi_k$. It follows that $\mathcal{M}', t' \models \langle a \rangle (\varphi_1 \wedge \cdots \wedge \varphi_n)$ and $\mathcal{M}, t \not\models \langle a \rangle (\varphi_1 \wedge \cdots \wedge \varphi_n)$, in contradiction to $\mathcal{M}, t \equiv \mathcal{M}', t'$.

Conditions 4 and 5 of definition 3.1.1 can be proved analogously. QED

3.2 Comparing Models

If you compare generalized game models with epistemic game models (EGMs), you find certain conditions making it possible to turn an generalized game model into a bisimilar epistemic game model. As well, not all epistemic game models are simple models, though that will just be seen in chapter 5 when we introduced an extended language that allows to look up the tree. In the end of the section I show that imperfect information games are a special case of simple models. Altogether, we will get an inclusion chain like this:

imperfect information games $\subset$ simple EGMs $\subset$ EGMs $\subset$ generalized game models

3.2.1 Comparison with Generalized Game Models

In the following I speak somewhat sloppily of bisimulations between a general model and an epistemic game model, when I mean a bisimulation between the general model and the generalized epistemic model associated with that epistemic game model.

Here, we have three different results: all of them share the condition that the counterparts of the B_i relations have to be serial as well and that the turn- and utility-relations are distributed adequately. Apart from this, you have bisimilarity 1) if the general model is rooted, 2) if the union of the action relations is well-founded, and 3) if there is just one action relation (in particular no relations inverse to each other). Note that this representation is not unique; for example you can choose which one is your 'original' game.

To prove these results I use the standard technique of "unraveling".[2]

Proposition 3.2.1 *Each rooted generalized game model is the bounded morphic image of an epistemic game model.*

Proof. Let $\mathbb{M} = (W, (S_a)_{a\in\mathsf{A}}, (B_i)_{i\in\mathsf{N}}, (C_i)_{i\in\mathsf{N}}, (D_u)_{u\in\mathsf{U}}, V)$ be a rooted generalized game model with root w_0. Let W' be the set of all finite sequences $(w_0, w_1, \dots, w_n)$ with $n \geq 0$ such that for every $1 \leq k \leq n$ either $w_{k-1}S_a w_k$ or $w_{k-1}B_i w_k$ for some $a \in \mathsf{A}$ and $i \in \mathsf{N}$, respectively. Define binary relations S'_a and B'_i on W' and subsets C'_i and D'_u of W' as follows:

$$S'_a = \{((w_0, w_1, \dots, w_n), (w_0, w_1, \dots, w_n, w_{n+1})) \mid w_n S_a w_{n+1}\},$$
$$B'_i = \{((w_0, w_1, \dots, w_n), (w_0, w_1, \dots, w_n, w_{n+1})) \mid w_n B_i w_{n+1}\},$$
$$C'_i = \{(w_0, w_1, \dots, w_n) \mid w_n \in C_i\}, \quad D'_u = \{(w_0, w_1, \dots, w_n) \mid w_n \in D_u\}.$$

Clearly, the relation B'_i inherits seriality from B_i. Define $S' := \bigcup_{a\in\mathsf{A}} S'_a$ and $S := \bigcup_{a\in\mathsf{A}} S_a$. Then C'_i is the set of all elements of W' that are non-terminal with respect to S', and D'_u is the set of all S'-terminal elements. For $(w_0, w_1, \dots, w_n) \in C'_i$ iff $w_n \in C_i$ iff there is a $w \in W$ such that $w_n S w$, that is, $(w_0, w_1, \dots, w_n) S' (w_0, w_1, \dots, w_n, w)$. You can use a similar argument to show the claim for D'_u. In addition, we have $C'_i \cap C'_j = \emptyset$ for $i \neq j$. Finally, define $V' : \Phi \to \mathfrak{P}(W')$ such that $(w_0, w_1, \dots, w_n) \in V'(p)$ iff $w_n \in V(p)$.

Then $\mathbb{M}' = (W', (S'_a)_{a\in\mathsf{A}}, (B'_i)_{i\in\mathsf{N}}, (C'_i)_{i\in\mathsf{N}}, (D'_u)_{u\in\mathsf{U}}, V')$ is a generalized game model. We claim that $\mathbb{M}$ is a bounded morphic image of $\mathbb{M}'$. We show that the function $f : W' \to W$ such that $f(w_0, w_1, \dots, w_n) = w_n$ is a surjective bounded morphism from $\mathbb{M}'$ to $\mathbb{M}$ (see definition A.0.2):

1. $f(w_0, w_1, \dots, w_n) \in V(p)$ iff $(w_0, w_1, \dots, w_n) \in V'(p)$.
2. If $(w_0, u_1, \dots, u_n) S'_a (w_0, v_1, \dots, v_m)$ then $u_n S_a v_m$.

   If $(w_0, u_1, \dots, u_n) B'_i (w_0, v_1, \dots, v_m)$ then $u_n B_i v_m$.
3. If $f(w_0, u_1, \dots, u_n) S_a v$ then $(w_0, u_1, \dots, u_n) S'_a (w, u_1, \dots, u_n, v)$ and $f(w_0, u_1, \dots, u_n, v) = v$.

   If $f(w_0, u_1, \dots, u_n) B_i v$ then $(w_0, u_1, \dots, u_n) B'_i (w, u_1, \dots, u_n, v)$ and $f(w_0, u_1, \dots, u_n, v) = v$.

[2]cf. [7], p. 62 f.

Moreover, $(w_0, u_1, \ldots, u_n) \in C'_i$ iff $u_n \in C_i$, and $(w_0, u_1, \ldots, u_n) \in D'_u$ iff $u_n \in D_u$. Since $\mathbb{M}$ is rooted, each node can be reached from the root and thus occurs at the end of a finite sequence. So f is surjective and $\mathbb{M}$ is a bounded morphic image of $\mathbb{M}'$.

Now we show that $\mathbb{M}'$ is associated with an epistemic game model. Let $(T_j)_{j\in J}$ be the family of S'-connected components of W', and let $S_{j,a}$, $C_{j,i}$, and $D_{j,u}$ be the restrictions of S'_a, C'_i, and D'_u to T_j, respectively. Then, by construction, $(T_j, (S_{j,a})_{a\in\mathsf{A}})$ is tree-like and $\mathcal{E}_j := (T_j, (S_{j,a})_{a\in\mathsf{A}}, (C_{j,i})_{i\in\mathsf{N}}, (D_{j,u})_{u\in\mathsf{U}}, V_j)$ is a game. Choose $\mathcal{G}$ as the $\mathcal{E}_k$ with $(w_0) \in T_k$; then $(\mathcal{G}, (\mathcal{E}_j)_{j\in J}, (B_i)_{i\in\mathsf{N}})$ is the desired epistemic game model. QED

The following result is a slight generalization of the foregoing one. It makes use of a slightly different unraveling construction that only unravels the action relations.

Proposition 3.2.2 *If the union of the action relations of a generalized game model is well-founded, that model is the bounded morphic image of an epistemic game model.*

Proof. Let $\mathbb{M} = (W, (S_a)_{a\in\mathsf{A}}, (B_i)_{i\in\mathsf{N}}, (C_i)_{i\in\mathsf{N}}, (D_u)_{u\in\mathsf{U}}, V)$ be a generalized game model. By assumption, $S := \bigcup_{a\in\mathsf{A}} S_a$ is well-founded. Let W_0 be the set of S-initial elements of W, that is, for all $w \in W_0$ there is no $v \in W$ with vSw. Since S is well-founded, W_0 is non-empty. Let W' be the set of all finite sequences $(w_0, w_1, \ldots, w_n)$ with $n \geq 0$ such that $w_0 \in W_0$, $w_k \in W$, and $w_{k-1}Sw_k$ for $1 \leq k \leq n$. Now define binary relations S'_a and B'_i on W' and subsets C'_i and D'_u of W' as follows:

$$S'_a = \{((w_0, w_1, \ldots, w_n), (w_0, w_1, \ldots, w_n, w_{n+1})) \mid w_n S_a w_{n+1}\},$$
$$B'_i = \{((v_0, v_1, \ldots, v_m), (w_0, w_1, \ldots, w_n)) \mid v_m B_i w_n\},$$
$$C'_i = \{(w_0, w_1, \ldots, w_n) \mid w_n \in C_i\}, \quad D'_u = \{(w_0, w_1, \ldots, w_n) \mid w_n \in D_u\}.$$

Moreover, define $V' : \Phi \to \mathfrak{P}(W')$ such that $(w_0, w_1, \ldots, w_n) \in V'(p)$ iff $w_n \in V(p)$. One easily checks that $\mathbb{M}' = (W', (S'_a)_{a\in\mathsf{A}}, (B'_i)_{i\in\mathsf{N}}, (C'_i)_{i\in\mathsf{N}}, (D'_u)_{u\in\mathsf{U}}, V')$ is a generalized game model. We claim that $\mathbb{M}$ is a bounded morphic image of $\mathbb{M}'$ by the function $f : W' \to W$ such that $f(w_0, w_1, \ldots, w_n) = w_n$. Since the proof is very similar to the one used for proposition 3.2.1, let us only check the back condition for B_i: Suppose $f(w_0, \ldots, w_m)B_i v$; by definition of W_0 there is a $v_0 \in W_0$ such that $(v_0, \ldots, v_n, v) \in W'$ for some v_k $(1 \leq k \leq n)$; then $(w_0, \ldots, w_n)B'_i(v_0, \ldots, v_n, v)$.

This leaves us to show that $\mathbb{M}'$ is associated with an epistemic game model. For each $w \in W_0$ let T_w be the set of all finite sequences starting with w. With $J := W_0$ and applying the definitions used at the end of the proof of proposition 3.2.1 we get a family $(\mathcal{E}_j)_{j\in J}$ of games. Choose one of them to be $\mathcal{G}$. Then $(\mathcal{G}, (\mathcal{E}_j)_{j\in J}, (B_i)_{i\in\mathsf{N}})$ is the desired epistemic game model. QED

The reader should be warned that the construction in the above proof does of course not enforce any of the resulting epistemic alternatives to be a finite structure.

It would be nice to have a result saying that every generalized game model is bisimilar to an epistemic game model. But this is not the case: Consider e.g. the integers together with $<$ and $>$ as actions, no epistemic relations, and some valuation. For the sake of contradiction, assume you can find a general model bisimilar to $(I\!N, <, >)$ that is associated with an epistemic game model. Regard a (sub)tree in the epistemic game model. Since $<$ and $>$ are inverse to each other and this is modally definable, at the root of that tree either $[<]\bot$ or $[>]\bot$ holds, both of which hold nowhere on the integers.

You can have a weaker result, though. If you have only one action, you can cope with non-well-foundedness by using infinitely many trees.

Proposition 3.2.3 *A generalized game model with only one action relation is the bounded morphic image of an epistemic game model.*

Proof. Suppose $\mathbb{M} = (W, S, (B_i)_{i \in \mathsf{N}}, (C_i)_{i \in \mathsf{N}}, (D_u)_{u \in \mathsf{U}}, V)$ is a generalized game model (with a single action relation S). Let W' be the set of all finite sequences $(w_1, \ldots, w_n)$ with $n \geq 1$, $w_k \in W$, and $w_{k-1} S w_k$ for $1 < k \leq n$. Define binary relations S' and B_i''s on W' such that

$$S = \{((w_1, \ldots, w_n), (w_1, \ldots, w_n)) \mid w S w_{n+1}\},$$
$$B_i' = \{((v_1, \ldots, v_m), (w_1, \ldots, w_n)) \mid v_m B_i w_n\}.$$

The definitions of C_i', D_u', and V' can be taken from the proof of the previous proposition. This gives a generalized game model $\mathbb{M}'$, of which $\mathbb{M}$ is a bounded morphic image by the function $f : W' \to W$ with $f(w_1, \ldots, w_n) = w_n$. (Note that f is surjective since $(w) \in W'$ for each $w \in W$.) Let T_w be the set of all finite sequences starting with w. Now take $J := W$ and proceed as in the proof of the previous proposition. QED

Note that if you ignore the belief relations, the previous result is just an instance of the following general fact: If you know that every point-generated submodel $\mathbb{M}_w$ of a model $\mathbb{M}$ is the bounded morphic image of some model $\mathbb{M}_w'$ (of the same similarity type) then $\mathbb{M}$ is the bounded morphic image of the disjoint union $\bigcup_w \mathbb{M}_w'$.

3.2.2 Models and Simple Models

There is a simple technical trick to turn each epistemic game model into a simple model: just add a common root to all trees. Then you have just one tree, so it is trivially a simple game. Adding a root might take infinitely many finite trees into one big infinitely branching tree, so you might not end up with something this is nice to handle. If you want to keep your basic actions deterministic, i.e. that each action relation is a partial function, you might be forced to introduce a large set of new actions that just serve the purpose of getting from the root to each tree.

With respect to the basic language you can easily find a bisimulation between both models: just do not take the new root into account. This, of course, is no game bisimulation. But if you extend the language by a modality capable of looking up a tree, this does not yield a bisimilar model anymore. Since we have not introduced the extended language yet, we will postpone discussing this to chapter 5.

Using this trick, you lose information about which of the new subgames is actually played.

3.2.3 Simple Models and Imperfect Information Games

One might be tempted to think that simple epistemic game models are bisimilar to imperfect information games, but this is not the case as you can easily see in figure 3.1. There you have that at the right state in the left game E thinks to be at the left state in the right game, and she does not consider her actual state as a possibility, i.e. her belief relation is not reflexive and in particular no equivalence relation.

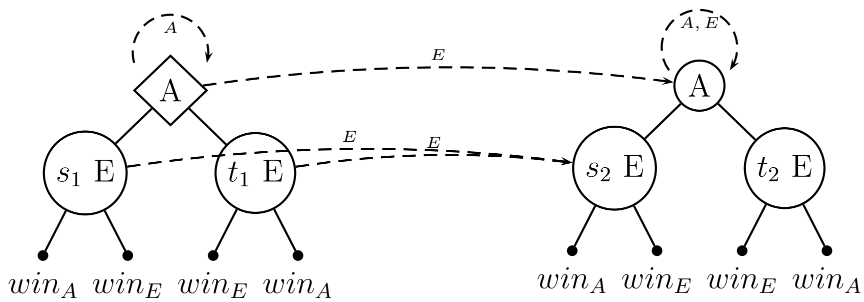

Figure 3.1: A Simple Epistemic Game Model

Even if you drop the restriction that all epistemic relations on imperfect information games have to be equivalence relations, you cannot find a bisimilar imperfect information game to the one represented in figure 3.1. The closest you could get is by contracting the two trees into one, keeping the epistemic relations between and in both trees. Since you want to keep all formulas of the form $\langle E\rangle\varphi$ satisfied at each state, and you have $s_1 \models \langle E\rangle\langle r\rangle\text{win}_E$, you need to have $(s, s) \in B_E$ in the one tree model. As you have $t_1 \models \langle E\rangle\langle r\rangle\text{win}_E$ (t, s) should be in B_E as well. By doing that you get a tree like in figure 3.2.

But this model is not bisimilar to the other one, just regard the root of the second tree: There you have perfect information if you go down right, while in the contracted tree you don't even know you are in that state when you go there. That is, B_E is too large. We have $t_2 \models [E]\langle l\rangle\text{win}_E$, but $t \models \langle E\rangle\langle l\rangle\neg\text{win}_E$. Since, as we argued above,

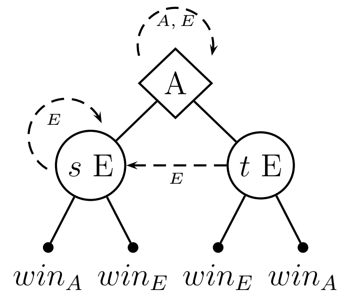

Figure 3.2: Almost an Imperfect Information Game

(t, s) has to be in the new B_E there can be no one-game model for the game in figure 3.1.

There are simple epistemic game models that you can contract into a single tree model. Suppose $\mathcal{M} = (\mathcal{G}, (\mathcal{E}_j)_{j\in J}, (B_i)_{i\in \mathsf{N}})$ is a simple epistemic game model, that is, $\mathcal{E}_j = \mathcal{G}$ for all $j \in J$. Suppose $\mathcal{M}' = (\mathcal{G}, \mathcal{G}, (B'_i)_{i\in \mathsf{N}})$ is a single tree model and $\mathcal{G} = (T, (S_a)_{a\in \mathsf{A}}, (C_i)_{i\in \mathsf{N}}, (D_u)_{u\in \mathsf{U}}, V)$. I now ask for conditions on B_i and B'_i such that (the general model associated with) $\mathcal{M}'$ is a bounded morphic image of (the general model associated with) $\mathcal{M}$ under the surjective function $\pi : \biguplus_{j\in J} T \to T$ with $\pi(\iota_j(t)) = t$. The back condition for π runs as follows: If $\pi(\iota_j(t))B'_i s$, i.e. $tB'_i s$, then there is a $k \in J$ and an $s' \in T$ such that $\iota_j(t)B_i\iota_k(s')$ and $\pi(\iota_k(s')) = s$. But $\pi(\iota_k(s')) = s'$. So the condition becomes: if $tB'_i s$ then $\forall j \exists k(\iota_j(t)\, B_i\, \iota_k(s))$. Since π must respect the relation B_i, we furthermore require that if $\iota_j(t)B_i\iota_k(s)$ then $tB'_i s$.

Assume now that $tB'_i s$ iff $\forall j \exists k(\iota_j(t)\, B_i\, \iota_k(s))$. If in this case π is a surjective bounded morphism, I speak of a *contraction* of $\mathcal{M}$ to $\mathcal{M}'$. According what has been said so far, π is a contraction iff

$$\iota_{j_0}(t)\, B_i\, \iota_{k_0}(s) \quad \text{implies} \quad \forall j \exists k(\iota_j(t)\, B_i\, \iota_k(s)).$$

for all $j_0, k_0 \in J$. Let me express this condition less formally: In a simple epistemic game model I call two states *similar* if they are the same or at the same place in two copies of the tree the simple model is based on; that is, two elements t' and t'' of $\biguplus_{j\in J} T$ are similar iff $t' = \iota_j(t)$ and $t'' = \iota_k(t)$ for some $j, k \in J$ and $t \in T$.

Proposition 3.2.4 *A simple epistemic game model can be contracted into a model with just one tree iff, for each B_i and nodes t, s, if $tB_i s$ then for each t' similar to t there exists a s' similar to s such that $t'B_i s'$.*

Note that this result just deals with keeping the underlying tree unchanged. If you unite all trees into a simple model by adding an extra root, you instantly have a one-tree-model.

3.3 Problems

We now turn to some applications of the apparatus developed so far.

Invariance for Infinite Epistemic Relations

In the invariance lemma, proposition 3.1.6, we restricted ourselves to image finiteness for both action and epistemic relations. It would be nice to extend this result by dropping one of the finiteness conditions. I have to leave it as an open question whether you can get invariance for infinite epistemic relations.

Staying in the Tree

With the language $\mathfrak{L}$ you have not enough expressiveness to distinguish between games consisting of one tree or of more than one tree when you are further down the tree. For example, at the two marked states in figure 3.3 exactly the same formulas hold. The reason for this is simple: we do not have a modality for reverse actions. We will come back to this in chapter 5.

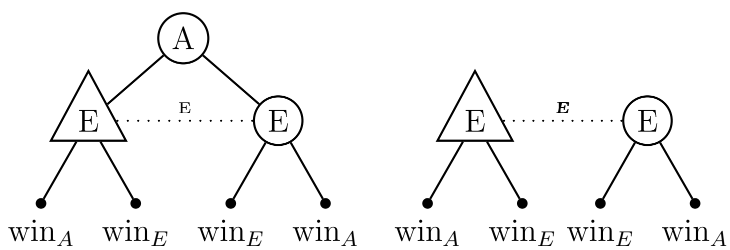

Figure 3.3: Same tree or not?

Chapter 4

Logics

4.1 Minimal Logic

In this chapter, I introduce a minimal axiomatic system Σ based on the language $\mathfrak{L}$, that is sound and strongly complete with respect to the class of epistemic game models. To this end, I assume in the following that the set N of players, the set A of actions, and the set U of utilities are finite.[1] The system Σ is given in figure 4.1.

Soundness for the Minimal Logic

Proposition 4.1.1 *Σ is sound with respect to the class of epistemic game models.*

Proof. Let $\mathcal{M}$ be any epistemic game model $(\mathcal{G}, (\mathcal{E}_j)_{j\in J}, (B_i)_{i\in\mathsf{N}})$, with $\biguplus_j \mathcal{E}_j = (T, (S_a)_{a\in\mathsf{A}}, (C_i)_{i\in\mathsf{N}}, (D_u)_{u\in\mathsf{U}}, V)$. Define $S := \bigcup_a S_a$. We first consider the basic axioms. Validity of the propositional tautologies is trivial. Checking the validity of normality is standard: Assume $\mathcal{M}, t \models [i]p \wedge [i](p \rightarrow q)$, i.e. for each node s such that tB_is it holds that $\mathcal{M}, s \models p$ and $\mathcal{M}, s \models p \rightarrow q$; hence $\mathcal{M}, s \models q$ and thus $\mathcal{M}, t \models [i]q$. So $\mathcal{M}, t \models [i]p \wedge [i](p \rightarrow q) \rightarrow [i]q$. As for $[i]$-seriality, since B_i is serial, each node is B_i-related to another node. So, if $[i]p$ holds at each node, we have $\langle i\rangle p$ as well. The action mix axiom is an immediate consequence of definitions: $\mathcal{M}, t \models \Box p$ iff $\mathcal{M}, t \models p$ for all s such that tSs; but $S = \bigcup_a S_a$.

Now consider the special axioms. Turn axiom 1 is valid because $C_i \cap C_j = \emptyset$. For the second turn axiom note that $\mathcal{M}, t \models \Diamond\top$ iff t in non-terminal with respect to S; hence it follows from our definition of epistemic game models that $\mathcal{M}, t \models \mathrm{turn}_i$ for some $i \in \mathsf{N}$. Validity of the utility axiom can be shown in the same way.

Let us finally consider the rule part. Soundness of modus ponens is obvious; necessitation is immediate too: for instance, if $\mathcal{M} \models \varphi$ then $\mathcal{M}, t \models \varphi$ for all nodes $t \in T$.

[1] Note that if you don't assume the action relations to be deterministic, i.e. a partial function, you can still have infinite branching in a game model.

Basic Axioms	
	all propositional tautologies
$[i]$-normality	$[i](p \to q) \to ([i]p \to [i]q)$
$[a]$-normality	$[a](p \to q) \to ([a]p \to [a]q)$
$\Box$-normality	$\Box(p \to q) \to (\Box p \to \Box q)$
$[i]$-seriality	$[i]p \to \langle i \rangle p$
action mix axiom	$\Box p \leftrightarrow \bigwedge_{a \in \mathsf{A}} [a]p$
Special Axioms	
turn axiom 1	$\text{turn}_i \wedge \text{turn}_j \to \bot \quad (i \neq j)$
turn axiom 2	$\Diamond\top \leftrightarrow \bigvee_{i \in \mathsf{N}} \text{turn}_i$
utility axiom	$\Box\bot \leftrightarrow \bigvee_{u \in \mathsf{U}} u$
Modal Rules	
modus ponens	From $\vdash \varphi$ and $\vdash \varphi \to \psi$ infer $\vdash \psi$
$[a]$-necessitation	From $\vdash \varphi$ infer $\vdash [a]\varphi$
$[i]$-necessitation	From $\vdash \varphi$ infer $\vdash [i]\varphi$
uniform substitution	From $\vdash \varphi$ infer $\vdash \vartheta$, if ϑ is obtained from φ by uniformly replacing propositional variables in φ by arbitrary formulas

Figure 4.1: The logical system Σ

So each node of $\mathcal{M}$ reachable by B_i from some other node satisfies φ true as well; hence $\mathcal{M} \models [i]\varphi$. Soundness of uniform substitution is proved as usual by induction on formulas. QED

Completeness of the Minimal Logic

In order to show the completeness of the minimal logic it suffices to show that each Σ-consistent set Γ of $\mathfrak{L}$-formulas is satisfiable on some epistemic game model. We proceed as follows:

1. We define a canonical model for Σ on which Γ is satisfiable; we do this by applying the standard construction of completeness proofs for normal modal logics.

2. We show that the canonical model gives us a generalized game model $\mathbb{M}$ on which Γ is satisfiable.

3. We take the submodel $\mathbb{M}'$ of $\mathbb{M}$ that is generated by some point that satisfies Γ. Since $\mathbb{M}'$ is rooted, it is the bounded morphic image of an epistemic game model $(\mathcal{G}, (\mathcal{E}_j)_{j \in J}, (B_i)_{i \in \mathsf{N}})$, by proposition in the sense of definition 3.1.2. Hence, by proposition 3.1.6, the root of $\mathcal{G}$ satisfies Γ, which concludes the proof.

As for step 1, we define the canonical model $\mathbb{M}^\Sigma$ for Σ along the standard definition (cf. definition A.0.6) as the tuple $(W^\Sigma, S^\Sigma, (S_a^\Sigma)_{a \in \mathsf{A}}, (B_i^\Sigma)_{i \in \mathsf{N}}, (C_i^\Sigma)_{i \in \mathsf{N}}, (D_u^\Sigma)_{u \in \mathsf{U}}, V^\Sigma)$ such that W^Σ is the set of all maximal Σ-consistent sets of $\mathfrak{L}$-formulas, $V^\Sigma(p) = \{w \in W^\Sigma | p \in w\}$, for $p \in \Phi$, and, for all $w, w' \in W^\Sigma$,

- $w S^\Sigma w'$ iff for all formulas φ, if $\Box\varphi \in w$ then $\varphi \in w'$,
- $w S_a^\Sigma w'$ iff for all formulas φ, if $[a]\varphi \in w$ then $\varphi \in w'$,
- $w B_i^\Sigma w'$ iff for all formulas φ, if $[i]\varphi \in w$ then $\varphi \in w'$,
- $w \in C_i^\Sigma$ iff $\text{turn}_i \in w$,
- $w \in D_k^\Sigma$ iff $u_k \in w$.

There are two technicalities the reader should notice. First, the model constructed this way is *not* a general model in the sense of definition 3.1.2 because the modal similarity type of general models does *not* include the operator $\Box$; this problem will be tackled in step 2 below. The second thing is the presence of the nullary modalities (or propositional constants) turn_i and u. But there is nothing in the standard completeness proof for normal modal logics as, for instance, given in [7, chapter 4] that precludes nullary modalities from being there – they just give not rise to any normality axioms and necessitation rules.

Recall the standard argument that any Σ-consistent set Γ of $\mathfrak{L}$-formulas is satisfiable on the canonical model $\mathbb{M}^\Sigma$: Since Γ is Σ-consistent, the Lindenbaum Lemma (appendix, lemma A.0.9) gives us a maximal Σ-consistent set Γ' with $\Gamma \subseteq \Gamma'$. According to the Truth Lemma (appendix, lemma A.0.10), we have that Γ is satisfied on the canonical model at $\Gamma' \in W^\Sigma$. For the sake of completeness, let me explicitly restate the Truth Lemma:

Lemma 4.1.2 (Truth Lemma) *For any formula φ in $\mathfrak{L}$ and $w \in W^\Sigma$ we have*

$$\mathbb{M}^\Sigma, w \models \varphi \quad \textit{iff} \quad \varphi \in w.$$

Proof. By induction on φ. Let φ be a nullary modality, say turn_i. Then $\mathbb{M}^\Sigma, w \models \text{turn}_i$ iff $w \in \text{turn}_i$, by definition of C_i^Σ. The case of propositional variables is equally trivial; the induction step for Boolean connectives is straightforward. Consider finally a binary modality, say $\varphi = \langle a \rangle \psi$. Then $\mathbb{M}^\Sigma, w \models \langle a \rangle \psi$ iff there is a w' with $w S_a^\Sigma w'$ and $\mathbb{M}^\Sigma, w' \models \psi$, i.e. $\psi \in w'$, by induction hypothesis; hence $\langle a \rangle \psi \in w$, by lemma A.0.7

of the appendix. Conversely, suppose $\langle a\rangle\psi \in w$. Then, by the Existence Lemma (appendix, lemma A.0.8), there is a w' such that $wS^{\Sigma}_a w'$ and $w' \in \psi$; hence, by induction hypothesis, $\mathbb{M}^{\Sigma}, w' \models \psi$ and thus $\mathbb{M}^{\Sigma}, w \models \langle a\rangle\psi$. QED

We now continue with step 2. As noted before, the canonical model $\mathbb{M}^{\Sigma}$ is not a general model in the sense of definition 3.1.2 but carries a little bit overhead, namely the relation S^{Σ}. In order to eliminate S^{Σ} in accordance with the truth definition for $\Box$ given in definition 2.2.3, we need to show that $S^{\Sigma} = \bigcup_a S^{\Sigma}_a$. Suppose $wS^{\Sigma}_a w'$; if $\Box\varphi \in w$ then, by the action mix axiom, uniform substitution, and modus ponens, $[a]\varphi \in w$; hence $\varphi \in w'$ and thus $wS^{\Sigma}w'$. For the converse direction, first note that the action mix axiom by uniform substitution, propositional tautologies, and modus ponens gives rise to the theorem $\Diamond\varphi \leftrightarrow \bigvee_{a\in\mathsf{A}}\langle a\rangle\varphi$. Now suppose $wS^{\Sigma}w'$; by lemma A.0.7 of the appendix, $\varphi \in w'$ implies $\Diamond\varphi \in w$; hence $\langle a\rangle\varphi \in w$ for some a, and thus $wS^{\Sigma}_a w'$, again by lemma A.0.7.

Next we verify that $\mathbb{M} := (W^{\Sigma}, (S^{\Sigma}_a)_{a\in\mathsf{A}}, (B^{\Sigma}_i)_{i\in\mathsf{N}}, (C^{\Sigma}_i)_{i\in\mathsf{N}}, (D^{\Sigma}_u)_{u\in\mathsf{U}}, V^{\Sigma})$ is a generalized game model. Let us first check that B^{Σ}_i is serial. Given $w \in W^{\Sigma}$ we need to show that there is a $w' \in W^{\Sigma}$ with $wB^{\Sigma}_i w'$. Since w is a maximal Σ-consistent set of $\mathfrak{L}$-formulas, it contains the $[i]$-seriality axiom $[i]p \to \langle i\rangle p$; in addition, because of its maximality, w is closed under Σ-derivability; so w contains $[i]\top \to \langle i\rangle\top$ by uniform substitution. Moreover, since $\top$ is a propositional tautology, we have $[i]\top \in w$, by $[i]$-necessitation, and hence $\langle i\rangle\top \in w$, by modus ponens. Application of the Existence Lemma (appendix, lemma A.0.8) then gives us a w' such that $wB^{\Sigma}_i w'$. Hence B^{Σ}_i is serial. Because of turn axiom 1, we have $C_i \cap C_j = \emptyset$ for $i \neq j$. Now we check that $w \in W^{\Sigma}$ is non-terminal with respect to S^{Σ} iff $w \in \bigcup_i C^{\Sigma}_i$. By the Existence Lemma, w is non-terminal with respect to S^{Σ} iff $\Diamond\top \in w$, that is, by the second turn axiom, iff there is an $i \in \mathsf{N}$ such that $\mathrm{turn}_i \in w$. The condition on the D_u's can be easily verified the same way.

All in all, we can conclude that Γ is satisfiable on a generalized game model. Since with respect to step 3 everything has been already said above, we finally have the desired result:

Proposition 4.1.3 Σ *is strongly complete with respect to the class of epistemic game models.*

4.2 Correspondence

In this section I will show for two properties to what frame classes they correspond. They are both not common in the literature, but handy in dealing with epistemic game models.

Bending Back

Now I want to show the correspondence of

$$p \to \Box p$$

to the class of frames where $\forall x, y(xRy \to x = y)$ holds, i.e. those frames that just consist of isolated points that are related to themselves at most. I call the property 'bending back'. If you have epistemic relations that are bending back and serial, you get perfect information. In fact, you have redundant modalities, because $p \leftrightarrow [i]p \leftrightarrow \langle i \rangle p$ holds. In the next paragraph I will define a similar property, almost bending back, of those models where each state is at most allowed to see bending back states.

Proposition 4.2.1 *Let $\mathcal{F}$ be a frame. Then $p \to \Box p$ is valid on $\mathcal{F}$ iff $\mathcal{F}$ is a bending back frame.*

Proof. Assume, $\mathcal{F}$ is a bending back frame. Let V be an arbitrary valuation on $\mathcal{F}$ and w a state in $\mathcal{F}$ such that $(\mathcal{F}, V), w \models p$. We need to show that $\Box p$ holds at w. For any u in $\mathcal{F}$ such that wRu we have that $w = u$. So $(\mathcal{F}, V), w \models \Box p$.

For the other direction, assume $\mathcal{F}$ is not a bending back frame, i.e. there are w, u in the frame such that $w \neq u$ and wRu. Now regard a valuation V on $\mathcal{F}$, such that $V(p) = \{w\}$. Then we have $(\mathcal{F}, V), w \models p$ but $(\mathcal{F}, V), w \not\models \Box p$. QED

Almost Bending Back

What is interesting for a belief relation is a slightly weaker condition, namely

$$\Diamond(p \land \Diamond q) \to \Diamond(p \land q). \tag{4.1}$$

This corresponds to frames where $\forall x, y, z((xRy \land yRz) \to y = z)$ holds for the relation. Those frames contain dead ends, points that only see themselves and points that see points that only see themselves. I call the property 'almost bending back'.

For a belief relation this means that whatever one thinks possible, one thinks it possible to believe/ know it. This is a very intuitive requirement on epistemic relations, but note that an equivalence relation, as in imperfect information games, is not almost bending back (just in the special case of perfect information). This is a rather strange aspect of (proper) imperfect information: A player can entertain a possibility, but he does not think it possible to be certain of that possibility, viz. to think that this is the case.

Proposition 4.2.2 *Let $\mathcal{F}$ be a frame. Then $\Diamond(p \land \Diamond q) \to \Diamond(p \land q)$ is valid on $\mathcal{F}$ iff $\mathcal{F}$ is almost bending back.*

Proof. Assume $\mathcal{F}$ is an almost bending back frame, V is a valuation for $\mathcal{F}$ and there is a state w such that $(\mathcal{F}, V), w \models \Diamond(p \wedge \Diamond q)$, i.e. there are states v and u in $\mathcal{F}$ such that wRv and vRu and $(\mathcal{F}, V), v \models p \wedge \Diamond q$ and $(\mathcal{F}, V), u \models q$. By almost bending back you have $v = u$. So $(\mathcal{F}, V), v \models p \wedge q$, thus $(\mathcal{F}, V), w \models \Diamond(p \wedge q)$.

For the other direction assume $\mathcal{F}$ is not almost bending back. That is, there are states w, v, u in $\mathcal{F}$ such that wRv and vRu and $v \neq u$. Consider a valuation on $\mathcal{F}$ with $V(p) = \{v\}$ and $V(q) = \{u\}$. Then $w \models \Diamond(p \wedge \Diamond q)$, but $w \not\models \Diamond(p \wedge q)$. QED

Note that both formulas discussed above are Sahlquist formulas (see appendix, definition A.0.13). For instance, consider the almost bending back formula $\Diamond(p \wedge \Diamond q) \to \Diamond(p \wedge q)$. Its antecedent $\Diamond(p \wedge \Diamond q)$ is built up from the atoms p and q by $\wedge$ and existential modal operators; hence it is a Sahlquist antecedent. The consequent $\Diamond(p \wedge q)$ is obviously positive in p and q. So the almost bending back formula is Sahlquist.

It follows that we can apply the Sahlquist Completeness Theorem (appendix, theorem A.0.15), which says that every Sahlquist formula is canonical for the first-order property it defines. So, if we add (almost) bending back to our axiom system, the resulting system is strongly complete with respect to *generalized game models* satisfying (almost) bending back. However, the unraveling procedure used in the proof of proposition 4.1.3 to get completeness with respect to epistemic game models will not preserve (almost) bending back in general. I leave as an open question whether one has completeness of the enriched logic with respect to epistemic game models with (almost) bending back, but I think it rather unlikely to be the case.

4.3 Finite Models and Decidability

Although these topics are important I keep the discussion on finite models and decidability on an informal level. The key question I am addressing here is whether the satisfiability problem for the minimal logic Σ introduced in section 4.1 is decidable. A first simple observation is that a formula φ of $\mathfrak{L}$ is satisfiable on an epistemic game model if and only if φ is satisfiable on a generalized game model. The argument, which we already used in the completeness proof, rests on the fact that every rooted generalized game model is the bounded morphic image of some epistemic game model (see proposition 3.2.1). So, if φ is satisfied at a node of a generalized game model then φ is satisfiable on the submodel generated by this node, and, consequently, φ is satisfiable on an epistemic game model (since satisfaction is preserved under bounded morphisms).

Since our logic is finitely axiomatizable, it is decidable if it has the finite model property – see appendix, theorem A.0.17. For the finite model property we need to show that every formula φ of $\mathfrak{L}$ satisfiable in a generalized game model is satisfiable in a *finite* generalized game model. The classical construction to build finite models is *filtration*.[2]

[2]see [7, section 2.3].

Given a generalized game model $\mathbb{M}$ on which φ is satisfiable, filtration of $\mathbb{M}$ through the set Γ_φ of subformulas of φ gives us a finite general model $\mathbb{M}^f$, the *filtration of* $\mathbb{M}$ *through* Γ_φ, on which φ is satisfiable. What filtration essentially does is to identify all nodes of $\mathbb{M}$ that are indistinguishable by any of the subformulas of φ. However, while filtration preserves the seriality of the epistemic relations, the structural constraints concerning the propositional constants turn_i and u will be destroyed in general. So either we find a way to transform $\mathbb{M}^f$ into a generalized game model (without loosing satisfiability of φ) or we make use of another method for constructing finite models. Two possible candidates are the technique of *selection* and the construction of *finite canonical models*.[3] I leave the problem of building finite generalized game models as a topic for future work. (Another interesting question is of course whether it is possible to get even finite epistemic game models.)

[3]see [7, section 2.3] and [7, section 4.8], respectively.

Chapter 5

Extended Languages

In this chapter, I will discuss an extended version of the language introduced in chapter 2. The extended language will allow us to express some new properties of game models, like being in a one tree game (see 5.3). Of course, you do not need to add all the new operators in one go; for the formulas discussed in section 5.5 you just need the inverse action relations and the Kleene star. Defining a game model by just one formula takes more heavy machinery: you need a modality that can reach every state in the game part of a model.

5.1 The Language $\mathfrak{L}^*$

Definition 5.1.1 (Alphabet) The alphabet of $\mathfrak{L}^*$ consists of

a) propositional constants: turn_i $(i \in \mathsf{N})$, elements of U, $\top$

b) propositional variables: elements of Φ

c) modal operators: $[a], [a]^{-1}(a \in \mathsf{A}), \Box, \Box^{-1}, [i](i \in \mathsf{N}), [*], [*]^{-1}$, C, P.

$\lhd$

Definition 5.1.2 (Formulas) The formulas of $\mathfrak{L}^*$ are of the form

$$p \mid \top \mid \text{turn}_i \mid u \mid \neg\varphi \mid \varphi \wedge \psi \mid [i]\varphi \mid [a]\varphi \mid [a]^{-1}\varphi \mid \Box\varphi \mid \Box^{-1}\varphi \mid [*]\varphi \mid [*]^{-1}\varphi \mid \mathsf{C}\varphi \mid \mathsf{P}\varphi$$

where p ranges over elements of Φ and a ranges over A. $\lhd$

Before we can define the semantics of the new modalities, we need a way to talk about the part of the model that is epistemically relevant for the real game, i.e. the part of the model the P-modality is supposed to reach.

Definition 5.1.3 (B-S-path) Let $\mathcal{M}$ be an epistemic game model $(\mathcal{G}, (\mathcal{E}_j)_{j\in J}, (B_i)_{i\in \mathsf{N}})$ with $\biguplus_j \mathcal{E}_j = (T, (S_a)_{a\in\mathsf{A}}, (C_i)_{i\in\mathsf{N}}, (D_u)_{u\in\mathsf{U}}, V)$. Given $t, t' \in T$, there exists a *B-S-path from t to t'* if there is a sequence $t_0, \ldots, t_n$ in T such that $t_0 = t$, $t_n = t'$, and for all k, $0 \le k < n$,

- there is an $i \in \mathsf{N}$ with $t_k B_i t_{k+1}$ or
- there is an $a \in A$ with $t_k S_a t_{k+1}$ or $t_k S_a^{-1} t_{k+1}$.

◁

In general, for a relation R, an *R-path* is a chain of R-connected nodes. Note that in the case of an B-S-path you can use the inverse of the action relation as well. With the help of the notion of a B-S-path we can define now the part of a game model that is relevant for the real game.

Definition 5.1.4 (Game Part) Let $\mathcal{M} = (\mathcal{G}, (\mathcal{E}_j)_{j\in J}, (B_i)_{i\in\mathsf{N}})$ be an epistemic game model and T_j the state set of $\mathcal{E}_j$. Let K be the least subset of J such that $\biguplus_{j\in K} T_j$ contains all nodes which are reachable from the root of $\mathcal{G}$ by a B-S-path. Then the *game part of* $\mathcal{M}$ is the epistemic game model $(\mathcal{G}, (\mathcal{E}_j)_{j\in K}, (B'_i)_{i\in\mathsf{N}})$ where B'_i is the restriction of B_i to a relation on $\biguplus_{j\in K} T_j$. ◁

For the definition of truth in a model, be reminded that for a binary relation R, R^* denotes the reflexive and transitive closure of R.

Definition 5.1.5 (Truth) Let $\mathcal{M}$ be an epistemic game model $(\mathcal{G}, (\mathcal{E}_j)_{j\in J}, (B_i)_{i\in\mathsf{N}})$ with $\biguplus_j \mathcal{E}_j = (T, (S_a)_{a\in\mathsf{A}}, (C_i)_{i\in\mathsf{N}}, (D_u)_{u\in\mathsf{U}}, V)$. The truth conditions for a formula φ of $\mathfrak{L}^*$ are the ones given in definition 2.2.3 plus the following (with $S = \bigcup_{a\in\mathsf{A}} S_a$ and $B = \bigcup_{i\in\mathsf{N}} B_i$):

$$
\begin{array}{lll}
\mathcal{M}, t \models [a]^{-1}\varphi & \text{iff} & \mathcal{M}, t' \models \varphi \text{ for all } t' \in T \text{ such that } t' S_a t \\
\mathcal{M}, t \models \Box^{-1}\varphi & \text{iff} & \mathcal{M}, t' \models \varphi \text{ for all } t' \in T \text{ such that } t' S t \\
\mathcal{M}, t \models [*]\varphi & \text{iff} & \mathcal{M}, t' \models \varphi \text{ for all } t' \in T \text{ such that } t S^* t' \\
\mathcal{M}, t \models [*]^{-1}\varphi & \text{iff} & \mathcal{M}, t' \models \varphi \text{ for all } t' \in T \text{ such that } t' S^* t \\
\mathcal{M}, t \models \mathsf{C}\varphi & \text{iff} & \mathcal{M}, t' \models \varphi \text{ for all } t' \in T \text{ such that } t B^* t' \\
\mathcal{M}, t \models \mathsf{P}\varphi & \text{iff} & \mathcal{M}, t' \models \varphi \text{ for all } t' \text{ such that there is a } B\text{-}S\text{-path from } t \text{ to } t'
\end{array}
$$

◁

Note that if you evaluate P within the real game, it ranges over the game part of the model. P is evaluated by going along all action and epistemic relations. So you could suppose that it can be defined by all action modalities plus common knowledge, if you allow use of the power of PDL, i.e. the possibility to apply the Kleene star to a union of action and epistemic relations. This is not the case: If you have an epistemic alternative in your game part that is not reached by an epistemic relation at the root, this root cannot be reached by actions and epistemic moves. In contrast, if you add reverse actions to your fund of PDL modalities, you have enough to define P. Note that this is still not possible in my account; it lacks rules for uniting modalities and applying the Kleene star from outside.

As an example of what we can express now, let us go back to an example in chapter 3: With upward looking modalities, the triangled states in figure 3.3 do have different theories: for example $\neg\Diamond^{-1}\text{turn}_E$ holds at the left state, while $\Box^{-1}\bot$ holds at the right state, and thus in particular $\Box^{-1}\text{turn}_E$.

5.2 Referring to the Past

With the new modalities, we can directly talk about the run of the game so far and what other turns one might have taken before. We get more restrictions on bisimulations as well. A *bisimulation in* $\mathfrak{L}^*$ satisfies all conditions of definition 3.1.1 plus the following:

1. If tZt' and $tS_a^{-1}s$ then there is an $s' \in T'$ such that sZs' and $t'S_a^{-1}s'$.
2. If tZt' and $tS_a^{*}s$ then there is an $s' \in T'$ such that sZs' and $t'S_a'^{*}s'$.
3. If tZt' and $t'S_a'^{*}s'$ then there is a $s \in T$ such that sZs' and $tS_a^{*}s$.

If you look more closely at these last two conditions, you see that they are superfluous: since $tS_a^{*}s$ implies there is an $n \in \mathbb{N}$ such that $tS_a^{n}s$, by 3.1.1-2 you even get there is an $s' \in T'$ such that $t'S_a'^{n}s'$ for the same n – which in turn implies $t'S_a'^{*}s'$. It works analogous with condition 2. On the same grounds we do not need to define bisimulation for C and P. Having a bisimulation that goes through the model step by step implies a bisimulation for these global modalities.

With the new modalities we can express properties of trees we could not talk about before, e.g. upward linearity. Since our upward looking modality $[a]^{-1}$ is not supposed to be transitive this amounts to saying that whenever you can go upwards by a reverse a action, there is just one state you can go to. In a formula: $\langle a\rangle^{-1}p \wedge \langle a\rangle^{-1}q \rightarrow \langle a\rangle^{-1}(p \wedge q)$.

5.3 Infinite Iterations

An axiom system for $\mathfrak{L}^*$ is given in table 5.1.

Models Compared Again

With the new modalities at our disposal we can go back to the comparison between epistemic game models, simple models and imperfect information games and gather their differences axiomatically.

Axioms with one modality	
	all axioms for $\mathfrak{L}$
$[a]^{-1}$-normality	$[a]^{-1}(p \to q) \to ([a]^{-1}p \to [a]^{-1}q)$
$\Box^{-1}$-normality	$\Box^{-1}(p \to q) \to (\Box^{-1}p \to \Box^{-1}q)$
$[*]$-normality	$[*](p \to q) \to ([*]p \to [*]q)$
$[*]^{-1}$-normality	$[*]^{-1}(p \to q) \to ([*]^{-1}p \to [*]^{-1}q)$
C-normality	$\mathsf{C}(p \to q) \to (\mathsf{C}p \to \mathsf{C}q)$
P-normality	$\mathsf{P}(p \to q) \to (\mathsf{P}p \to \mathsf{P}q)$
$[*]$-transitivity	$[*][*]p \to [*]p$
$[*]^{-1}$-transitivity	$[*]^{-1}[*]^{-1}p \to [*]^{-1}p$
C-transitivity	$\mathsf{CC}p \to \mathsf{C}p$
P-transitivity	$\mathsf{PP}p \to \mathsf{P}p$
Interrelation Axioms	
$[a]$-inversion axioms	$\neg p \to [a]\neg[a]^{-1}p$ and $\neg p \to [a]^{-1}\neg[a]p$
$\Box$-inversion axioms	$\neg p \to \Box\neg\Box^{-1}p$ and $\neg p \to \Box^{-1}\neg\Box p$
$[*]$-inversion axioms	$\neg p \to [*]\neg[*]^{-1}p$ and $\neg p \to [*]^{-1}\neg[*]p$
$[*]$-mix axiom	$\langle * \rangle p \leftrightarrow (p \vee \Diamond\langle * \rangle p)$
$[*]$-induction axiom	$[*](p \to \Box p) \to (p \to [*]p)$
$[*]^{-1}$-mix axiom	$\langle * \rangle^{-1} p \leftrightarrow (p \vee \Diamond\langle * \rangle^{-1} p)$
$[*]^{-1}$-induction axiom	$[*]^{-1}(p \to \Box^{-1}p) \to (p \to [*]^{-1}p)$
Modal Rules	
	all rules for $\mathfrak{L}$
$[a]^{-1}$-necessitation	From $\vdash \varphi$ infer $\vdash [a]^{-1}\varphi$
$\Box^{-1}$-necessitation	From $\vdash \varphi$ infer $\vdash \Box^{-1}\varphi$
$[*]$-necessitation	From $\vdash \varphi$ infer $\vdash [*]\varphi$
$[*]^{-1}$-necessitation	From $\vdash \varphi$ infer $\vdash [*]^{-1}\varphi$
C-necessitation	From $\vdash \varphi$ infer $\vdash \mathsf{C}\varphi$
P-necessitation	From $\vdash \varphi$ infer $\vdash \mathsf{P}\varphi$

Figure 5.1: A logical system for $\mathfrak{L}^*$

With the possibility of looking up the tree by inverse action relations, the technical trick of turning every epistemic game model into a simple model by adding a new, common root to all trees in the game (see 3.2.2) does not yield an even locally bisimilar game anymore, because now you see the common root from further down the tree.

With the extended language we can express the property of being a simple epistemic game model within our language (compare to 3.2.2). This is expressed by the following formula

$$\langle i\rangle\varphi \rightarrow \langle * \rangle^{-1}\langle * \rangle\varphi, \tag{5.1}$$

where all modalities in φ are action modalities.

If you drop the restriction on the modalities in φ, you even get something stronger: whatever you believe possible at a state t in a game $\mathcal{G}$ is something that is in that same game $\mathcal{G}$. That is, you enforced the game-part of the model to consist of just one tree. Thus we call the property expressed by the formula 5.1 the property of *staying in the tree*. We can now restate proposition 3.2.4. Game models that can be represented by one game tree are those where (5.1) holds.

Now we have a characterization of imperfect information games: An imperfect information game is a one tree game with epistemic S5 relations.

Finiteness

Is our extended language expressible enough to characterize finite trees? The answer is no, for a simple reason: the starred modalities are reflexive. Unraveling a finite tree with a reflexive relation gets you an infinite tree that is bisimilar to the finite one. So for finiteness you have to introduce a weaker modality, [+], the transitive (but not reflexive) closure of $\Box$. Its semantics for a game model $(\mathcal{G}, (\mathcal{E}_j)_{j\in J}, (B_i)_{i\in\mathsf{N}})$ with $\biguplus_j \mathcal{E}_j = (T, (S_a)_{a\in\mathsf{A}}, (C_i)_{i\in\mathsf{N}}, (D_u)_{u\in\mathsf{U}}, V)$ are given by

$$\mathcal{M}, t \models [+]\varphi \text{ iff } \mathcal{M}, t' \models \varphi \text{ for all } t' \in T \text{ such that } tS^{+}t'.$$

Now the Löb axiom $[+]([+]p \rightarrow p) \rightarrow [+]p$ together with the minimal logical system from chapter 4 gives a sound and weakly complete logic with respect to the class of models based on finite transitive trees. We have just weak completeness because the class of transitive finite trees is not first-order definable; compactness fails (compare [7], p. 130 ff.[1]). Note that this finiteness does not mean 'a finite number of nodes'. For this you would have to require branching to be finite, as well. In order to characterize that modally, you need modalities capable of referring to sister nodes in a tree. Languages like that are discussed in section 5.6.

[1]There the class of models is not restricted to trees, but the arguments can be used in the same way.

5.4 A State Definition Lemma

With the path-modality at our hands, we can define the whole game part of a model by one formula up to bisimulation.

Lemma 5.4.1 (State Definition Lemma)[2] *Assume $\mathfrak{L}$ contains only finitely many propositional variables. For each finite model $\mathcal{M} = (\mathcal{G}, (\mathcal{E}_j)_{j\in J}, (B_i)_{i\in\mathsf{N}})$ with $\biguplus_j \mathcal{E}_j = (T, (S_a)_{a\in\mathsf{A}}, (C_i)_{i\in\mathsf{N}}, (D_u)_{u\in\mathsf{U}}, V)$ and node $s \in T$ there is a $\mathfrak{L}$-formula β, such that the following are equivalent:*

1. $\mathcal{N}, t \models \beta$
2. $\mathcal{M}, s \leftrightarroweq \mathcal{N}, t$.

Proof. Note that, since the model is finite, all relations on it are image finite.

Let $\mathcal{M} = (\mathcal{G}, (\mathcal{E}_j)_{j\in J}, (B_i)_{i\in\mathsf{N}})$ be a model with $\biguplus_j \mathcal{E}_j = (T, (S_a)_{a\in\mathsf{A}}, (C_i)_{i\in\mathsf{N}}, (D_u)_{u\in\mathsf{U}}, V)$ and let s be a node in T. Without loss of generality assume that $\mathcal{M}$ is its own $\mathcal{G}$-part.

Claim 1 There exist a finite set of formulas $\varphi_k, 1 \leq k \leq n$ such that

1. each node in $\mathcal{M}$ satisfies exactly one of them and
2. if two nodes satisfy the same formula φ_k then they agree on all formulas.

Proof of Claim Let t be a node in T. For each $t' \in T$ let $\delta_{t,t'}$ be a formula that is true in t but not in t', and $\top$ if such a formula does not exist. Let φ_k be the conjunction of all $\delta_{t,t'}$ and all (negative) propositional variables true at t. ◀

Before we construct β, one more remark: If any node satisfying a formula φ_k is modally (by action or epistemic access) linked to a world that satisfies φ_ℓ (where ℓ need not be different from k), then all nodes satisfying φ_k satisfy $\langle a\rangle\varphi_\ell/\langle i\rangle\varphi_\ell$ resp. For otherwise they would not satisfy the same formulas.

Now, let $\beta_{\mathcal{M},s}$ be the conjunction of

- the unique φ_k true at s,
- P of
  - $\bigvee_{t\in T} \varphi_k$,
  - $\bigwedge_{k\neq\ell} \neg(\varphi_k \wedge \varphi_\ell)$,
  - all implications of the form $\varphi_k \to \langle i\rangle\varphi_\ell$ and $\varphi_k \to \langle a\rangle\varphi_\ell$ that hold in all nodes in $\mathcal{M}$ and

[2] cf. van Benthem [4], p. 13f. The original result is due to Alexandru Baltag.

– all implications $\varphi_k \to [i] \bigvee \varphi_\ell$ and $\varphi_k \to [a] \bigvee \varphi_\ell$ where the disjunctions run over all situations listed in the previous clause.

Note that $\beta_{\mathcal{M},s}$ is still a finite conjunction, since there are just finitely many φ_k to regard.

Claim 2 $\mathcal{M}, s \models \beta_{\mathcal{M},s}$.

This is by construction of $\beta_{\mathcal{M},s}$.

Claim 3 If $\mathcal{N}, t \leftrightarroweq \mathcal{M}, s$ then $\mathcal{N}, t \models \beta_{\mathcal{M},s}$.

Proof of Claim If $\mathcal{N}, t \leftrightarroweq \mathcal{M}, s$, then by proposition 3.1.6 they have the same theory, i.e. $\mathcal{N}, t \models \beta_{\mathcal{M},s}$ iff $\mathcal{M}, s \models \beta_{\mathcal{M},s}$. ◀

Claim 4 If $\mathcal{N}, t \models \beta_{\mathcal{M},s}$ then there is a bisimulation between $\mathcal{N}, t$ and $\mathcal{M}, s$.

Proof of Claim Assume at a state t in the model $\mathcal{N}$ $\beta_{\mathcal{M},s}$ holds. Without loss of generality let $\mathcal{N}$ be its own game-part. The φ_k partition $\mathcal{N}$ into disjoint zones Z_k of worlds satisfying these formulas. Now relate all the worlds in such a zone to all worlds that satisfy φ_k in the model $\mathcal{M}$. In particular, t gets connected to s. It is left to check that this connection is a bisimulation. The atomic clause is clear by definition of φ_k. But also, the back and forth conditions follow from the given description:

- Any R_a- or B_i-successor step has been encoded in a formula $\varphi_k \to \langle a \rangle \varphi_\ell$, $\varphi_k \to \langle i \rangle \varphi_\ell$ resp. which holds everywhere in $\mathcal{N}$ producing the required successor there.
- Conversely, if there is no S_a/B_i-successor in $\mathcal{M}$, this shows up in the formulas $\varphi_k \to [a] \bigvee \varphi_\ell / \ \varphi_k \to [i] \bigvee \varphi_\ell$, which hold also in $\mathcal{N}$, so that there is no 'excess' successor there either.

◀

QED

5.5 Talking About Trees

In this section we briefly discuss some new properties of an epistemic game model we can express now.

Getting Noticed

The formula

$$\varphi \to \langle * \rangle^{-1} \langle * \rangle \langle i \rangle \varphi$$

expresses a non-ignorance property: each state in the game is at least somewhere considered possible by player i. This ensures that i cannot be unaware of parts of the game (i.e. it excludes games of the kind presented in chapter 1 as changing winners).

Complete Uncertainty

Assume you have a complete information game, that is - from the point of view of frames - the epistemic relations never leave the game itself. Then the worst case of uncertainty you can have is considering all states in the game possible. This is expressed by

$$\langle * \rangle^{-1} \langle * \rangle \varphi \to \langle i \rangle \varphi.$$

There is a reverse worst case. Wherever a player is in the game, he believes to be at one node, the node where the formula is evaluated:

$$\langle * \rangle^{-1} \langle * \rangle \langle i \rangle \varphi \to \varphi.$$

Small Trees

The extended language allows you even to 'count' nodes. The formula

$$\langle * \rangle^{-1} \langle * \rangle \varphi \to \varphi$$

characterizes the frame property of having at most one node in the tree. More generally, the formula

$$\langle * \rangle^{-1} \langle * \rangle \varphi_1 \wedge \cdots \wedge \langle * \rangle^{-1} \langle * \rangle \varphi_n \to \varphi_1 \vee \cdots \vee \varphi_n$$

characterizes tree frames with at most n nodes.

Again, you can have epistemic versions of this. A player can believe there are at most n nodes in the game (this is expressed by putting the formula under the scope of an epistemic boxed modality). Another limitation of a player is expressed by

$$\langle i \rangle \langle * \rangle^{-1} \langle * \rangle \varphi \to \varphi.$$

This means that all the player thinks possible somewhere in the game, holds already at the state he is in.

5.6 Other Tree Languages

The modalities I discussed are not fit to describe intrinsic features of a tree, e.g. being a sister node. In [8] Blackburn, Meyer-Viol and de Rijke define a language $\mathfrak{L}_B$ for trees that can reach sister nodes directly. They propose a fragment of PDL that has programs for going upwards, downwards, left and right and their Kleene-starred versions. You can regard a bigger fragment, developed by Palm, that has a test program as well. This program gets just executed if the test turned out positive. In [6] Blackburn, Gaiffe and Marx prove that Palm's language is as expressible as $\mathfrak{L}_B$ with four until operators added. You can add the full power of all PDL programs to the four tree operations. This has been done by Kracht. With that, you can even express some second order properties of nodes.

So what would you get by adding PDL-style modalities to epistemic game models? Identifying a sister node is not of much interest in analyzing games; the order in which you represent possible actions in a tree when you draw it is completely arbitrary. It is important *what* options you have for planning your strategies, but not their order in a game tree. On the other hand, it is interesting for complexity considerations, since now you can control branching. A test modality is interesting for game models as such. In the next chapter I discuss a dynamic approach to epistemic models, which uses tests (or preconditions) for giving a condition when to carry out an action.

Chapter 6

Update

So far, the epistemic game models considered have been static. This has several advantages. For one, we can look at a whole game model (with all its possible runs) and evaluate strategies. Furthermore it is no problem to distribute actions and epistemic relations freely, whatever strange situation we want to model. On the other hand, dynamic models, while being unwieldy for developing strategies, allow you to *calculate* each step of a game. So if you want certain restrictions to hold, you can incorporate them elegantly into the rules for the calculation. In addition, starting to model a game at some initial position and then developing it along the course of actions keeps more of the flavor of actually playing the game.

In their paper 'The Logic of Public Announcement, Common Knowledge and Private Suspicions' ([1]) Baltag, Moss and Solecki (BMS) develop a skilful way to model epistemic updates. They merge two kinds of models: an epistemic model and an action structure. With BMS, actions are epistemic updates (though they do not restrict their model in principle to actions that do not change the facts (compare [1], p. 87)), like public or private announcements. In particular, actions like cheating or suspicions can be modeled in their system.

In the next section I will give a brief outline of the BMS system.

6.1 Outline of the BMS System

6.1.1 The State Model

First, BMS fix a set $AtSen$ of atomic sentences and a set $\mathcal{A}$ of agents. The presentation is based on state models. The idea behind a state model is to give a model for all epistemic relations at one stage. Worlds are possible states of affairs (i.e. sets of atomic sentences), connected by epistemic relations, so called *accessibility relations*, for each agent. A state s is i-connected to a state t iff agent i thinks it possible at s that t is the actual state. Note that BMS do not give a modality for knowledge, just for belief. Of course, if there is just one belief option, this will feel like knowledge to the agent whether or not he is right in his belief. Technically, a *state model* looks as follows

$$\mathbf{S} = (S, \xrightarrow{\mathcal{A}}_{\mathbf{S}}, \|.\|_{\mathbf{S}}),$$

where $\mathbf{S}$ is a set of states, $\xrightarrow{\mathcal{A}}_{\mathbf{S}} = (\xrightarrow{A}_{\mathbf{S}} \subseteq \mathbf{S} \times \mathbf{S})_{A \in \mathcal{A}}$ and $\|.\|_{\mathbf{S}}$ is a valuation function from the set of atomic sentences to $\mathfrak{P}(S)$. The arrows do not underlie any restriction, in particular they need not be equivalence relations. In their examples, BMS just regard the arrow-connected component of the actual state.

As an illustration, regard an example BMS give in their paper. They model the following situation: a coin has been thrown, two agents, A and E, have seen this, but neither knows whether the coin lies up heads or tails now. The state model $\mathbf{S}$ for this situation consists of two states, a and b, an equivalence relation for both agents and a valuation function with $\|heads\|_{\mathbf{S}} = \{a\}$ and $\|tails\|_{\mathbf{S}} = \{b\}$. This is pictured in figure 6.1.

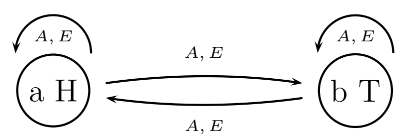

Figure 6.1: The state model $\mathbf{S}$

So far, we can just talk about the other states within the same state model. Of course, BMS represent the rest of the game as well. They define an epistemic relation between two state models.[1] The epistemic relation holds after some kind of epistemic update has taken place. More formally, an *epistemic relation* between two state models $\mathbf{S}$ and $\mathbf{T}$ is a relation $r \subseteq S \times T$. To indicate the state models it is written $r : \mathbf{S} \to \mathbf{T}$. Now we can define an *update* as the pair $\mathfrak{r} = (\mathbf{S} \mapsto \mathbf{S}(\mathfrak{r}), \mathbf{S} \mapsto \mathfrak{r}_{\mathbf{S}})$ consisting of an *update map*, $\mathbf{S} \mapsto \mathbf{S}(\mathfrak{r})$, and an *update relation*, $\mathbf{S} \mapsto \mathfrak{r}_{\mathbf{S}}$, where $\mathfrak{r}_{\mathbf{S}}$ is an epistemic relation.

The semantic counterpart of an atomic sentence is an atomic proposition; i.e. the set of states where the atomic sentence holds. From atomic propositions BMS build epistemic propositions the usual way, including an operator for common knowledge.[2]

6.1.2 Simple Action Structures

In the next step, BMS introduce a way to calculate an update, which so far has been just arbitrary. What they introduce is a second structure, a simple action structure.

[1]The name 'epistemic' stems from BMS dealing with epistemic actions only, it is not to be confused with epistemic relations in an epistemic game model.

[2]or, more accurately, common belief.

This action structure keeps an eye on what actions are possible in a given state, i.e. for each action there is a precondition that has to hold at a state for the action to take place. In symbols, we have a *simple action structure* $\mathbf{\Sigma} = (\Sigma, \stackrel{\mathcal{A}}{\rightarrow}, \text{pre})$, where Σ is a set of simple actions, $\stackrel{\mathcal{A}}{\rightarrow} = \{\stackrel{A}{\rightarrow} | A \in \mathcal{A}\}$ and pre : $\Sigma \rightarrow \Phi$ with Φ the collection of all epistemic propositions.

As an example, BMS have the simple action structure consisting of the action σ, of A looking at the coin that shows heads without E noticing, and the action τ that nothing happens. We have that A accesses at each state just that state, while E in both states thinks to be in the τ state. Precondition for A cheating is that the coin actually shows heads, the other action has an empty precondition. This is pictured in figure 6.2.

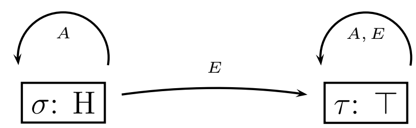

Figure 6.2: The action signature $\mathbf{\Sigma}$

Now for the calculation. If you merge a state model $\mathbf{S} = (S, \stackrel{\mathcal{A}}{\rightarrow}_{\mathbf{S}}, [\![.]\!]_{\mathbf{S}})$ with a simple action structure $\mathbf{\Sigma} = (\Sigma, \stackrel{\mathcal{A}}{\rightarrow}, \text{pre})$, you get out a new state model, their *product update*. The new states are those elements of the Cartesian products where the precondition for an action is fulfilled at a state in the old state model. An accessibility relation holds if it holds for both pairs that go into the new states. The valuation of atomic sentences is just taken over from the old state model. This gets calculated as follows:

$$\mathbf{S} \otimes \mathbf{\Sigma} = (S \otimes \Sigma, \stackrel{\mathcal{A}}{\rightarrow}_{\mathbf{S}\otimes\mathbf{\Sigma}}, [\![.]\!])_{\mathbf{S}\otimes\mathbf{\Sigma}},$$

with $S \otimes \Sigma = \{(s, \sigma) \in S \times \Sigma | s \in [\![\text{pre}(\sigma)]\!]_{\mathbf{S}}\}$. For $(s, \sigma), (t, \tau) \in S \times \Sigma$ we have $(s, \sigma) \stackrel{A}{\rightarrow}_{\mathbf{S}\otimes\mathbf{\Sigma}} (t, \tau)$ iff $s \stackrel{A}{\rightarrow}_{\mathbf{S}} t$ and $\sigma \stackrel{A}{\rightarrow} \tau$. The valuation $[\![.]\!] : AtSen \rightarrow \mathfrak{P}(S \otimes \Sigma)$ is defined by $[\![p]\!]_{\mathbf{S}\otimes\mathbf{\Sigma}} = \{(s, \sigma) \in S \otimes \Sigma | s \in [\![p]\!]_{\mathbf{S}}\}$.

Taking the update product of both examples from above, we get a state model $\mathbf{T}$ with

- $T = \{(a, \sigma), (a, \tau), (b, \tau)\}$, where we call (a, σ) c, (a, τ) d and (b, τ) e,
- $\stackrel{A}{\rightarrow}_{\mathbf{T}} = \{(c, c), (d, d), (e, e)\}$,
- $\stackrel{E}{\rightarrow}_{\mathbf{T}} = \{(c, d), (c, e), (d, d), (e, e), (d, e), (e, d)\}$,
- $[\![heads]\!]_{\mathbf{T}} = \{c, d\}$ and

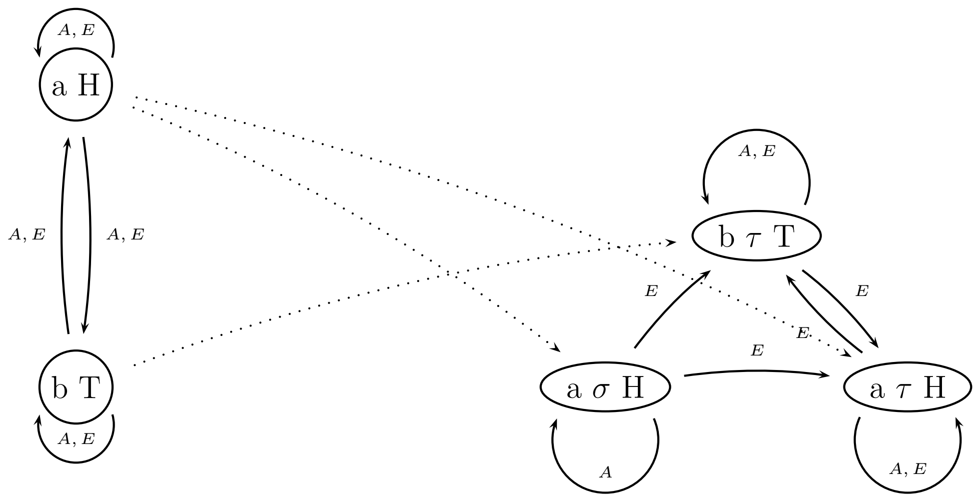

Figure 6.3: The update product

- $[\![tails]\!]_{\mathbf{T}} = \{e\}$.

An *epistemic action model* is of the form $(\mathbf{\Sigma}, \Gamma)$, where $\mathbf{\Sigma}$ is a simple action structure and Γ a set of *designated simple actions*. The idea behind this is that $\gamma \in \Gamma$ is a possible 'deterministic resolution' of a non-deterministic action α. Given an epistemic action model $(\mathbf{\Sigma}, \Gamma)$ an update is defined by $\mathbf{S}(\mathbf{\Sigma}, \Gamma) = \mathbf{S} \otimes \mathbf{\Sigma}$ and $s(\mathbf{\Sigma}, \Gamma)_{\mathbf{S}}(t, \sigma)$ iff $s = t$ and $\sigma \in \Gamma$. This update gets denoted $(\mathbf{\Sigma}, \Gamma)$ as well.

In figure 6.4 you see how the update product would look pictured as an epistemic game model. You could add turns for 'nature' (throwing the coin) and A (cheating), but there obviously are no utilities. But they are not needed yet – the game is not over and we just computed one further step with the update product.

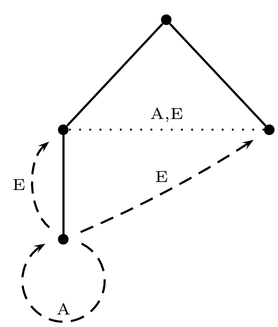

Figure 6.4: $\mathbf{S} \otimes \mathbf{\Sigma}$ as an epistemic game model

6.1.3 Language and Axioms

BMS give several languages for their models. I will present here the language $\mathfrak{L}(\Sigma)$. $\mathfrak{L}(\Sigma)$ (where Σ is an action signature) is build of sentences φ of the form

$$\top \mid p_i \mid \neg\varphi \mid \varphi \wedge \psi \mid \Box_A \varphi \mid \Box^*_{\mathcal{B}} \varphi \mid [\pi]\varphi,$$

and actions (or programs) π of the form

$$skip \mid crash \mid \sigma\psi_1, \ldots, \psi_n \mid \pi + \rho \mid \pi \cdot \rho \mid \pi^*.$$

Here $\Box^*_{\mathcal{B}}$ is the modality for common knowledge among the members of the group $\mathcal{B}$. The semantics of $\mathfrak{L}(\Sigma)$ is given as follows for sentences and actions, respectively:

$$\begin{array}{lcl}
[\![\top]\!] & = & \mathbf{True} \\
[\![p]\!] & = & \mathbf{p} \\
[\![\varphi \wedge \psi]\!] & = & [\![\varphi]\!] \wedge [\![\psi]\!] \\
[\![\neg\varphi]\!] & = & \neg[\![\varphi]\!] \\
[\![\Box_A\varphi]\!] & = & \Box_A[\![\varphi]\!] \\
[\![\Box^*_{\mathcal{B}}\varphi]\!] & = & \Box^*_{\mathcal{B}}[\![\varphi]\!] \\
[\![[\pi]\varphi]\!] & = & [[\![\pi]\!]][\![\varphi]\!]
\end{array}$$

$$\begin{array}{lcl}
[\![skip]\!] & = & \mathbf{1} \\
[\![crash]\!] & = & \mathbf{0} \\
[\![\sigma_i\psi_1 \ldots \psi_n]\!] & = & (\mathbf{\Sigma}, \sigma_i)([\![\psi_1]\!], \ldots, [\![\psi_n]\!]) \\
[\![\pi \cdot \rho]\!] & = & [\![\pi]\!] \cdot [\![\rho]\!] \\
[\![\pi + \rho]\!] & = & [\![\pi]\!] + [\![\rho]\!] \\
[\![\pi^*]\!] & = & [\![\pi^*]\!]
\end{array}$$

6.2 BMS in Games

At first glance, it looks like BMS models and epistemic game models do not have much to do with each other. Looking a bit closer, they do look very similar indeed, with the only difference that BMS start out with epistemic relations and add actions, while epistemic game models are based on trees to which you add epistemic relations.

Both accounts are oversimplified. You can certainly write down a static version of every BMS update product, but you lack a criterion what the 'whole' game shall look like. What kind of actions are allowed? Who is taking them? And what is the aim of playing?

On the other hand, you can model parts of an epistemic game model dynamically. But your dynamic state models might not go conform with intuitions about what should be modeled in each state model. For example, you cannot always get the whole next layer of the game. In the next section, I give an example of a game that you can

Basic Axioms	
	all sentential tautologies
$[\pi]$-normality	$[\pi](p \to q) \to ([\pi]p \to [\pi]q)$
$\Box_A$-normality	$\Box_A(p \to q) \to (\Box_A p \to \Box_A q)$
$\Box^*_{\mathcal{C}}$-normality	$\Box^*_{\mathcal{C}}(p \to q) \to (\Box^*_{\mathcal{C}} p \to \Box^*_{\mathcal{C}} q)$
action mix axiom	$\Box p \leftrightarrow \bigwedge_{a \in \mathsf{A}}[a]p$
Action Axioms	
Atomic Permanence	$[\sigma_i \vec{\psi}]p \leftrightarrow \psi_i\ top$
Partial Functionality	$[\sigma_i \vec{\psi}]\neg\chi \leftrightarrow (\psi_i \to \neg[\sigma_i \vec{\Psi}]\chi)$
Action-Knowledge	$[\sigma_i \vec{\psi}]\Box_A \varphi \leftrightarrow (\psi_i \to \bigwedge\{\Box_A[\sigma_j \vec{\psi}]\varphi \mid \sigma_i \xrightarrow{A} \sigma_j \text{ in } \mathbf{\Sigma}\})$
Action Mix Axiom	$[\pi^*]\varphi \to \varphi \wedge [\pi][\pi^*]\varphi$
Epistemic Mix Axiom	$\Box^*_{\mathcal{C}}\varphi \to \varphi \wedge \bigwedge\{\Box_a \Box^*_{\mathcal{C}}\varphi \mid A \in \mathcal{C}\}$
Skip Axiom	$[skip]\varphi \leftrightarrow \varphi$
Crash Axiom	$[crash]\bot$
Composition Axiom	$[\pi][\rho]\varphi \leftrightarrow [\pi \cdot \rho]\varphi$
Choice Axiom	$[\pi + \rho]\varphi \leftrightarrow [\varphi] \wedge [\rho]\varphi$
Modal Rules	
modus ponens	From $\vdash \varphi$ and $\vdash \varphi \to \psi$ infer $\vdash \psi$
$[\pi]$-necessitation	From $\vdash \varphi$ infer $\vdash [\pi]\varphi$
$\Box_A$-necessitation	From $\vdash \varphi$ infer $\vdash \Box_A \varphi$
$\Box^*_{\mathcal{C}}$-necessitation	From $\vdash \varphi$ infer $\vdash \Box^*_{\mathcal{C}}\varphi$
Program Induction Rule	From $\vdash \chi \to \psi \wedge [\pi]\chi$, infer $\vdash \chi \to [\pi^*]\psi$
Action Rule	Let ψ be a sentence, let α be an action, and let $\mathcal{C}$ be a set of agents. Let there be sentences χ_β for all β such that $\alpha \to^*_{\mathcal{C}} \beta$ (including α itself), and such that 1. $\vdash \chi_\beta \to [\beta]\psi$. 2. If $A \in \mathcal{C}$ and $\beta \xrightarrow{A} \gamma$, then $\vdash (\chi_\beta \wedge PRE(\beta)) \to \Box_A \chi_\gamma$ From these assumptions, infer $\vdash \chi_\alpha \to [\alpha]\Box^*_{\mathcal{C}}\psi$.

Figure 6.5: The logical system for $\mathfrak{L}(\Sigma)$

(partially) produce with BMS updates. In section 6.2.2 I show where BMS updates are too strict to properly produce an epistemic game model. The last to sections are concerned with what restrictions BMS updates have in general and ways to relax them.

6.2.1 A Good Example...

To illustrate how to turn an epistemic game model into a dynamic model, I will use one of my running examples. In the game pictured in 6.6 seen as a BMS model we have six state models, five of which consist of only one state (i.e. a, d, e, f and g).

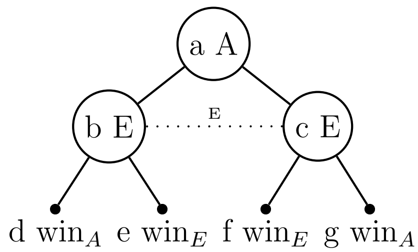

Figure 6.6: A game

The state model **S** modeling the root looks as follows: $S = \{a\}$, $\xrightarrow{A}_{\mathbf{S}} = \{(a,a)\}$, $\xrightarrow{E}_{\mathbf{S}} = \{(a,a)\}$, $\|\text{turn}_A\|_{\mathbf{S}} = \{a\}$ and $\|\text{turn}_E\|_{\mathbf{S}} = \emptyset$. The simple action structure $\mathbf{\Sigma}$ is given by $\Sigma = \{l, r\}$, $\xrightarrow{A} = \{(l,l),(r,r)\}$, $\xrightarrow{E} = \{(l,l),(r,r),(l,r),(r,l)\}$, $\text{pre}(l) = \text{turn}_A \vee \text{turn}_E$ and $\text{pre}(r) = \text{turn}_A \vee \text{turn}_E$.

Their update product is given by $S \otimes \Sigma = \{(a,l),(a,r)\}$. Let us call the state (a,l) b and the state (a,r) c. Then $\xrightarrow{A}_{\mathbf{S}\otimes\mathbf{\Sigma}} = \{(b,b),(c,c)\}$ and $\xrightarrow{E}_{\mathbf{S}\otimes\mathbf{\Sigma}} = \{(b,b),(b,c),(c,b),(c,c)\}$.

So far, we got what we wanted. But now calculating the valuation yields a problem: the valuation of turn$_A$ and turn$_E$ stays the same, in that turn$_A$ is true at each state in the model and turn$_E$ is not true throughout the model. What we want, though, is exactly the opposite: we want turn$_E$ to hold after A has made his move. Since BMS restrict their attention explicitly to actions that do not change the facts, this shortcoming is not surprising. We will turn to the question how to fix that problem in section 6.2.4.

6.2.2 ...Turning Bad

In section 6.2.1 we stopped the process of updating after calculating one new layer. But what happens if we go on? We get a new state model with states $(b,l), (b,r), (c,l)$ and (c,r) and a reflexive accessibility relation for A - but for E we get an accessibility relation that is an equivalence relation on all states. What happened is that we repeated the move that got E into the imperfect information situation in the first place.

One way to solve this is to use a different action structure for the second update product. Let us try this: we take new actions l_1 and r_1 as E's possible actions and define $\xrightarrow{A}=\xrightarrow{E}=\{(l_1,l_1),(r_1,r_1)\}$ while leaving the precondition as in the other action structure. This gives us no new uncertainty for E – but it does not give E more information, either. She can distinguish whether the last move was left or right but she still does not know about the move before that.

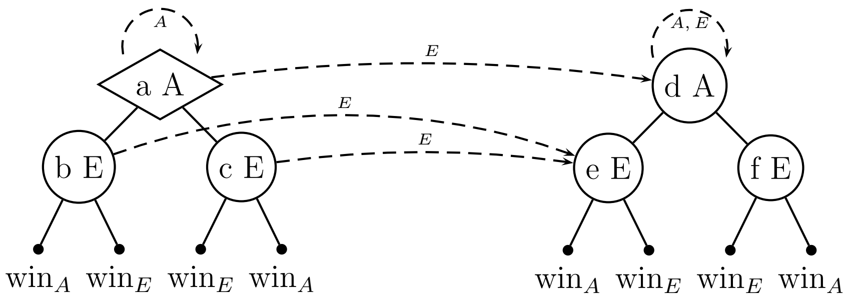

Figure 6.7: An epistemic game model

Let us look at another example. In the game pictured in figure 6.7 we have eleven state models, nine of which consist of just one state (to avoid overcrowding I did not draw the reflexive epistemic connections for A and E at those states). The two interesting state models are given as follows:

$$\mathbf{S}=(\{a,d\},\{a\rightarrow_A a, d\rightarrow_A d, a\rightarrow_E d, d\rightarrow_E d\},\|.\|_{\mathbf{S}})$$

where $\|\text{turn}_A\|=\{a,d\}$ and all other atomic sentences are evaluated by the empty set, which you can picture as in figure 6.8

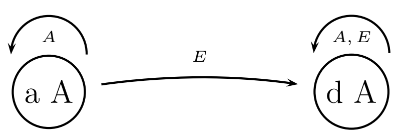

Figure 6.8: state model **S**

The second state model is formalized as:

$$\mathbf{T}=(\{b,c,e\},\{b\rightarrow_A b, c\rightarrow_A c, e\rightarrow_A e, b\rightarrow_E e, c\rightarrow_E e, e\rightarrow_E e\},\|.\|_{\mathbf{T}})$$

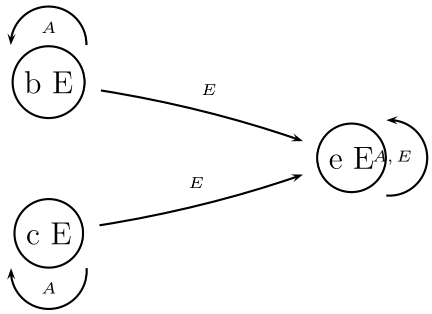

Figure 6.9: state model **T**

where turn$_E$ holds throughout the model and no other atomic sentences do. In figure 6.9 you see a picture for this.

How does the update product look in our example? In addition to **S** from above we have a set of simple actions $\Sigma = \{l, r\}$ ('going left' and 'going right'). For A, the accessibility relation is just the identity, while for E we have $l \xrightarrow{E} l$ and $r \xrightarrow{E} l$. Let the precondition for going left hold at all states in the model, while the precondition for going right shall just hold at a. So, the precondition for going left is just that 'true' holds at the state, while formulating the precondition for going right does take some atomic sentence that is supposed to hold at the left state but not at the right one. Since see from the point of epistemic game models the right state is supposed to be a copy of the left state with respect to atomic propositions, this is a problem, but you could have an atomic sentence saying 'This is the root of the real game' that can be true of at most one state in a state model. When we calculate the update[3] we get states $(a, l), (a, r)$ and (d, l). $\xrightarrow{A}$ turns out to be just the identity relation on the states, while $\xrightarrow{E} = \{((a, l), (d, l)), ((a, r), (d, l)), ((d, l), (d, l))\}$. This is depicted in figure 6.10.

Again, this update product is rather arbitrary, in that I chose a simple action structure that would yield my example. So one way to work with the BMS model would be to search for action structures modeling a given example. This is discussed in section 6.2.4. Another way would be to examine which kind of models get produced in the update product by which kind of state models and action structures. This is discussed in the next section.

Seen from the point of view of games, it would be informative not to just calculate one connected component in the update, but a whole next layer of the game. Now, how would that work here? We could change the preconditions to allow (d, r) as a state and add (r, r) to E's accessibility relation. Then we would get $(d, r) \xrightarrow{E} (d, r)$ as desired - but $(a, r) \xrightarrow{E} (d, r)$ as well. There is a simple solution to this; just draw a new

[3] We do not dicuss the valuation at this point; since an epistemic game model does 'change the facts' you cannot just take over atomic sentences from the old state model.

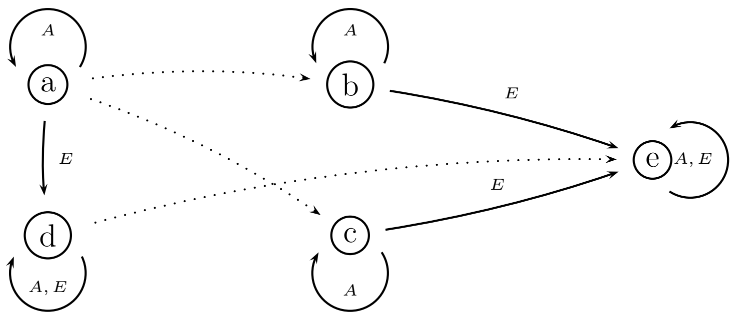

Figure 6.10: The update product

distinction between the action going right in state a and the action of going right in state d, but this is rather artificial.

6.2.3 A Characterization of Update Products

In [2] van Benthem gives a theorem characterizing product updates for accessibility relations that are equivalence relations.

Theorem 6.2.1 *The following properties determine product updates for equivalence-accessibility (and disregarding preconditions):*

1. *Perfect Recall:*

   If $(x, a) \sim_i z$ *then* z *is of the form* (u, b) *and* $x \sim_i u$ *and* $a \sim_i b$*.*

2. *Uncertainty is propagated (there are no 'miracles'):*

   if $x \sim_i y$ *and* $a \sim_i b$*, then after performance of* a *and* b*,* $(x, a) \sim_i (y, a)$*.*

3. *Uniformity:*

   Actions are either always distinguishable of never: if $(x, a) \sim (y, b)$*, then, whenever* $u \sim_i v$*, also* $(u, a) \sim_i (v, b)$*, provided the latter moves can be performed at all.*

For the proof, see [2], p. 19.

Apart from no 'changing the facts' in the BMS account, this theorem points out exactly the problems we had in modeling the example of figure 6.6: Uncertainty comes in unhandy in calculating the second layer in the example. It makes E identify states d and f for example. Uniformity causes a similar problem: states d and e are identified by E by this condition.

Note that perfect recall is no problem in the example, because the game is so short. But you get a problem with perfect recall when you try to model a game as pictured in figure 6.11. There you have an uncertainty link appearing between two nodes whose mothers E could distinguish.

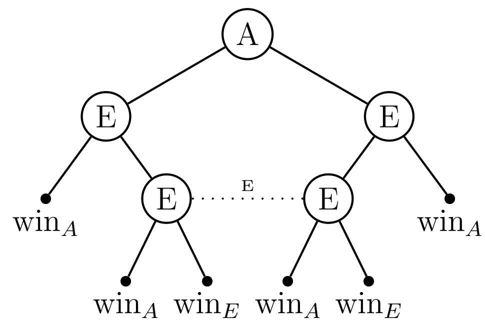

Figure 6.11: No perfect recall

Clearly also the general case is depended on the definition of the accessibility relations in the update product. For states $(x, a), (y, b) \in S \otimes \Sigma$ we have $(x, a) \xrightarrow{A}_{\mathbf{S}\otimes\mathbf{\Sigma}} (y, b)$ *iff* $x \xrightarrow{A}_{\mathbf{S}} y$ and $a \xrightarrow{A} b$.

If $x \xrightarrow{A}_{\mathbf{S}} y$ and $a \xrightarrow{A} b$ then the only thing preventing having $(x, a) \xrightarrow{A}_{\mathbf{S}\otimes\mathbf{\Sigma}} (y, b)$ in the new model are well-chosen preconditions (because then one of the tuples might not be in $S \otimes \Sigma$). For these two states we can of course chose preconditions that forbid the actions to be carried out. But remember the last example in the last section. In calculating a whole next layer of the game we got an unwanted accessibility relation.

On the other hand, we can get no 'new' accessibility relations. If $u \xrightarrow{A}_{\mathbf{S}\otimes\Sigma} v$ then there must be (x, a) and (y, b) in $S \otimes \Sigma$ such that $u = (x, a), v = (y, b)$, $x \xrightarrow{A}_{\mathbf{S}} y$ and $a \xrightarrow{A} b$. This prevents sudden loss of information; you are stuck with perfect recall.

In the next section I will discuss ways to loosen the rules of the update product without losing them completely.

6.2.4 Merging and Extending

So far, we have encountered different points where problems arise in modeling an epistemic game model with BMS actions. I will discuss them shortly in this section. The first and most obvious is that BMS's epistemic actions do not change facts, while in a game facts normally do change while playing. Another problem is that we would want to have a perfect information situation after the game is over. This might clash with the inheritance of uncertainty from above. A more aesthetic problem is that we sometimes cannot calculate a whole next layer of the game. As well, you might wonder how the 'horizontal' approach of BMS deals with epistemic relations that do not stay in the same horizontal layer of a game. It turns out that they can model such phenomena

– with cheating and suspicions, their most fancy epistemic actions. BMS models are made for arbitrary announcements whenever someone wants to announce something. In the last subsection we give an idea how to model such announcements in epistemic game models.

Changing Facts

In section 6.2.1 we encountered the problem of changing facts. This can be dealt with by an addition to the action structure. Add a function $c : AtSen \times \mathfrak{P}(\mathbf{S}) \rightarrow \mathfrak{P}(\mathbf{S} \otimes \mathbf{\Sigma})$ ('c' as a reminder of 'changing facts'). The function BMS use is c_{id} with $c_{id}(p, \|p\|_{\mathbf{S}}) = \{(s, \sigma) \subseteq S \otimes \Sigma | s \in \|p\|_{\mathbf{S}}\}$, and the function we would like to have in the example in figure 6.6 is given by $switch(p, \|p\|_{\mathbf{S}}) = S \otimes \Sigma \setminus \{(s, \sigma) | s \in \|p\|_{\mathbf{S}}\}$.

Getting Told

Normally, when a game ends, every player knows that it has ended and how it has ended. With BMS updates the 'final' action of the game need not bring about an all-knowing epistemic status of the players. In real games, this need not be the case either. Just consider a game where you gather points along while playing – after the game has ended, you first have to count to find out who has won. So we could just add an epistemic action of announcing everything that is the case, which leads to the 'real' end of the game. If we generally put the condition that each action leading to a terminal state has to be combined with announcing everything, we have ensured the 'normal ending' of a game.

Suggestive Models

The problem of calculating a whole next layer that we addressed at the end of section 6.2.2 might not lie in the BMS account, but in the overly suggestive way epistemic game models are presented. Figure 6.12 shows a bisimilar model to the one in figure 6.7 that does not suggest there is a node missing in the state model of the next layer. As long as you do not have inverse action relations in your language, you can treat each epistemic relation that goes out of a tree as a relations that points to the root of a game.

Calculating such a model by an update relation causes two problems. First, the new small tree starts out of the blue, so there is no way to calculate it BMS-style. You would have to put a 'dummy node' into the state model of the first layer to make the new tree possible. Second, the way they are defined so far, the accessibility relations cannot be shifted from the tree to the right to the tree to the left.

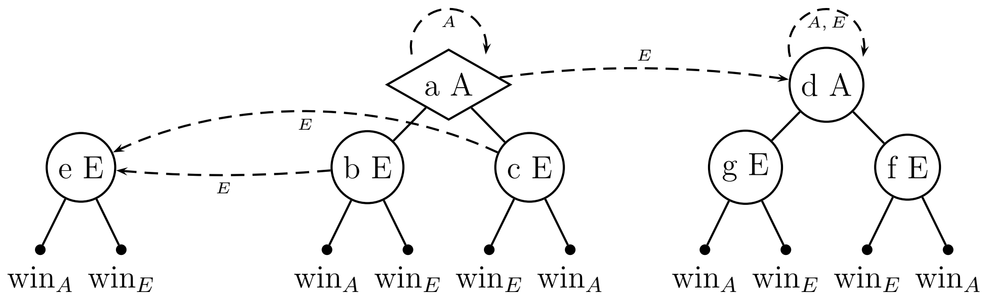

Figure 6.12: Another rendering of the game model in figure 6.7

Layers

The BMS models work horizontally, producing one state model and then updating to the next with no accessibility relations between those models. So you might expect a problem in representing epistemic relations that do not stay in their layer of the tree. This is not so. What BMS do, is copy nodes to the next layer (compare figures 6.3 and 6.4 for an example), or produce some nodes that would just show up later in the tree an update earlier. Epistemic actions that produce such a phenomenon are cheating or completely private announcements for arrows 'going up' the tree and suspicion for arrows 'going down'.

Adding Announcements

BMS's actions are mostly announcements. So if you want to render the BMS account in a static way, how can you represent the possibility of 'chatty' players who keep on saying something about the game? A way to put in announcements at any time of the game is to use non-deterministic actions. If an action a is made, this can lead to a node where it is the next player's turn to carry out an action (let that be player i) – or the game is interrupted by an announcement, in which case the action leads to that announcement which in turn should lead to a state where it is i's turn – unless another announcement takes place first.

On the left side in figure 6.13 you see a game with normal actions a and b. If A takes action a, in the second picture you have the alternative option of announcing φ before it is E's turn, and in the third there is the option of announcing ψ after φ has been announced before E goes on playing. You can of course go on iterating announcements – maybe players keep on announcing and never get back to playing. Or you can have rules that restrict the number of announcements possible before the next move is made. I left out the distinction who the announcer is (A,E, or even 'nature'), but you can easily add that information.

For the run of the game, this announcements should not make any difference (that

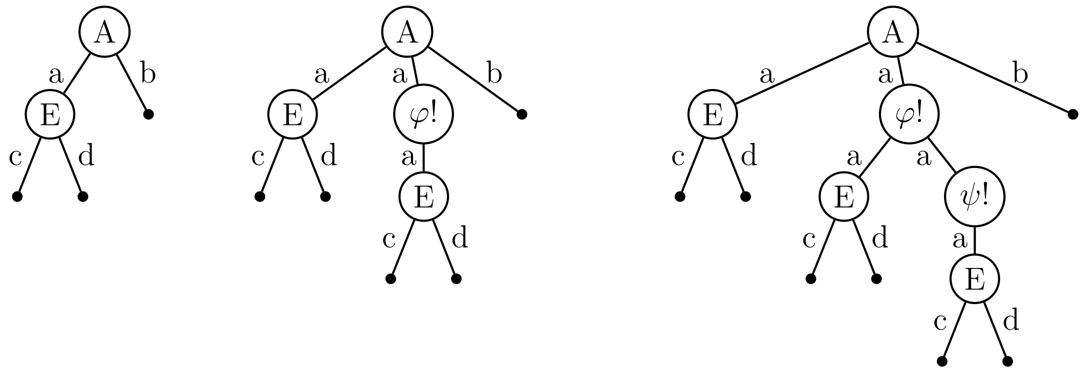

Figure 6.13: Games with and without announcements

is why we introduced the restriction that the game should not change whose turn it is next), but they can make a great difference concerning epistemic alternatives, and thus concerning the players' behavior.

BMS vs EGM

Summing up the discussions above, for perfect information games we just need to add changing facts to BMS's system. Modeling proper imperfect information (with perfect information at the leaves) needs apart from this a public announcement added to the actions that end in a leaf. Furthermore they might need a way to get around perfect recall. Games with defective information as the one in figure 6.7 show the problem of modeling a whole layer of the game as the result of an action. Also, we have the problem of getting states 'out of the blue', i.e. without inheriting them from above.

So far, we restricted the epistemic game models discussed in this chapter to those that can be described by the basic language. We did this with good reason, for BMS do not allow looking back in the game. They do allow reasoning about the future, but not as general as we might want. Here, the example from the introduction that we did not mention so far comes in. Misty games express an explosion of uncertainty – but not with the next action but somewhere in the future. In a BMS model you cannot read possible futures off the model, you have to calculate each alternative to regard it. The idea of mistiness is that it might recede the further you get (e.g. such that you can always see the two next layers of the game). But if you calculate an update, this does not change according to whether the agent really carries out the actions or just thinks about the possibilities. So whenever the agent gets to a misty point, the uncertainty has already receded further.

Chapter 7

Misty Games

In this chapter we explore the possibilities of misty games. The important feature of misty games is uncertainty about the future structure of the game. In contrast, imperfect information games focus on ignorance about the past. While the solution in imperfect information games is to play a uniform strategy, in misty games you might cannot give such a general recipe to handle the problem. What you have to do is to find strategies to reason with total uncertainty. As you will see below, the only thing that helps is other players acting irrationally. If you still want to believe in their rationality, this apparent irrationality might give you a clue about your own blind spots.

As a real life example think about a little boy whose parents keep telling him it is bad to run around barefooted. He does not see there could be connections between this behavior and getting a cold, but he can assume that his partners have their reasons and that it is really better for him to heed their advice. Here, you have an announcement by the parents and the trust of the boy that it is truthful. The game is rather non-mathematical: it is the boy's life.

But that is just the basic situation. Assume the child has grown up and his parents want to persuade him to study medicine. Now the young man can doubt their intentions: Do they just want him to make the most out of his talents in a job that gives him financial security? Or is their interest mainly in their own social status that rises by him becoming a valuable member of society? If he has some other clues, like a profound disinterest in natural sciences and other people's well-being, he should withdraw his trust in his parents' wisdom and make up his own mind. This situation has a new twist to it: both parties know that the child does not know what will happen if he studies medicine, but it might be that the parents see such a big possible value for themselves that they do not care about the possible disastrous outcomes for their son.

7.1 New Phenomena

Guessing by Actions

Assume you are in a misty game and then the other makes a move that helps you suspect what is in the 'mist'.

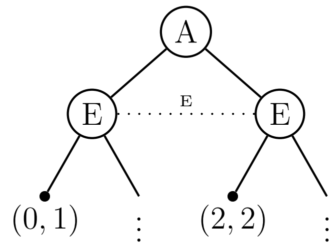

Figure 7.1: Guessing from Behavior

Regard the following example (figure 7.1). E does not know what happens if she goes right, but she believes that A believes that she has perfect information. As far as she can see, it would be rational for A to go right. Now, if A moves left she has to assume that A has more information than she has - and that he is confident that she would not go left. Her only reason for not going left would be that going right would give her a better utility, too. So it would be rational for her to choose right, in spite of not knowing what awaits her there.[1]

Of course, this kind of reasoning can easily backfire: for one, A could cheat. But it could be that A has even less information than E and makes a random move. But then, in any game he could just have plain wrong information that he acts upon. The safest strategy for E stays playing left, and hoping that she did at least get the utilities right that she could see.

This safe strategy just tunes out the parts of the game where you have too many possibilities. You just play the game like you never knew there were other ways to go. Again, this strategy is not safe against surprises: your opponent could just choose to go in such an area. And this regards mistiness as something static. Of course you can always ignore new information you get while the game runs - but what if everything looks like perfect information - until after one fatal move mist rises everywhere around you; even deleting your memory?

Learning About Ignorance

Mistiness can help you solve another dilemma. If all players are rational, and there is common knowledge about that (in fact, it suffices if the player in question believes the

[1] You can easily add a mean twist to that: What if A does know about E's epistemic situation and the utilities are (5,0)? He could just 'trick' her into going right - by her own rationality.

other players rational), and still a player does a really stupid move, the other player could stick to his belief in rationality if he assumes mistiness. In figure 7.2 this is exemplified. E is not aware that there is any option in her behavior in case A goes left. And she knows that A's utility is better when he goes right. Now, if he does go left, this will make her wondering whether there is some option hidden for her, something that A knows about but she does not. Actually, in this situation this does not help her; unless she knows what it is she could do, she cannot do it.[2]

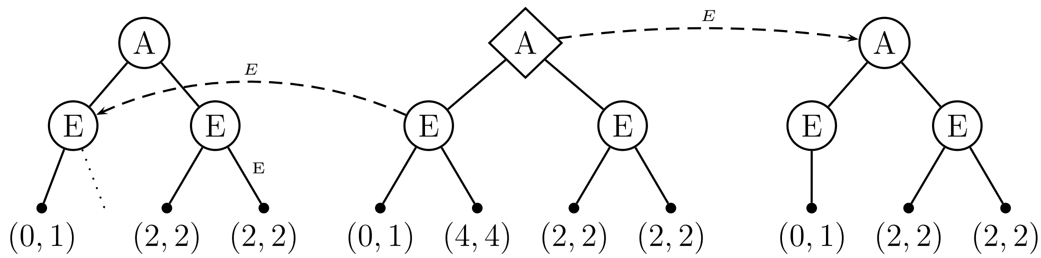

Figure 7.2: Staying Rational

Here another opportunity for putting BMS to use comes in. If you are in a misty game - what happens if you announce your ignorance? Will it lead to less security in planning a strategy (because now the other knows your weakness) or will it help getting the best outcome for all? One issue in this is whether all announcements have to be made truthfully. Another player can tell you that some action will be good for you. Shall you trust him? On the other hand, maybe lying about your own epistemic situation will give you information by evaluating the other players' reactions.

7.2 Logical Issues

One issue is how to express 'not knowing anything'. My intuition is that you would have a sentence like $\Diamond\varphi$ hold at the starting state of some misty part, where φ can be any formula that is not inconsistent. Van Benthem suggests in [2] the use of global modalities to express bounded memory. One could try to adapt this idea to bounded foresight.

Another issue is the effects mistiness can have on figuring out strategies. We do not develop a formal way to talk about strategies here, but as argued above, reasoning about strategies and other players' strategies and intentions can get very subtle. From another point of view, you can think of mistiness as a form of *bounded rationality*, i.e. the player's mental capacities are just not enough to consider the whole of the future

[2] This is why I introduced the condition $\text{turn}_i \wedge \langle a\rangle\top \rightarrow [i]\langle a\rangle\top$.

game for his strategy; he has to leave parts of it as a blanc. Seen like this, it does even make sense to reason about repeated misty games. If something good is usually hidden in the misty part, this might lead to an increase in the player's willingness to take risks, i.e. to explore unknown territory.

Chapter 8

Conclusion

In this thesis, I have introduced and investigated a new framework for modeling epistemic information in games that allows to express more general types of uncertainty than, for instance, imperfect information. In particular, we liberated the view on epistemic alternatives: not just other states of the same tree are considered, but completely independent other game trees. In doing this, we opened a wider perspective on modeling games in epistemic logic, beyond existing work on standard imperfect information games.

We gave a basic language and a minimal logic for epistemic game models and proved completeness. Furthermore we discussed several other modal formulas one might take as axioms for special subclasses of epistemic game models, such as almost bending back. By extending the language with inverse action relations and some iteration operators, we increased its expressiveness, e.g. it now became possible to define the property of being a single tree model by a modal formula. In comparing our static approach to the dynamic update systems of Baltag, Moss and Solecki [1] we found several means of extending their account to include the treatment of games.

Of course, there are still open questions. Of a more technical nature are questions about the complexity of various sorts of game models in different languages. Special conditions might play a role here. For example, Halpern and Vardi showed that dynamic epistemic logic with perfect recall becomes undecidable (see [2]). What are 'dangerous' conditions for epistemic game models?

So far, my reasoning about strategies was kept informal. It would be worthwhile to formalize it. For a language for this kind of reasoning see e.g. [2]. As well, I left the concept of rationality rather unspecified and intuitive. There is a wide field of reasoning about different kinds of rationalities and what players think other players kind of rationality is. Since my models are static, I did not discuss belief revision either. My intuition would be to model belief revision as a special kind of BMS-style update action and then look at its static counterpart. In epistemic game models, players can have more than one belief alternative. So far I left them without information about which alternative a player might think the most plausible. You could add such information by adding a weight to each epistemic arrow. This would lead naturally into epistemic probability logic.

Appendix A

Some Modal Logic

In this appendix, I state some of the main definitions and results from modal logic that I use in my thesis; see [7] for a full account.

Bisimulation and Bounded Morphisms

Definition A.0.1 (Bisimulation) Let $\mathbb{M} = (W, (R_j)_{j \in J}, V)$, $\mathbb{M}' = (W', (R'_j)_{j \in J}, V')$ be models of the same similarity type. A non-empty binary relation $Z \subseteq W \times W'$ is called a *bisimulation* between $\mathbb{M}$ and $\mathbb{M}'$ (notation: $Z : \mathbb{M} \leftrightarroweq \mathbb{M}'$) if the following conditions are fulfilled:

1. If wZw' then w and w' satisfy the same propositional variables.
2. If wZw' and $R_j w v_1 \ldots v_n$ then there are $v'_1, \ldots, v'_n \in W'$ such that $R'_j w' v'_1 \ldots v'_n$ and $v_k Z v'_k$ for all $1 \leq k \leq n$ (the *forth condition*).
3. If wZw' and $R'_j w' v'_1 \ldots v'_n$ then there are $v_1, \ldots, v_n \in W$ such that $R_j w v_1 \ldots v_n$ and $v_k Z v'_k$ for all $1 \leq k \leq n$ (the *back condition*).

$\lhd$

Definition A.0.2 (Bounded Morphism) Suppose $\mathbb{M} = (W, (R_j)_{j \in J}, V)$ and $\mathbb{M}' = (W', (R'_j)_{j \in J}, V')$ are models of the same similarity type. A function $f : W \to W'$ is a *bounded morphism* from $\mathbb{M}$ to $\mathbb{M}'$ if for all $j \in J$:

1. w and $f(w)$ satisfy the same propositional variables.
2. if $R_j w v_1 \ldots v_n$ then $R'_j f(w) f(v_1) \ldots f(v_n)$.
3. if $R'_j f(w) v'_1 \ldots v'_n$ then there exist $v_1, \ldots, v_n \in W$ such that $R_j w v_1 \ldots v_n$ and $f(v_k) = v'_k$ for all $1 \leq k \leq n$ (the *back condition*).

If f is surjective, f is called a *surjective bounded morphism*. $\lhd$

It is an immediate consequence of these definitions that every bounded morphism (seen as a relation) is a bisimulation.

Completeness and Canonical Model

Definition A.0.3 (Normal Modal Logic) A normal modal logic is a set of modal formulas that contains all tautologies, $\Box(p \to q) \to (\Box p \to \Box q)$ and that is closed under modus ponens, uniform substitution, and necessitation (or generalization). We call the smallest normal modal logic **K** $\lhd$

Definition A.0.4 (Soundness) Let M be a class of models (or frames). A normal modal logic Λ is sound with respect to M if $\Lambda \subseteq \Lambda_{\mathsf{M}}$. $\lhd$

Definition A.0.5 (Strong Completeness) Let M be a class of models (or frames). A logic Λ is *strongly complete* with respect to M if for any set of formulas $\Gamma \cup \{\varphi\}$, if $\Gamma \models_{\mathsf{M}} \varphi$ then $\Gamma \vdash_{\Lambda} \varphi$. That is, if Γ semantically entails φ on M then φ is Λ-deducible from Γ. The logic Λ is *weakly complete* with respect to M if this condition is satisfied by $\Gamma = \emptyset$. $\lhd$

Proofs for the following results can be found in [7, Section 4.2]. For simplicity, I consider only the basic similarity type. Let Λ be a normal modal logic of the basic similarity type.

Definition A.0.6 (Canonical Model) The canonical model $\mathbb{M}^{\Lambda}$ of Λ is the triple $(W^{\Lambda}, R^{\Lambda}, V^{\Lambda})$ such that

- W^{Λ} is the set of all maximal Λ-consistent sets of formulas,
- R^{Λ}, the *canonical relation*, satisfies: $R^{\Lambda}(w, w')$ iff for all formulas φ, if $\Box\varphi \in w$ then $\varphi \in w'$
- V^{Λ}, the *canonical valuation*, is defined by $V^{\Lambda}(p) = \{w \in W^{\Lambda} | p \in w\}$.

$\lhd$

Lemma A.0.7 *Given $w, w' \in W^{\Lambda}$, then $wR^{\Lambda}w'$ if and only if $\varphi \in w'$ implies $\Diamond\varphi \in w$ for all formulas φ.*

Lemma A.0.8 (Existence Lemma) *For all $w \in W^{\Lambda}$, if $\Diamond\varphi \in w$ then there is a $w' \in W^{\Lambda}$ such that $\varphi \in w'$ and $wR^{\Lambda}w'$.*

Lemma A.0.9 (Lindenbaum Lemma) *If Σ is a Λ-consistent set of formulas then there is a maximal Λ-consistent set Σ' such that $\Sigma \subseteq \Sigma'$.*

Lemma A.0.10 (Truth Lemma) *For any formula φ, $\mathcal{M}^{\Lambda}, w \models \varphi$ iff $\varphi \in w$.*

Theorem A.0.11 (Canonical Model Theorem) *Any normal modal logic is strongly complete with respect to its canonical model.*

Sahlquist Theorems

Definition A.0.12 (Positive) An occurrence of a propositional variable p is a *positive* occurrence if it is in the scope of an even number of negation signs; it is a *negative* occurrence if it is in the scope of an odd number of negation signs.

This definition refers to primitive connectives; for example $\Diamond(p \to q)$ has p occurring *negatively*, for this formula is shorthand for $\Diamond(\neg p \vee q)$.

A modal formula φ is *positive in* p (*negative in* p) if all occurrences of p in φ are positive (negative). A formula is called *positive* (*negative*) if it is positive (negative) in all propositional variables occurring in it. $\lhd$

Definition A.0.13 (Sahlquist Formula)[1] Let τ be a modal similarity type. A *Sahlquist antecedent* over τ is a formula build up from $\top, \bot$, boxed atoms, and negative formulas, using $\vee, \wedge$ and existential modal operators. A Sahlquist implication is an implication $\varphi \to \psi$ in which ψ is positive and φ is a Sahlquist antecedent.

A *Sahlquist formula* is a formula that is build up from Sahlquist implications by freely applying boxes and conjunctions, and by applying disjunctions only between formulas that do not share any proposition letters. $\lhd$

Theorem A.0.14 (Sahlquist Correspondence Theorem) *Let τ be a modal similarity type, and let χ be a Sahlquist formula over τ. Then χ locally corresponds to a first-order formula $c_\chi(x)$ on frames. Moreover, c_χ is effectively computable from χ.*

For the proof see [7], p. 165.

Theorem A.0.15 (Sahlquist Completeness Theorem) *Every Sahlquist formula is canonical for the first-order property it defines. Hence, given a set of Sahlquist axioms Γ, the logic* $\mathbf{K\Gamma}$ *is strongly complete with respect to the class of frames defined by Γ.*

For the proof see [7], p. 322 ff.

Finite Model Property and Decidability

Definition A.0.16 (Finite Model Property) A normal modal logic Λ has the *finite model property* with respect to some class of models M if $\mathcal{M} \models \Lambda$ and every formula *not* in Λ is refuted in a finite model $\mathcal{M}$ in M. Λ has the finite model property if it has the finite model property with respect to some class of models. $\lhd$

[1] Compare with [7], p. 164.

Theorem A.0.17 *If Λ is a finitely axiomatizable normal modal logic with the finite model property, then Λ is decidable.*

For the proof see [7], p. 344.

Appendix B

Some Game Theory

In this appendix I state some of the standard game theoretical definitions I use in my thesis. For a detailed introduction to game theory see e.g. [13].

Basic Notions

There are two basic forms of games (which can occur mixed in an actual game). One option is that players play one after another, so they can decide on their next move on the grounds of what has been done so far. These games are called sequential. Another option is that players have to decide on their move without knowing anything about the run of the game so far. This could be because all players put their move in an envelope, hand it to a referee, and then this referee plays the game. Games like this are called simultaneous.

Definition B.0.18 (Sequential Game) A sequential game is a game with moves following a certain predefined order. At least some players decide on their next move with information about what has happened before in the game. Sequential games are represented in extensive form (by game trees). $\triangleleft$

Definition B.0.19 (Simultaneous Game) A simultaneous game is a game in which players decide on their moves without information about what moves other players chose. Simultaneous games are usually represented in normal form. $\triangleleft$

Simultaneous games are mostly represented by a matrix, while sequential games are represented by a labeled tree.

Definition B.0.20 (Extensive Form) A game in extensive form is graphically represented by a tree. The nodes of the trees represent states of the game where a player has to take an action, while the edges represent that actions that can be taken at that node. $\triangleleft$

Definition B.0.21 (Normal Form) A game in normal form is given by a matrix. An entry $(p_1, \ldots, p_n)$ at coordinate $(a_1, \ldots, a_k)$ represents the payoffs for all players in case player i plays action a_i, $1 \leq i \leq k$. $\lhd$

In this thesis, I talk about sequential games, presented in extensive form.

In an extensive game, each leaf gets assigned an utility for each player.

Definition B.0.22 (Utility) A utility represents the motivation of a player. A higher number implies that the outcome is more preferred. $\lhd$

Definition B.0.23 (Strategy) A strategy is setting a course of action a player will follow during the game. For an extensive game, a strategy is represented by a function from all nodes in the game tree where it is the player's turn to the set of actions available to him. $\lhd$

Games and Information

Games can be classified by the amount of information their players are allowed. There is information about the structure of the game, about the history of the current run of the game, about the current state the game is in and about the future course of the current run of the game. The letter information is never assumed to be available. The main distinction is between perfect and imperfect information games. While the definition of perfect information is uncontroversial, there are a wide range of accounts what imperfect information means. One way to see imperfect information is to say that every game that is not a perfect information game is an imperfect information game. I use a different definition in my thesis that is based on the use of 'imperfect information' in [3]. Here perfect information is seen as a special case of imperfect information, while there are still other 'imperfectnesses' not captured in imperfect information games. The reason for this is the following: each definition gives some minimal requirements on the amount of information still available to the players. These minimal requirements are of course fulfilled by all kinds of games that require even more information.

Definition B.0.24 (Perfect Information Game) A *perfect information game* is a game where each player knows what the structure of the game is like and what has happened up to the current state. That is each player knows everything about the game and the run of the game, except how the other players might act in the future. $\lhd$

For the sake of completeness, you could introduce the notion of a *decided game* that has even full information about the future.

Definition B.0.25 (Imperfect Information Game) An *imperfect information game* is a game where each player knows the structure of the game though he might be uncertain of which moves were taken up to the current state. He thus might be in a situation where he considers more than one state as candidates for the current state. The current state is always among those option. ◁

In a tree-like representation of a game, this means that all epistemic relations are equivalence relations. So obviously each perfect information game is an imperfect information game.

Both kinds of games are of *complete information*, i.e. the structure of the game is known by each player. Of of course, this concept is mirrored by *incomplete information*, where the structure of the game need not be known by every player. Again, I prefer viewing complete information games as a special case of incomplete information games.

If you look at imperfect information games from a different angle, their main feature is that there are states that a player cannot tell apart from each other, because he has exactly the same beliefs in them. If you emphasis just this point, you can define what I call defective information games:

Definition B.0.26 (Defective Information Game) A game has *defective information* if there are different states in which a player has the same beliefs. ◁

For example, the game in figure 1.5 (p. 8) is of defective information. Note that defective information games do not require knowledge about the structure of the game.

Bibliography

[1] Baltag, A., Moss, L., Solecki, S.: *The Logic of Public Announcement, Common Knowledge and Private Suspicions*, manuscript, originally presented at TARK 98, 2003

[2] van Benthem, J.: *Games in Dynamic-Epistemic Logic*, in: *Bulletin of Economic Research* 53:4, ed. by Bonanno and van der Hoek, p. 219-248

[3] van Benthem, J.: *Logic in Games: Lecture Notes*, Universiteit van Amsterdam 2001

[4] van Benthem, J.: *One is a Lonely Number: on the Logic of Communication*, 2002, to appear

[5] van Benthem, J.: *Rational Dynamics and Epistemic Logic in Games*, 2003, to appear

[6] Blackburn, P., Gaiffe, B., Marx, M.: *The Complexity of Modal Reasoning on Finite Trees*, to appear

[7] Blackburn, P., de Rijke, M., Venema, Y.: *Modal Logic*, Cambridge University Press, Cambridge 2001

[8] Blackburn, P., Meyer-Viol, W., deRijke, M.: *A Proof System for Finite Trees*, in: *Computer Science Logic*, ed. by Brüning, Springer, Heidelberg 1996

[9] Fagin, Halpern, Moses, Vardi: *Reasoning about Knowledge*, MIT Press, Cambridge 1995

[10] Hughes, G.E. and Cresswell, M.J.: *A New Introduction to Modal Logic*, Routledge, London 1996

[11] Kutschera: *Einführung in die intensionale Semantik*, de Gruyter, Berlin 1976

[12] Marx, M., Venema,Y.: *Multi-Dimensional Modal Logic*, Kluwer, Dordrecht 1997

[13] Osborne, Rubinstein: *A Course in Game Theory*, MIT Press, Cambridge 1994