Knowledge games

Hans P. van Ditmarsch

ILLC Dissertation Series DS-2000-06

INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

For further information about ILLC-publications, please contact

Institute for Logic, Language and Computation
Universiteit van Amsterdam
Plantage Muidergracht 24
1018 TV Amsterdam
phone: +31-20-525 6051
fax: +31-20-525 5206
e-mail: illc@wins.uva.nl
homepage: http://www.illc.uva.nl/

Printed by Grafimedia Groningen University

ISBN: 90–367–1296–3

RIJKSUNIVERSITEIT GRONINGEN

Knowledge games

Proefschrift

ter verkrijging van het doctoraat in de

Wiskunde en Natuurwetenschappen

aan de Rijksuniversiteit Groningen

op gezag van de

Rector Magnificus, dr. D.F.J. Bosscher,

in het openbaar te verdedigen op

maandag 20 november 2000

om 14.15 uur

door

Hans Pieter van Ditmarsch

geboren op 21 mei 1959

te Eindhoven

Promotores:	Prof.dr. G.R. Renardel de Lavalette
	Prof.dr. J.F.A.K. van Benthem
Beoordelingscommissie:	Prof.dr. J.-J.Ch. Meyer
	Prof.dr. E.C.W. Krabbe
	Prof.dr. L.S. Moss

ISBN-nummer: 90–367–1296–3

Contents

Preface

Acknowledgements

I thank my supervisors Gerard Renardel de Lavalette and Johan van Benthem, the reading committee consisting of John-Jules Meyer, Erik Krabbe and Larry Moss, my colleagues, my friends, and my family, for their help. Without that I would not have completed this thesis.

Including and apart from them, I thank the following persons for their contributions to the completion of my thesis, in alphabetical order: Alexandru Baltag, Pietro Cenciarelli, Roy Dyckhoff, Jelle Gerbrandy, Rob van Glabbeek, Rix Groenboom, Wim Hesselink, Wiebe van der Hoek, Barteld Kooi, Josje Lodder, Alessio Lomuscio, Fred Mulder, Marc Pauly, Ariel Rubinstein, Marleen Sint, Keith Stenning, Yde Venema, Rineke Verbrugge, Evert van de Vrie, Ferjan de Vries. I thank many students asking many questions during many lectures.

For their financial contributions I thank the IWI Research Institute from Groningen University, the Studeerbaarheidsfonds, the Cognitive Science and Engineering Program from Groningen University, the Dutch Graduate School in Logic, the Netherlands Organization for Scientific Research, and my parents. For the hospitality offered during extended stays I thank the Human Communication Research Centre (HCRC), Edinburgh University, and the Center for the Study of Language and Information (CSLI), Stanford University.

In chapter 1, figures 1.1 and 1.2 have been reprinted with permission by Hasbro.

Overview of contents

The interaction between logic and game theory is currently of interest to the scientific community. One issue of interest in this area are games where the information contained in a game state and the information change due to a game action may be rather complex. As a concrete example of such games we define knowledge games: card games where a number of cards is distributed over a number of players, and where moves consist of information exchange, such as showing cards to other players. We characterize knowledge game states, and we define a general language for dynamic epistemics in which we can describe game actions.

Chapter 1 is a playful introduction to this thesis. It contains an analysis of the murder detection board game Cluedo. Chapter 2 provides a more technical introduction to this thesis. In that chapter, we define knowledge games, deals of cards, game states, game actions and game action execution. In chapter 3 we describe game states. In chapter 4 we introduce a language and a semantics for dynamic epistemics. Fundamental is the notion of local interpretation of actions: interpretation for a subgroup of agents only. In chapter 5 we describe game actions by means of the logical language introduced in chapter 4. In chapter 6 we give examples and applications. In chapter 7 we compare our work to that of other researchers.

Chapter 1

Cluedo

Imagine a country mansion with a couple of partying guests. Suddenly the host is discovered, lying in the basement, and murdered. The guests decide to find out among themselves who committed the murder.

The body is discovered by the butler, under suspicious circumstances that indicate that the location is not the actual murder room. In order to solve the murder it is required to find out who the murderer is, what the murder weapon was, and in which room the murder was committed. The butler is exonerated, the six guests are therefore the suspects. The guests are: Colonel Mustard (colour yellow), Professor Plum (colour pink), the Reverend Green, Mrs. Peacock (colour blue), Ms. Scarlett (colour red, i.e. 'scarlet'), and Mrs. White. There are six possible murder weapons: candlestick, rope, leaden pipe, wrench, gun, knife. The house consists of nine different rooms: hall, kitchen, dining room, study, sitting room, patio, ballroom, library, pool room.

The game consists of a board with a picture of the house, with the nine rooms in it and 'paths' leading in a certain number of steps from one room to another. Also there are six suspect cards, six weapon cards, nine room cards. A pair of dice, six pawns for the (six) players, in colours matching the guests' names, and six weapon tokens complete the picture.

There are six players. The three types of cards are shuffled separately. One suspect card, one weapon card and one room card are blindly drawn and put apart. These 'murder cards' represent the actual murderer, the murder weapon and the murder room. All remaining cards are shuffled together. They are then dealt to the players. Every player gets three cards. Some player starts the game, which is determined by throwing dice, or by the general rule that the player with the red pawn starts. That player then makes a move. A move consists of the following:

- throwing the dice
- trying to reach a room by walking your pawn over the game board
- *if* a room is reached voicing a *suspicion* about it, i.e. about a suspect, a weapon and that particular room

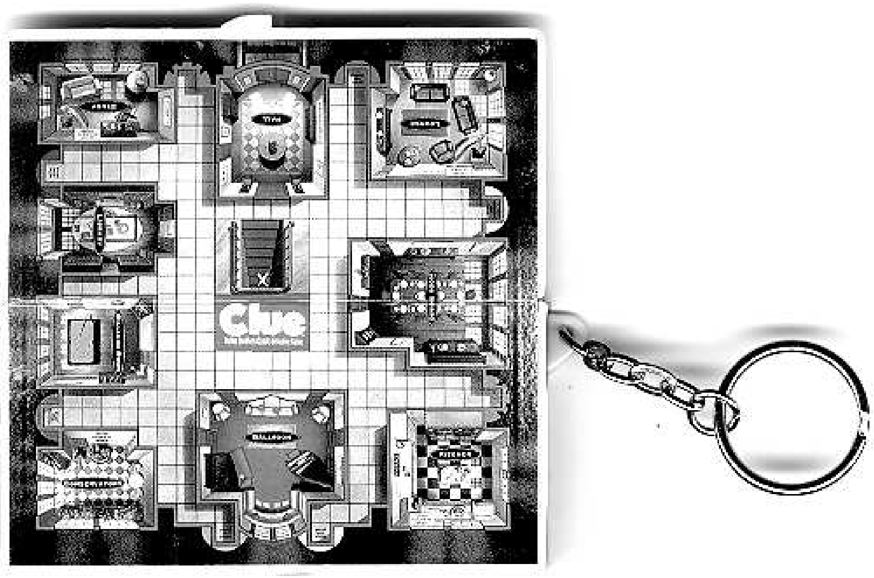


Figure 1.1: Keychain with miniature Cluedo game board

- gathering responses to that suspicion from the other players
- optional: making an *accusation* about a suspect, a weapon and a room

The number of steps on the board may not exceed the outcome of the throw of dice. As a consequence of the suspicion, the pawn with the same colour as that of the suspected player is moved to the suspected room. The other players are supposed to respond to the suspicion in clockwise fashion: the first player that holds at least one of the three cards mentioned in the suspicion, must show exactly one of those to the requesting player, and to him only. This ends the move. Whoever is next in turn is again determined clockwise. Just like a suspicion, also an *accusation* is the combination of a suspect, a weapon and a room card. Each player can make an accusation only once in the game. It is not voiced but written down. The accusing player then checks the three murder cards, without showing them to others. If the accusation is false, that player has lost and the game continues. The first player who correctly guesses the murder cards, wins the game. Note that, although pawns are identied with guests, you don't even know whether you have committed the murder 'yourself'.

The suspicion one makes is naturally supposed to elicit as much information as possible. It is based on knowledge of one's own cards and on knowledge of other players' cards. In order to justify preferring one suspicion over another, we have to determine *what* knowledge is gained from the possible answers to a suspicion.

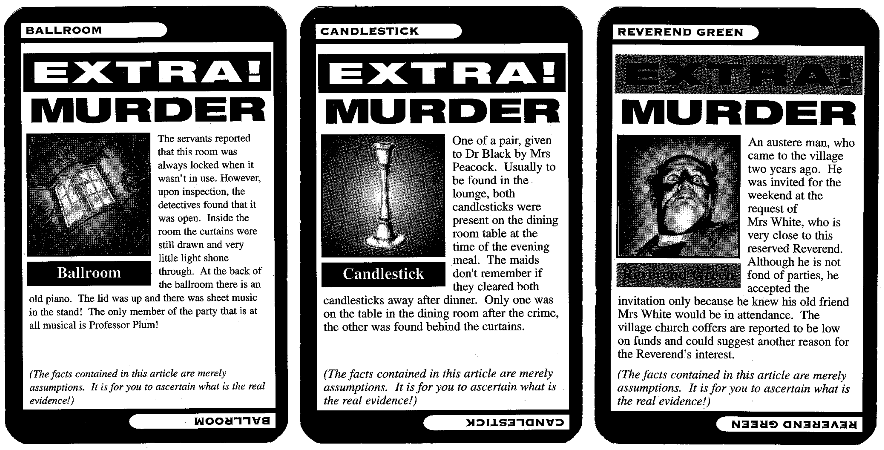


Figure 1.2: Examples of a guest, a weapon and a room card

1.1 Actions

We first discuss some examples of moves in Cluedo.

Example 1
Assume that one of the cards of player 3 is the candlestick card and that two of the cards of player 4 are the green card and the ballroom card, see figure 1.2. Assume that player 1 starts the game. In his first move, player 1 reaches the ballroom. Now the following happens, see also figure 1.3:

- **suspect** Player 1 says 'I think Reverend Green has committed the murder with a candlestick in the ballroom';
- **noshow** Player 2 says that he does not have any of the requested cards;
- **show** Player 3 shows the candlestick card to player 1;
- **nowin** Player 1 ends his move.

Players 4, 5 and 6 never get to respond to the suspicion by player 1. If 4 had been asked to, he could have chosen between two cards to show to player 1.

What is the effect of these four actions on the players' *knowledge*? Note that, as this is the first move in the game, the players only know their own cards.

suspect Any combination of a room, weapon and guest card can be asked, provided one occupies that room. Also, it is permitted to ask for one or more of one's own cards (so that one *knows* the suspicion to be false). Therefore, nothing can be deduced from the suspicion.

noshow After player 2 has said that he does not have any of the requested cards, this is commonly known to all players: player 1 knows that player 2 doesn't have

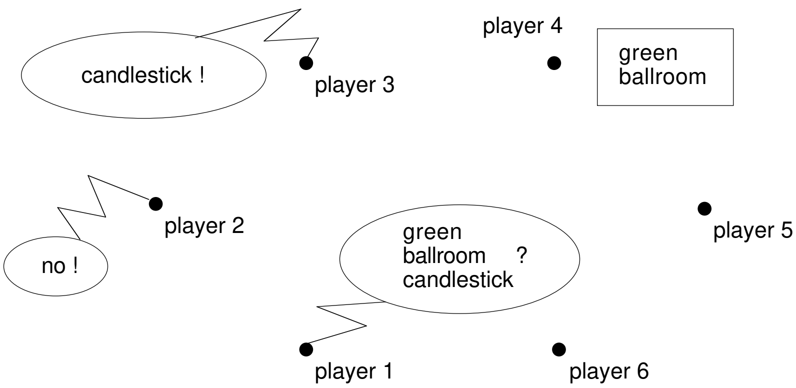


Figure 1.3: Green has done it with a candlestick in the ballroom

them, but also player 5 knows that player 1 knows that, etc. What can further be deduced from that information depends on the players' own cards. E.g. player 5 now knows that player 2 does not have 6 particular cards from the total of 21 cards: the three cards asked for by 1, and the three (different) cards that 5 holds himself.

show Player 3 has the candlestick card and shows this card to player 1. He shows the card to player 1 *only*, by handing the card face down to player 1. Player 1 then looks at the card, and returns the card the same way. The other players therefore only see that a card has been shown, and know that the others have seen that, etc.

Player 1 now knows that player 3 holds the candlestick card. Player 1 doesn't know whether player 3 holds one, two or all three of the requested cards. That he just holds candlestick, is only known by player 4, by deduction. Curiously enough, player 1 doesn't know that player 4 knows that. Nor does player 3. Common knowledge among the 6 players is only, that player 3 holds at least one of the three requested cards. From this, e.g., everybody can deduce that 3 does not hold the cards white, scarlett, and conservatory.

nowin Player 1 ends his move. This is an implicit action, only inferred because 1 does not make an *accusation*. A successful accusation corresponds to publicly announcing knowledge of the murder cards. Not accusing therefore corresponds to a public announcement that you are ignorant of the murder cards. In this example, it is unclear how that announcement changes the knowledge of other players. We therefore present a different example in which it is obvious:

Example 2
Suppose that player 1 had moved to the kitchen instead of to the ballroom, and that the murder cards are 'kitchen, scarlett, knife'. Player 1, by mere incredible luck, chooses to voice the suspicion 'I think that Scarlett has committed the

murder in the kitchen with a knife'. The player playing red moves his pawn to the kitchen. None of the other players can show a card. Player 1 writes down the accusation 'kitchen, scarlett, knife', checks it, and announces that he has won.

Example 3
Now compare the previous example with the state of the game where, instead, the cards 'kitchen, scarlett, knife' are not on the table but are held by player 1. Again, 1 voices the suspicion 'I think that Scarlett has committed the murder in the kitchen with a knife'. Obviously, again none of the other players can show a card. However: player 1 now ends his move. The other players now deduce that player 1 holds at least one of the requested cards! Observe that they do *not* know that he holds all three of them: it could have been the case that, instead, player 1 only holds 'kitchen' and that the murder cards are 'conservatory, scarlett, knife'. Once more, nobody would have shown a card to 1.

Apart from the actions apparent in example 1, three other sorts of action may occur in Cluedo: accuse, check and announce.

accuse At any moment during his move, a player may *guess* what the murder cards are, i.e. he may write down a final accusation, that will then be checked by him. *Any* combination of a room, weapon and guest card is permitted. One does not have to occupy the room of the accusation, as was required for a suspicion. Note that the content of the accusation is hidden to other players. An accusation does not effect the knowledge of the players.

check After a player has checked his accusation and has told the other players that he has indeed won, it is public knowledge (common knowledge to all) that he knows the cards on the table. However, as the game is over, it serves no purpose to describe these changes. If he told them that he lost, the game continues. Now the other players haven't learnt anything at all, because the content of the accusation was hidden to them.

announce What makes a real play of the Cluedo game even more interesting, is that players allow themselves slips of the tongue such as: 'Ha! I now know who the murderer is', or: 'Arrrgh, I still don't know the murder room!'. Such announcements often result in interesting updates.

On perfect logicians We assume that the players are perfectly logical. In actual plays of Cluedo, this assumption is dangerous. You may have all the knowledge required to deduce the cards on the table, but not make the deduction. Also, you may have forgotten earlier moves. Therefore, ending your own move does not imply that you cannot win. The next player to move, who is reasoning from the incorrect assumption that you do not know the murder cards, may now incorrectly deduce what these cards are, make that accusation, and lose.

Although it is illegal not to show a card if you hold it, it is perfectly legal not to win even though you can.

1.2 Strategy

Some actions affect the knowledge of the players, such as show. Other actions don't, such as accuse. However, actions also may effect the *beliefs* of the players. Such beliefs determine preferences among their strategies: what suspicion to make, which card to show, make an accusation now or later? E.g., one can ask for a combination of three cards from which one holds either none, one, two or three oneself. From what type of suspicion can we expect to gain the most information? We discuss this topic by way of examples.

Example 4
It is your turn. You may go to either the billiard room or to the conservatory. You know that player 2, the first to answer your suspicion, has the billiard room card or the scarlett card or the rope card. You don't know whether 2 has the conservatory card. Should you prefer a suspicion about the card that he is more likely to have, billiard room? Or about the card that he is less likely to have, conservatory? If you ask for the conservatory card, and if 2 does not hold that card, the next player after 2 will still have to answer your request, so you can gather yet more information in your move. On the other hand, now *everybody* will know that that 2 does not hold the conservatory card, which may not be to your advantage.

Example 5
You are being asked to show one of the cards scarlett, gun, and kitchen. You hold the cards scarlett, gun and conservatory. Is it better to show scarlett or gun? If you have previously shown scarlett to the requesting player, it is better to show scarlett again.

Example 6
You have just been shown a card, and have to decide whether to pass your move to the next player. You know the murder room and the murder weapon, and you know that the murderer is scarlett or green. Should you make an accusation or should you wait another round? This may never come to pass, as one of your opponents may win before. Now imagine that you also know that one of your opponents must already have gained full knowledge of the murder cards because of your move. Now, clearly, you must guess the murder cards and make an accusation.

Example 7
It seems not smart to ask for three cards that you all hold yourself, because you will not gain any information from the other players that way. However, you may

successfully mislead your opponents that way. By tricking an opponent into making a false accusation, based on the incorrect assumption that you have *not* asked for your own cards, you may get another turn and a chance to win. Otherwise, that opponent may have won instead, by preferring a different accusation (from the false one that he actually made), that turns out to be correct.

Example 8
Apart from the issue what suspicion *should* be preferred, actual Cluedo players *do* prefer some suspicions over others. It is claimed to be a good strategy to try to prevent opponents from reaching a specific room. Suppose you want to prevent player 2, who plays white, to reach the kitchen. You can try to prevent that by suspecting white to have committed the murder in the ballroom, in your turn. Also, actual Cluedo players often follow the tactic of asking for a combination of cards from which they hold one or two cards themselves. If we take (a computational model of) limited processing capability and limited short-term memory into account, this behaviour may be quite justified, although not rational. See, for a different application, [Taa99].

1.3 Simplifications

Before we model Cluedo game states and describe Cluedo game actions, we make some simplifications: We disregard the role of the board, dice and pawns. We only model actions that affect the *knowledge* of the players. We ignore that there are different types of cards. As we are only interested in the dynamics of knowledge, the second simplification is obvious. As concerns the first and third, we will argue that the difference with the real game is less than one may think.

Board, dice and pawns Board, dice, and pawns determine which suspicions players can make. The outcome of the throw of dice determines which rooms you can reach with your pawn, and therefore about which rooms you can utter a suspicion. Also, when voicing a suspicion, the pawn for that guest is moved to the room of the suspicion. Therefore the player with that pawn has to start his next move from that room. Again, that determines what room that player can reach later.

Disregarding board, dice and pawns is less of a simplification than one might think, because one can generally reach a room, and because it is totally unclear why some suspicions should be preferred over others.

For each player, the first move in the game starts from his initial pawn position. For Peacock, the closest room is reached from that position in 7 steps, for the other pawns this is 8 steps. Therefore, the player playing with the pawn Peacock will reach a room with probability $\frac{21}{36}$. For the other players the probability is $\frac{15}{36}$.

Other moves typically start from a room. From any room on the board, the number of steps needed to reach the closest room is at most 4. (From a corner room one can reach the opposite corner with any throw.) The average outcome of a throw of – two – dice is 7. Therefore, the probability to reach some other room in one's move is at least $\frac{33}{36}$. Also, one may ask a different question about the room one already occupies.

Actions that do not change knowledge Some sorts of action do not effect the knowledge of players: suspect, and accuse. Although a suspect action has no epistemic effects, it raises the issue to which noshow and show actions are the answer. We will model the combination of a suspicion, i.e. a question, with an answer to it as a *game action* in the more technical meaning of the word. Similarly, an accuse action raises an issue, although not publicly. If an accusation is *confirmed* by a check action, that combination has epistemic effects. However, as the game is over, we are not interested in those effects. If the accusation is falsified by the check action, the players are just being told that it is unsuccessful. In that case there are no epistemic effects. An announce action is not strictly a move according to the rules of Cluedo, also it is always to your disadvantage. Therefore, we will only model the actions noshow, show and nowin.

Types of cards Not just any suspicion can be made but only a suspicion consisting of a card of each type. This restricts the strategies for gathering information. Also, not just any three cards lie on the table but one of each type. This restricts the number of different deals of cards that players have to consider. In the initial state of the game there are $6 \times 6 \times 9 = 324$ possible combinations of cards on the table. Without this restriction, there would have been $\binom{21}{3} = 1330$ to consider.

The epistemic consequences of a show action are independent of the type of card that is shown. Also, the issue of strategic preference seems complicated enough when all cards are of the same type. Therefore, we have abstracted from that information too.

1.4 Historical note

Cluedo was invented by Anthony E. Pratt, a solicitor's clerk, in 1944. He said to have invented it when he was temporarily laid off because of World War II and instead doing a, mostly boring, fire brigade duty. Cluedo was first marketed by Waddington's Games, England, in 1949. In the USA, the game is called Clue instead of Cluedo. Anthony Pratt died in 1994, in obscurity. His death only became generally known in 1996, after a public appeal by Waddington's (he had already sold his rights to the game in the fifties). His tombstone reads 'inventor of Cluedo'.

Chapter 2

Knowledge games

The interaction between logic and game theory is currently of interest to the scientific community. Well-known are game theoretical foundations for logical semantics, and other applications of game theory in logic. For applications of logic in game theory, we may mention the formalization in logical theories of game theoretical notions such as game trees, plays of a game, and equilibria. One issue of interest in this area are games where the information contained in a game state and the information change due to a game action may be rather complex, and therefore become objects of study in themselves. Cluedo is a concrete example of such a game.

Given some simplifications, the game of Cluedo is nothing but a game where a finite number of cards (in this case 21) are dealt over a finite number of players (in this case 7: six 'real' players and the table), and where actions consist of either questions and answers about cards, or are announcements about (not) winning. A deal of cards is a *function* from cards to players. We will show that game states, game actions, and the transitions resulting from their execution can, amazingly, *all* be defined by operations on the function space of deals of a given finite number of cards to a given finite number of players.

We call Cluedo and similar games *knowledge games*. A knowledge game is defined by a deal of cards over players, a set of possible game actions (or moves), an order protocol to determine who is to move next, and a procedure to determine who wins. Cards do not change hands during a play of the game, although they may be shown. Players know their own cards and know how many cards all players have. The state of the game is fully determined by the deal of cards and by the game action sequence, initially empty. Although cards do not change hands, *knowledge* about cards does change during the game, and only that: therefore we have named these games *knowledge* games.

In section 2.1 we give an example of a very simple knowledge game, played by three players each with one card. In section 2.2 we discuss deals of cards. In section 2.3 we define the state of a game, as a pointed multiagent $S5$ model. In chapter 3 we will continue the treatment of this topic on a logical level: initial knowledge game states can be described in a standard multiagent epistemic logical language. In section 2.4 we define game actions. In section 2.5 we define the execution of a game action in a knowledge game state. In chapter 4 we will continue the treatment of this topic on a logical level: game actions can be described as knowledge actions in a multiagent epistemic logical language that also contains dynamic modal operators for these knowledge actions.

2.1 Three players and three cards

Even when reduced to a knowledge game, the game of Cluedo is rather complex. We start by giving an example of a simpler knowledge game. The game for three players each holding a card, is the simplest kind of knowledge game that still contains most of the features that we consider interesting.

Example 9 (The hexa game)
Consider the following game. There are three players. They are called 1, 2 and 3. There are three cards. The cards are called red, white and blue, or r, w, b (the colours of the Dutch flag). Every player is holding one card. Players can only see their own cards. A player can ask a question to another player. The question should always be answered. Also, after the question has been answered, the requesting player may announce that he knows what the deal of cards is. The first player to do so, wins the game. Players never lie, are perfect reasoners, know the kind of game they are playing, etc. We call this game the hexa game.

Only some kinds of question are permitted. A player can ask another player for one particular card, or for one of two cards, or for for one of three (one of all) cards. A question for one of three cards is a question for 'his card'. It is a rule of the game, for how many cards one can ask. In other words: you cannot choose between asking for one card or for two cards, depending on how informative you expect an answer to be.

Suppose the actual deal of cards is: 1 holds red, 2 holds white, and 3 holds blue. We call that deal: rwb. Suppose player 2 asks player 1 "do you have the red or the blue card?". Given that the actual deal of cards is rwb, we can imagine player 1 to respond by saying "yes" (i), by privately showing player 2 the red card (ii), by only showing player 2 the red card (iii), or by publicly (face up) showing player 2 the red card (iv). In (ii), by 'privately' we mean that player 3 is not aware of (and does not suspect) the question being answered. In the resulting state player 3 would incorrectly still 'know' that 2 does not know that 1 holds red. Therefore, answer (ii) does not make sense in a *game*. In (iii), by 'only' we mean that player 3 is seeing that a card is being shown, and that 1 and 2 know that he is seeing it, etc., but that 3 cannot see that it is the red card. In knowledge games we permit only answer (iii). Answer (iv) is also equivalent to *saying* "yes, namely the red card".

Apart from showing a card, there is one other type of answer to a card request. If, given deal rwb, player 2 asks player 1: "do you have the blue card", player 1 says: "no". Thus we allow two types of answer to a request: showing a requested card to the requesting player, and to him only; or saying that you do not have any of the requested cards. The combination of a request with an answer is a *game action*.

We now play an entire game. The deal of cards is rwb. The questions must be for one of three cards (a card). Player 2 starts. Player 2 asks player 1 for his card. Player 1 shows player 2 the red card. Player 2 says that he knows the deal of cards. Player 2 has won. Player 2 cannot lose this game.

We play again. The questions must be for one of two cards. Player 2 starts. Player 2 asks player 1 for the red or the white card. Player 1 shows player 2 the red card. Player 2 says that he knows the deal of cards. Player 2 has won. Player 2 cannot lose this game.

We play again. The questions must be for one card. Player 2 starts. Player 2 asks player 1 for the red card. Player 1 shows player 2 the red card. Player 2 says that he knows the deal of cards. Player 2 has won. Player 2 could have lost this game by playing differently: Player 2 starts. Player 2 asks player 1 for the white card.[1] Player 1 answers 'no, I don't have it'. Player 2 ends his move. It is now the turn of player 3. Player 3 says that he knows the deal of cards. Player 3 has won.

2.2 Deal of cards

We continue by introducing relevant concepts for knowledge games.

Definition 1 (Deal of cards)
A *deal* is a function $d : \mathbf{C} \rightarrow \mathbf{A}$ from a finite set $\mathbf{C}$ of cards to a finite set $\mathbf{A}$ of players or agents.

We can think of each player a holding the cards $c \in d^{-1}(a)$. Observe that some players may hold zero cards. We generally name the players $\mathbf{A} = \{1, 2, ..., n\}$, and the cards $\mathbf{C}$ with lower case letters. We sometimes distinguish a nonactive player, the cards on the table so to speak. In that case we assume $0 \in \mathbf{A}$ and take player 0 to be the 'table'.

We assume that deals are *total* functions, i.e. that all cards have been dealt. This is without loss of generality: suppose a deal d were partial, so that some cards are not dealt to any player, and, so to speak, remain in the stack of cards. We assume these 'remaining cards' to be 'on the table', i.e. they are held by the imaginary player 0.

[1] Although players are perfectly logical, we do not require them to be perfectly rational: they know all the deductive consequences of answers to their questions, but they cannot justify preferences among questions.

Notation Let d be the deal of three cards over three players such that 1 holds red, 2 holds white, and 3 holds blue; thus $d(r) = 1$, $d(w) = 2$, $d(b) = 3$. We introduce a shorthand notations for deals. We leave the players implicit and list only the cards they hold, assuming the numerical order of players, (possibly) separated by vertical bars. In this case we get $r|w|b$, or simply rwb, as above. If a player doesn't hold any cards, write ε. If player 3 holds a fourth, yellow (y) card as well, we get $r|w|by$ (or $rwby$). If, instead, player 2 holds no card at all, we get $r|\varepsilon|b$ (or $r\varepsilon b$).[2]

We assume that players see their own cards, and see how many cards every other player holds. This induces an equivalence relation on $\mathbf{A}^{\mathbf{C}}$.

Definition 2 (Accessibility between deals of cards)
Let $a \in \mathbf{A}$, let $d, e \in \mathbf{A}^{\mathbf{C}}$. Then:

$$d \sim_a e \Leftrightarrow d^{-1}(a) = e^{-1}(a) \text{ and } \forall a \in \mathbf{A} : |e^{-1}(a)| = |d^{-1}(a)|$$

Let $a \in \mathbf{A}, B \subseteq \mathbf{A}$. We write $\sim_B := (\bigcup_{a \in B} \sim_a)^*$ and $\sim_{\cup B} := \bigcup_{a \in B} \sim_a$ (see also appendix A).

Definition 3 (Size of a deal of cards)
Let $d \in \mathbf{A}^{\mathbf{C}}$. Write $|\mathbf{A}| = n$. The *size of deal* d, notation $\sharp d$, lists for each player the number of cards he holds:

$$\sharp d := d^{-1}(1)|...|d^{-1}(n)$$

Deals where all players hold the same number of cards are said to be of the same size. Unless confusion results, we delete the vertical bars and write $d^{-1}(1)...d^{-1}(n)$. Thus $\sharp(r|w|b) = 1|1|1$ (or 111), $\sharp(r|w|by) = 1|1|2$ (or 112), etc.

Definition 4 (Set of deals of the same size)
Given a deal $d \in \mathbf{A}^{\mathbf{C}}$, $D_{\sharp d}$ is the set of deals of the same size as d (the set of deals where all players hold the same number of cards as in d):

$$D_{\sharp d} := \{e \in \mathbf{A}^{\mathbf{C}} \mid \sharp d = \sharp e\}$$

Given a deal d, another deal e is *relevant* for the players in (that state of) the game, if all players have the same number of cards in e as in d, and if e is $\sim_{\mathbf{A}}$-accessible from d, i.e. if e is it not publicly known to be 'irrelevant', in the common meaning of that word.

[2] A alternative shorthand notation is the one where we leave the cards implicit and list only the players that hold them, assuming a fixed order of cards. I.e., the sequence $i_1...i_{|\mathbf{C}|}$ stands for the deal where the m-th card is held by player i_m. Instead of rwb we get 123. This notation, although more economic than its alternative, appears to confuse logicians.

Definition 5 (Set of relevant deals)
Let $d \in \mathbf{A}^{\mathbf{C}}$. Then

$$D_d := [d]_{\sim_{\mathbf{A}}}$$

is the set of *relevant* deals given deal d.

Instead of $d' \in D_d$ we say that d' *is relevant*. This means that a player has to take d' in consideration when reasoning about an initial state of the game for deal d (see section 2.3, next). If there are more than two players, all deals in $D_{\sharp d}$ are relevant at the beginning of a knowledge game for an actual deal d. In the proof of this proposition, we use the following notion of *transposition* of a deal: let $d \in \mathbf{A}^{\mathbf{C}}$, then $d[c, c']$ is the deal such that $d[c, c'](c) = d(c')$, $d[c, c'](c') = d(c)$, and for all other cards $c'' \in \mathbf{C}$, $d[c, c'](c'') = d(c'')$. Note that $\sharp d[c, c'] = \sharp d$.

Proposition 1
If $|\mathbf{A}| > 2$, then for all $d \in \mathbf{A}^{\mathbf{C}}$, $D_d = D_{\sharp d}$.

Proof We may assume $\mathbf{C} \neq \emptyset$ (otherwise, both D_d and $D_{\sharp d}$ are undefined). If there is one card only, all players know the deal of cards, because they can see who holds that card, and $D_d = D_{\sharp d} = \{d\}$. Now suppose there is more than one card. Let $e \in D_{\sharp d}$. Let $a_1 \in \mathbf{A}$. Let n be the number of cards that a_1 holds in d but doesn't hold in e, in other words: the number of differences between d and e. We now prove by induction on n that $d \sim_{\mathbf{A}} e$.

If $n = 0$ then $d^{-1}(a_1) = e^{-1}(a_1)$ (a_1 holds the same cards in d and e), thus $d \sim_{a_1} e$ and thus $d \sim_{\mathbf{A}} e$.

Suppose $d^{-1}(a_1)$ and $e^{-1}(a_1)$ differ in $n+1$ cards for player a_1. As there is more than one card, there are $c, c' \in \mathbf{C}$ such that $d(c) = a_1$, $e(c) \neq a_1$, $e(c') = a_1$, and $d(c') = a_2 \neq a_1$. As there are more than two players, there is a player $a_3 \neq a_1, a_2$. It holds that $d \sim_{a_3} d[c, c']$. As $d[c, c']$ differs in n cards from e for player a_1, by the induction hypothesis we can assume $d[c, c'] \sim_{\mathbf{A}} e$. From $d \sim_{a_3} d[c, c']$ and $d[c, c'] \sim_{\mathbf{A}} e$ follows $d \sim_{\mathbf{A}} e$. ■

A different way to express proposition 1 is to say that the equivalence relation $\sim_{\mathbf{A}}$ is the universal relation on $D_{\sharp d}$: $\sim_{\mathbf{A}} = D_{\sharp d} \times D_{\sharp d}$. One can even prove[3] that the maximum length of a path to link two arbitrary deals is at most three: $(\sim_{\cup \mathbf{A}})^3 = D_{\sharp d} \times D_{\sharp d}$. As $\langle D_{\sharp d}, (\sim_a)_{a \in \mathbf{A}} \rangle$ is nothing but a multiagent $S5$ *frame*, this property may help in characterizing it. We have not pursued that topic further. For the characterization of a different type of multiagent frame, see [Lom99, LvdMR00, LR98a].

[3]Proof suggested by Gerard Renardel and by Josje Lodder. Let $d \in \mathbf{A}^{\mathbf{C}}$. Let $d_1 \neq d_2 \in D_{\sharp d}$. Let $a_1 \neq a_2 \neq a_3 \in \mathbf{A}$. Let $k = |d^{-1}(a_3)|$. If $k = 0$, then $d_1 \sim_{a_3} d_2$. Otherwise, choose k cards $c^1, ..., c^k$ from $\mathbf{C} \setminus (d_1^{-1}(a_1) \cup d_2^{-1}(a_2))$. Let d_3 be of size $\sharp d$ such that $d_3(c) = a_1 \Leftrightarrow d_1(c) = a_1$ and such that $d_3(c^1) = ... = d_3(c^k) = a_3$. Let d_4 be of size $\sharp d$ such that $d_4(c) = a_2 \Leftrightarrow d_2(c) = a_2$ and such that $d_4(c^1) = ... = d_4(c^k) = a_3$. Then $d_1 \sim_{a_1} d_3 \sim_{a_3} d_4 \sim_{a_2} d_2$. ■

If there is only one player, he must necessarily hold all cards. If there are two players, each of them knows that the other player holds all other cards. Therefore, both players have full knowledge of the deal of cards. Only the actual deal of cards is relevant to them:

Fact 1
If there are only one or two players, $\sim_{\mathbf{A}}$ is the identity on $D_{\sharp d}$.

2.3 State of the game

A model for the state of a knowledge game should contain all the information that the players have about the cards and about each other. Any game state is represented by a pointed multiagent $S5$ model. In the initial state of the game, players only know their own cards. In other game states, they may know more than that. We give some examples.

2.3.1 Initial state of a knowledge game

To represent the initial state of the game for the *actual deal* of cards $\mathsf{d} \in \mathbf{A}^{\mathbf{C}}$, we propose a pointed $S5$ model. Its worlds are deals, its domain is the set $D_{\sharp\mathsf{d}}$ of deals of the size of d, its point is the actual deal. We write it in sans serif font, in order to distinguish the actual deal from other relevant deals. For each agent $a \in \mathbf{A}$, the accessibility relation between worlds is the equivalence relation $\sim_a$ as defined in definition 2. Two worlds/deals are indistinguishable from each other for a, if they agree on his cards, and if in both deals all players hold the same number of cards. Before we can define a valuation on the worlds, we have to introduce atomic propositions: $\mathbf{P}$ is the set of $|\mathbf{C}| \cdot |\mathbf{A}|$ atomic propositions c_a corresponding to player $a \in \mathbf{A}$ holding card $c \in \mathbf{C}$. For any deal $e \in \mathbf{A}^{\mathbf{C}}$, V_e is the valuation such that $V_e(c_a) = 1 \Leftrightarrow e(c) = a$. We now define a global valuation $V : D_{\mathsf{d}} \to \mathbf{P} \to \{0, 1\}$ that maps a deal e to such a (local) valuation V_e (with its argument, the deal e, as subscript). We sum it up in the following definition:

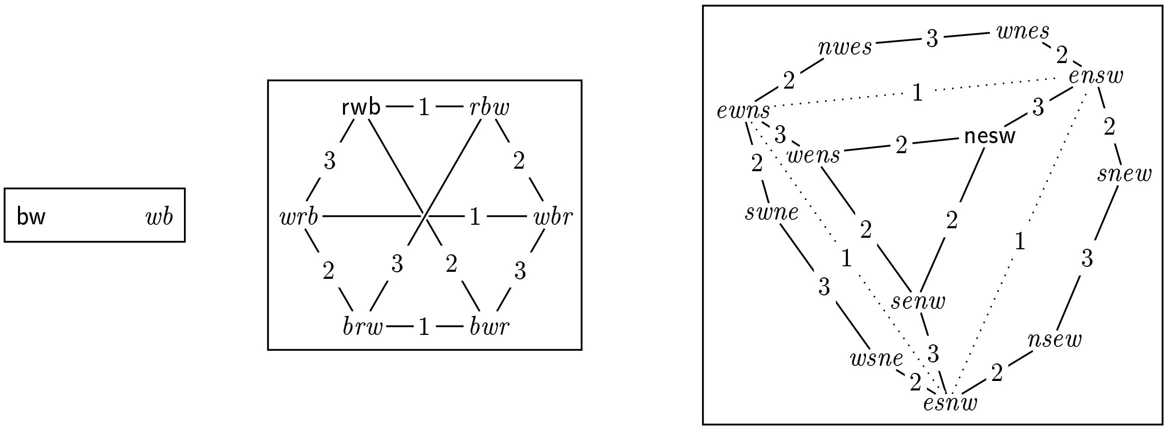
Definition 6 (Initial state of a knowledge game)
Let $\mathsf{d} \in \mathbf{A}^{\mathbf{C}}$, then the initial state of a game for actual deal of cards d is:

$$(\langle D_{\sharp\mathsf{d}}, (\sim_a)_{a\in\mathbf{A}}, V\rangle, \mathsf{d})$$

where:

$$\begin{array}{rrcl} \forall a \in \mathbf{A} : \forall d_1, d_2 \in D_{\sharp\mathsf{d}} : & d_1 \sim_a d_2 & \Leftrightarrow & d_1^{-1}(a) = d_2^{-1}(a) \\ \forall e \in D_{\sharp\mathsf{d}} : \forall c_a \in \mathbf{P} : & V_e(c_a) = 1 & \Leftrightarrow & e(c) = a \end{array}$$

Notation The model underlying an initial knowledge game state for actual deal d is written I_{d} (I for *i*nitial model), and thus the state itself is written $(I_{\mathsf{d}}, \mathsf{d})$. Instead of $(I_{\mathsf{d}}, \mathsf{d})$ we also write si_{d} (si for *s*tate and *i*nitial).



(I_{bw}, bw) (hexa, rwb) (tetra, nesw)

Figure 2.1: Examples of initial knowledge game states

Figure 2.1 presents three initial knowledge game states. See also chapter 6. In the figures, we assume reflexive access for all worlds for all agents, as the relations $\sim_a$ are equivalence relations.

Example 10 (Letter)
The left picture in figure 2.1 represents the initial state $(I_{\mathsf{bw}}, \mathsf{bw}) = si_{\mathsf{bw}}$ for the knowledge game for two players $1, 2$ and two cards b, w (black and white) with actual deal of cards $b|w$ (or bw). The point bw is in sans serif roman font. Note that the other deal $w|b$ is not relevant given actual deal $b|w$.

Example 11 (Hexa)
The middle picture in figure 2.1 represents the initial state si_{rwb} for the knowledge game for three players $1, 2, 3$ and three cards r, w, b with actual deal of cards $r|w|b$ (or rwb). The point rwb is in sans serif roman font. We call the underlying model hexa. There are six different initial states of that game, corresponding to choosing a different deal as point in hexa. As the figure has the shape of a hexagon, it will now be clear why we call the corresponding game the hexa game.

Example 12 (Tetra)
The right picture in figure 2.1 represents the initial state si_{nesw} for the knowledge game for three players $1, 2, 3$ and four cards n, e, s, w (north, east, south and west) with actual deal of cards $n|e|sw$ (or $nesw$). The point nesw is in sans serif roman font. Access for agent 1 is only given in some typical cases. We call the underlying model tetra, because the figure has the shape of a semi-regular polyhedron called a truncated tetrahedron.

2.3.2 State of a knowledge game

After the cards have been dealt and everybody has seen his cards, players can learn about the cards of other players by means of game actions. Also other knowledge game states can be represented by pointed multiagent $S5$ models. We only require that agents *at least* know their own cards. In the next section, 2.4, we then define a game action; the execution of a game action induces a binary relation between such states.

Definition 7 (Knowledge game state)
A knowledge game state for deal of cards d is a pointed $S5$ model

$$(\langle W, (\sim_a)_{a\in\mathbf{A}}, V\rangle, v)$$

where $v \in W$, and $V_{\mathsf{d}} = V_v$, and:

$$\forall w \in W : \exists d' \in D_{\sharp\mathsf{d}} : V_w = V_{d'}$$

and for all $a \in \mathbf{A} : \forall w_1, w_2 \in W : \forall d_1, d_2 \in D_{\sharp\mathsf{d}}$:

$$(w_1 \sim_a w_2, V_{w_1} = V_{d_1}, V_{w_2} = V_{d_2}) \Rightarrow d_1^{-1}(a) = d_2^{-1}(a)$$

Notation Unless confusion arises, we prefer to name worlds by the deals that atomically characterize them. So, worlds that are named by the same deal only differ in their access to other worlds. We then can continue to write d for the point of a state, instead of v. We often write s_v, or s_{d}, for a knowledge state for dealing d with point v.

As the game progresses, more and more deals of cards become irrelevant, in the sense that all players are known not to consider them any longer. This is captured by the following definition.

Definition 8
Let $s = (\langle W, (\sim_a)_{a\in\mathbf{A}}, V\rangle, v)$ be a knowledge game state for $\mathsf{d} \in \mathbf{A}^{\mathbf{C}}$, then:

$$D_s = \{d' \in D_{\mathsf{d}} \mid \exists w \in W : w \sim_{\mathbf{A}} v \text{ and } V_w = V_{d'}\}$$

Even though it is now clear what kind of mathematical objects knowledge game states are, this does not clarify what agents actually *know* in such a state of the game. In chapter 3 we characterize the knowledge of the agents in initial knowledge game states, by describing these states in a multiagent epistemic logic with common knowledge operators. Because knowledge game states are finite models, their description can be computed with standard modal techniques, see [vB98, BM96]. The description of the model underlying the initial state of a knowledge game for deal $d \in \mathbf{A}^{\mathbf{C}}$ is equivalent to (the $\mathbf{S5_n}$ axioms plus) $\bigvee_{d'\in D_d} \delta_{d'}$ and $\bigwedge_{a\in\mathbf{A}} \bigwedge_{d'\in D_d}(K_a\delta^a_{d'} \leftrightarrow M_a\delta_{d'})$. Here, $\delta_{d'}$ is the atomic description of world (deal) d', and $\delta^a_{d'}$ is the part of that description about agent a (i.e. the conjunction of atoms c_a or their negations that describe the cards of a).

2.4 Game action

Given a knowledge game state, we now define game actions for that state. First we give an example of a game action and the knowledge state that we expect to result from its execution. Then, we speculate on a desirable format for game actions. Only after that, we present the definition of game actions. We conclude with applying the definition to the example action and with an overview of game actions in knowledge games.

Example 13 (1 shows red to 2)
In the initial state (hexa, rwb) of the hexa game, player 2 asks player 1 "please show me your card". The question is public: 3 hears it too. Player 1 answers this request (as in response (iii) related to example 9 on page 10) by handing his red card face down to player 2. Player 2 then looks at that card, after which he returns the card face down to player 1.

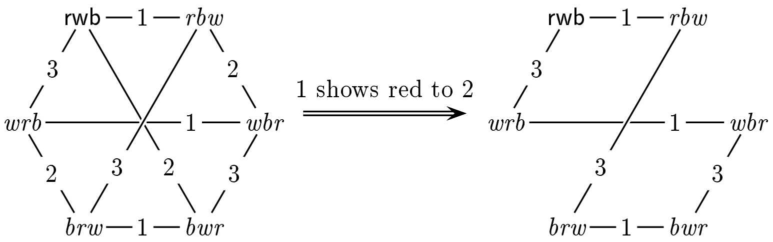


Figure 2.2: Player 1 shows (only) player 2 his red card

The resulting state of the game is pictured on the right in figure 2.2. In the resulting game state, player 2 knows that player 1 has the red card. As 2 also knows his own card, and he knows that there are three cards, he therefore also knows that player 3 holds the blue card. In other words: 2 knows the actual deal of cards. Players 1 and 3 don't, but compared to the initial game state, they still have learnt something. Now they know, e.g., that player 2 knows the actual deal of cards.

Before we can compute game states resulting from game actions, we have to define what an action 'is'. In chapter 4 we define a logical language to describe such actions. Here, we will investigate game actions from a purely semantical point of view: juggling with sets of deals, so to speak. So what is happening here? Player 2 knows that player 1 holds a card. In other words, 2 is aware of the partition of hexa induced by the equivalence relation $\sim_1$. By asking player 1 for his card, player 2 is presenting to player 1 the three different equivalence classes of $\sim_1$, for 1 to choose from: $\{rwb, rbw\}$, $\{wrb, wbr\}$, and $\{brw, bwr\}$. Player 1 hasn't much to choose, in this case, and has to answer by affirming that his information state (i.e. the set of worlds that he considers to be possible, see appendix A) is $\{rwb, rbw\}$, which corresponds to 1 holding the red card. Player 3 does not receive the answer in the detail in which player 2 gets it. E.g. 3 cannot distinguish the answer red from the answer white.

A game action is a question with an answer to it We suggest that, just as in the action of example 13 where 1 shows red to 2, all game actions in knowledge games are the combination of a question with an answer to it. Another parameter of crucial importance is what other players perceive of the answer to the question, just as in the action where 1 shows red to 2. We call that the *publicity* of the game action.

Definition 9 (Game action)
Let $s = (\langle W, (\sim_a)_{a\in\mathbf{A}}, V\rangle, \mathsf{d})$ be a knowledge game state. A *game action* μ for state s is a quintuple

$$\mu = \langle q, Q, r, R, \mathsf{pub}\rangle$$

where $q, r \in \mathbf{A}$, Q is a covering of W that is coarser than $\sim_r$, $R \in Q$, and pub is a function from agents $a \in \mathbf{A}$ to equivalence relations pub_a on Q, and pub_r is the identity '=' on Q.

For 'coarse' and 'covering', see appendix A. In definition 9, q is the requesting player, Q is the *Q*uestion, r is the respondent, R is the answer or *R*esponse, and pub is the 'publicity': how and what the respondent r makes public to other players of his answer to q. We can think of the elements $R_1, ..., R_{|Q|}$ of Q as 'possible answers' or 'alternative answers'. Obviously, r is informed about his own behaviour: the respondent can distinguish all possible answers from each other. In other words, the equivalence relation pub_r on the set of alternatives Q is the identity '='. We do not assume that the requesting player q is also fully informed, corresponding to $\mathsf{pub}_q =$ '=', although this is often a reasonable assumption.

Player 0 (the table), if there is one, cannot ask questions. Also it is only allowed to respond in certain reactive ways, and not proactively. What ways is determined by pub. You can draw a card from a stack on the table, and let the table respond to the request "show me one of your cards" in that way. You cannot ask the table "do you have the red or white card" and get "no" as an answer, or have it 'decide' which one of these two cards to show you.

Definition 10 (Executable game action)
Let $s = (\langle W, (\sim_a)_{a\in\mathbf{A}}, V\rangle, w)$ and $\mu = \langle q, Q, r, R, \mathsf{pub}\rangle$. Game action μ is *executable* in knowledge game state s if the answer R contains actual world w:

$$\mu \text{ is executable in } s \Leftrightarrow w \in R$$

In simpler words: a game action is executable if the respondent r answers the question truthfully; we may also say: if r's information state, i.e. $[w]_{\sim_r}$, is contained in the answer. In section 2.4.2, definitions 9 and 10 will be applied to example 13.

2.4.1 Publicity

Because the concept of 'publicity' is central to our approach, we give some motivations for it.

Knowledge games are all about getting information. The way to obtain information is to ask questions, with the expectation of getting certain answers, or to observe others asking and responding. We are only interested in (perfect) information: what other players know, what cards they hold; and not in strategic information: how likely it is they will ask a certain question, etc. Given this restriction, we can state that obtaining information (from questions and answers) is learning about the information state of the respondent. This 'learning' is not just individual but on the level of *subgroups* that gain *common knowledge* about the information state of the respondent.

There is a smallest nonempty subgroup $Br \subset \mathbf{A}$ – the broadcast unit so to speak – that receives the answer R to q.[4] Obviously $r \in Br$. Often, $q \in Br$. If the broadcast unit $Br = \mathbf{A}$, then the response R is publicly learnt. Otherwise, the broadcast unit Br is contained in at least one larger subgroup $B' \subseteq \mathbf{A}$. For the players in B' that are not in Br, that group Br learns answer R might be just one of several alternatives. For all they know, Br learns an alternative $R' \in Q$ that differs from response R. Or instead of Br learning R, a different subgroup $B'' \subset B'$ learns R. Even then, $r \in B''$, because the respondent r controls the publicity. Every such subgroup B' that is smaller than $\mathbf{A}$, is again contained in a larger one for which analogous restrictions hold. At some stage the entire group of agents $\mathbf{A}$ learn something: every action must have a public part.

If an action would not have a public part, some agents would learn nothing and would therefore think that nothing happened. In that case they have false knowledge of the state of the game: they 'know' that nothing happened, they 'know' that the broadcast unit Br has not learnt R, etc. As they do not consider the actual state of the game to be possible, the resulting pointed multiagent modal model is not reflexive and therefore not an $S5$ state, so certainly not a knowledge game state.

We only require that learning subgroups contain the respondent r, at whatever level of the transmission. We might additionally have required that subgroups

[4] Strictly speaking, we mix up syntax and semantics here: one doesn't learn R but one learns a proposition φ_R with interpretation R. As our models are finite we can safely assume that such a proposition exists. We postpone introducing the logic to chapters 3 and 4.

also contain q, the requesting player. We haven't done that, because we also want to model actions such as: 'the respondent r showed a card to the requesting player q and his other card to player a'.

This may seem rather complex, but the very simple way to fulfill these constraints is to define for each agent a an equivalence relation pub_a on the set of alternative answers of a game action, as in definition 9. Having done that, for each subgroup $B \in \mathbf{A}$ we can, if so desired, compute pub_B (i.e.: $(\bigcup_{a \in B} \mathsf{pub}_a)^*$). *The equivalence class of* $\sim_B$ *that contains the answer* Q *stands for what subgroup* B *learns in that game action.*

2.4.2 Examples

Example 14 (1 shows red to 2, continued)
In example 13 we described the game action of 1 showing red to 2. This corresponds to the following game action:

$$\langle 2, \{\{rwb, rbw\}, \{wrb, wbr\}, \{brw, bwr\}\}, 1, \{rwb, rbw\}, \mathsf{show}\rangle$$

Player 2 asks the question. The question is $\{\{rwb, rbw\}, \{wrb, wbr\}, \{brw, bwr\}\}$, i.e. the three equivalence classes of $\sim_1$. Player 1 answers the question. The answer is $\{rwb, rbw\}$. This corresponds to 1 showing the red card. The publicity show is defined as follows: show_1 and show_2 are the identity '$=$' on the question, and show_3 is the universal relation U on the question.

According to definition 10, the actual deal of cards rwb should be contained in the given answer $\{rwb, rbw\}$. This is indeed the case. Therefore this game action is executable in initial state $(\mathsf{hexa}, \mathsf{rwb})$.

For player 3, the action where 1 shows red is indistinguishable from the action where 1 shows white. Because 3 holds blue himself, he does not consider that 1 actually shows blue, so that the action where 1 shows blue *can* be distinghuished by player 3 from the action where 1 shows red. However, because 1 doesn't know that 3 holds blue, 1 can imagine that 3 can imagine that 1 shows blue, or in other words: at some point or other, all three action alternatives have to be taken into account: 3 is not publicly known to be able to distinguish between the alternatives.

The observations on publicity in subsection 2.4.1 are mirrored by the computations we can do on publicity show in example 14. Using the equivalences $\mathsf{show}_1 = \mathsf{show}_2 =$ '$=$' and $\mathsf{show}_3 = U$, we can compute for every subgroup of the public $\{1, 2, 3\}$ what that subgroup has learnt. E.g. show_{12} is also the identity, whereas show_{13} is, again, the universal relation: 1 and 3 do not 'share' that the action of showing blue can be eliminated, as pointed out in the previous paragraph. Also, show_{123} is the universal relation: it is not publicly known which action has been taking place.

The next example illustrates why the alternatives are only required to *cover* the domain, and not to *partition* it. If the alternatives overlap, the responding player may *choose* from the alternatives that contain his information state:

Example 15
Assume initial knowledge game state $(\mathsf{hexa}, \mathsf{rwb})$. Consider the following action: Player 2 (publicly) asks player 1 "please tell me a card that you do not hold". As a response player 1 whispers in 2's ear "I do not hold white". Player 3 cannot hear the answer, although he knows that an answer has been given. This game action is described as follows; abbreviate $\{wrb, wbr, brw, bwr\}$ as R_{notred}, $\{rwb, rbw, brw, bwr\}$ as R_{notwhite}, and $\{rwb, rbw, wrb, wbr\}$ as R_{notblue}:

$$\langle 2, \{R_{\mathsf{notred}}, R_{\mathsf{notwhite}}, R_{\mathsf{notblue}}\}, 1, R_{\mathsf{notwhite}}, \mathsf{show}\rangle$$

Again, show_1 and show_2 are the identity and show_3 is the universal relation on the question. The alternative answers indeed cover the domain. E.g., $R_{\mathsf{notwhite}} = \{rwb, rbw, brw, bwr\}$ is the union of the two classes $\{wrb, wbr\}$ and $\{brw, bwr\}$ of $\sim_1$. The alternatives also overlap, e.g. R_{notwhite} and R_{notblue}: in the given state, player 1 could also have answered that he doesn't have blue.

In a knowledge game state where a player holds more than one card, he can choose between cards to show, given a request:

Example 16
Consider the state on the right in figure 2.1 on page 15, where player 3 holds the south and the west card. If player 2 asks player 3 for a card, player 3 may choose between showing south and showing west.

In chapter 4, we introduce a multiagent dynamic epistemic language for describing game actions. For example, the *game action* of 1 showing red to 2 is described by the *knowledge action* $L_{123}(!L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1)$. Informally, we can read this expression as follows: 1 and 2 learn that 1 holds red, and 1, 2 and 3 learn that either 1 and 2 learn that 1 holds red, or that 1 and 2 learn that 1 holds white, or that 1 and 2 learn that 1 holds blue. In chapter 6, the game action examples from this section are described as knowledge actions and are treated in more detail.

2.4.3 Game actions in knowledge games

Only the following sorts of action occur in knowledge games: showing a card, not showing a card, winning, and not winning:

Definition 11 (Legal game actions in knowledge games)
Let $s = (\langle W, (\sim_a)_{a\in\mathbf{A}}, V\rangle, v)$ be a knowledge game state for a deal of cards $d \in \mathbf{A}^{\mathbf{C}}$. The sorts of action show, noshow, win and nowin are defined as follows; the

middle column presents the abbreviated notations that we often use for them:

sort	*name*		*definition*
show	$\mathsf{show}^{q,\{c^1,\dots,c^t\}}_{r,c^j}$	$=$	$\langle q, \{R_{c^1_r}, \dots, R_{c^t_r}, Comp\}, r, R_{c^j_r}, \mathsf{show}\rangle$
noshow	$\mathsf{noshow}^{q,\{c^1,\dots,c^t\}}_{r}$	$=$	$\langle q, \{R_{c^1_r}, \dots, R_{c^t_r}, Comp\}, r, Comp, \mathsf{show}\rangle$
win	win^q	$=$	$\langle q, \{Win, W \setminus Win\}, q, Win, \mathsf{id}\rangle$
nowin	nowin^q	$=$	$\langle q, \{Win, W \setminus Win\}, q, W \setminus Win, \mathsf{id}\rangle$

We use the following abbreviations in the definition: $R_{c^i_a}$ stands for the union of equivalence classes of $\sim_a$ where a holds card $c^i \in \mathbf{C}$; $Comp$ stands for $W \setminus \cup^t_{i=1} R_{c^i_r}$, the complement of the union of all alternatives that correspond to r showing card c^i; Win stands for $\cup^t_{i=1} R_i$, the union of all equivalence classes R_i of $\sim_q$ where q can win. Publicity id maps each agent $a \in \mathbf{A}$ to the identity on the question, i.e. $\mathsf{id}_a =$ '$=$'. Publicity show is defined as follows: show_q and show_r are the identity on the question, and for all other agents a, show_a is: *universal* on the question minus $Comp$, and the *identity* on that complement: $\mathsf{show}_a(Comp, Comp)$.

show and noshow In a show action player q asks player r to show him one of t cards $c^1, \dots, c^t$, and r responds by showing (only) q that card. In a noshow action player q asks player r to show him one of t cards $c^1, \dots, c^t$, and r responds by saying "no, I don't have any of those". Given some current state of the game, it can be that one or more of the R_{c^i} are empty. Also, if $t = |\mathbf{C}|$, the set $Comp$ is empty. In that case, we assume that they do not occur in the question: the question may contain only nonempty alternatives.

win and nowin The actions of winning and not winning are public announcements. In our game action format an announcement is a (public) question to oneself that is publicly answered. If winning is knowing the actual deal of cards, Win is the union of all equivalence classes of $\sim_q$ that are characterized by a single deal of cards. If there are none, $Win = \emptyset$ and we assume that the question consists of W only: the question may contain only nonempty alternatives. Of course, $W \setminus Win$ corresponds to the union of all the equivalence classes where q cannot win.

The publicity functions show and id are the only common publicity functions that we have encountered.

2.4.4 Questions can always be answered

To illustrate how general the game action format is, we give two more examples.

Example 17 (What you don't know, is trivial)
Consider the state (hexa, rwb). Suppose player 2 asks player 1 "which card do I have?". Player 1 now answers "I don't know".

This appears to be a question that cannot be answered and therefore doesn't fit our game action format. It turns out that we can rephrase it to fit the format:

A question covers the domain of the game state, and is coarser than the partition induced on it by the equivalence relation for the responding player, in this case player 1. How does the question "which card do I have?" cover the domain of hexa? The informative answers to "which card do I have" are "red", "white" and "blue". Such an answer should contain the information state of 1, or, in other words, 1 should know that 2 holds that card. We first investigate the alternative corresponding to the answer "red". In no equivalence class of $\sim_1$, or union of classes, does 1 know that 2 holds red. Therefore, to this alternative corresponds the empty set $\emptyset$. Same for "white" and "blue". To these 'alternatives' we add the complement of their union, in this case the entire domain of hexa: $\mathcal{D}(\mathsf{hexa})$. The answer "I don't know" corresponds to choosing that complement. Therefore this game action is represented by:

$$\mu = \langle 2, \{\mathcal{D}(\mathsf{hexa})\}, 1, \mathcal{D}(\mathsf{hexa}), \mathsf{id}\rangle$$

This is a trivial game action, where player 1 asserts that his information state is contained in the domain of the state of the game. This was already known to all players, so that there is nothing to be gained from executing this game action: the resulting game state is, again, (hexa, rwb). In the initial state of the game it is senseless to ask others about your own cards, and everybody knows that.

A question is commonly considered 'trivial', if the person asking knows in advance the answer to his question. We suggest that a game action is trivial in a technical sense, if that is the case for all players.

Definition 12 (Trivial action / question)
A game action $\langle q, Q, r, R, \mathsf{pub}\rangle$ for state s_w is trivial, when for all $a \in \mathbf{A}$: $[\mathsf{d}]_{\sim_a} \subseteq R$ and $\forall R' \neq R \in Q : w \notin R'$.

Without the constraint expressing that w is only in R, the responding player might have chosen a possibly informative (and therefore non-trivial) answer $R' \neq R$. Example 17 was an illustration of this definition.

We have thought of several other sorts of game action, apart from the five introduced in this and in the previous subsection, but we do not pursue this topic further.

We have modelled questions and answers in *games*. The logical modelling of questions and answers is studied more generally by Jeroen Groenendijk in [Gro99], 'The Logic of Interrogation'. In his approach, a question induces a *partition* on a *part* of the domain of a state. In our approach, questions induce a *covering* of the *entire* domain. Because questions are not partial on the domain, they can always be answered. Because possible answers can overlap, the respondent may choose between alternative answers to a question. Also because of that, the model resulting from execution of a game action, i.e. a question/answer combination, may be more complex than the model in which the question was posed. First, we have to define the construction of that resulting model.

2.5 Computing the next state of the game

We know what knowledge game states are and how to model game actions. We still have to define what game state results from executing a game action. We do not need all the parameters of a game action for that: who asked the question and who responded to it, is irrelevant for computing the information changes. Stripped from these two parameters, what remains is a multiagent $S5$ frame that we call a game action frame.

Definition 13 (Game action frame)
Let $\mu = \langle q, Q, r, R, \mathsf{pub}\rangle$ be a game action. To this game action corresponds the pointed multiagent $S5$ frame:

$$\mu^- = (\langle Q, \mathsf{pub}\rangle, R)$$

Note that the frame may consist of disconnected parts, both when the publicity of the action is id as when it is show.

We can now define how to compute the next state of a game from a given knowledge game state and a game action. A knowledge game state is represented by a pointed $S5$ model. A game action is represented by a pointed $S5$ frame. The computation of the next game state from the current state and an action, or in other words the execution of that action in that state, can be seen as *multiplying* the pointed $S5$ model for that state with the pointed $S5$ frame for that action: it resembles the computation of a direct product (see appendix A).

Definition 14 (Executing a game action in a knowledge game state)
Let $s = (\langle W, (\sim_a)_{a\in\mathbf{A}}, V\rangle, v)$ be a knowledge game state for actual deal d and let $\mu = \langle q, Q, r, R, \mathsf{pub}\rangle$ be a game action executable in s. The knowledge game state $s \otimes \mu$ resulting from executing μ in s is defined as follows:

$$s \otimes \mu := (\langle W', (\sim'_a)_{a\in\mathbf{A}}, V'\rangle, (v, R))$$

where:

$$\begin{array}{rcl} W' & = & \{(w, R') \in W \times Q \mid w \in R'\} \\ \text{and } \forall a \in \mathbf{A} : \forall w, w' \in W : \forall R', R'' \in Q : & & \\ (w, R') \sim'_a (w', R'') & \Leftrightarrow & w \sim_a w' \text{ and } \mathsf{pub}_a(R', R'') \\ V'_{(w,R')} & = & V_w \end{array}$$

The general idea of this construction is, that the next state of the game consists of all pairs (w, R') such that R' 'could also have been' the given answer and w 'could also have been' the (point of the) current state, plus access appropriately defined. The computation of $s \otimes \mu$ does not depend on the roles of the player asking the question and the player responding to it. Therefore, instead of $s \otimes \mu$ we may also write $s \otimes \mu^-$, where μ^- is the pointed frame corresponding to game action μ. We still have to prove that the resulting model $s \otimes \mu$ is a knowledge game state. This is indeed the case:

Proposition 2 ($s \otimes \mu$ is a knowledge game state)
Let $s = (\langle W, (\sim_a)_{a \in \mathbf{A}}, V \rangle, v)$ be a knowledge game state for deal d, and let $\mu = \langle q, Q, r, R, \mathsf{pub} \rangle$ be a game action executable in s. Then $s \otimes \mu$ is a knowledge game state for deal d.

Proof We check the requirements from definition 7:

- $V'_{(v,R)} = V_{\mathsf{d}}$:
  This follows from $V'_{(v,R)} = V_v$ and $V_v = V_{\mathsf{d}}$.
- every world in W' is characterized by a(n) (initially) relevant deal:
  This follows from $V'_{(w,R')} = V_w = V_{d'}$ for some $d' \in D_s \subseteq D_{\mathsf{d}}$.
- for all $a \in \mathbf{A}$, $\sim'_a$ is an equivalence relation, which is obvious, such that:
- if players cannot distinguish between two worlds, they hold the same cards in (the deals that characterize) those worlds:
  If $(w, R') \sim'_a (w', R'')$, then $w \sim_a w'$, so a holds the same cards in w and w' and therefore also in (w, R') and (w', R''). ■

In general, it does not hold that the product of two connected structures is connected. However, when executing game actions in game states, this property is preserved:

Proposition 3 (Preservation of connectedness)
Let $s = (\langle W, (\sim_a)_{a \in \mathbf{A}}, V \rangle, v)$ be a knowledge game state for deal d, let $\mu = \langle q, Q, r, R, \mathsf{pub} \rangle$ be a game action executable in s. Write $s \otimes \mu$ as above.

If $[v]_{\sim_{\mathbf{A}}} = W$ and $[R]_{\mathsf{pub}_{\mathbf{A}}} = Q$, then $[(v, Q)]_{\sim'_{\mathbf{A}}} = W'$.

Proof Let $(w, R'), (w', R'') \in W'$. From $w, w' \in W$ follows $w \sim_{\mathbf{A}} w'$. From $R', R'' \in Q$ follows $\mathsf{pub}_{\mathbf{A}}(R', R'')$. Because $w \sim_{\mathbf{A}} w'$ there is a finite chain $w \sim_{a_1} \ldots \sim_{a_n} w'$. Also, because for all $a \in \mathbf{A}$, pub_a is an equivalence relation and therefore reflexive, we have (writing infix) $R'\ \mathsf{pub}_{a_1} \ldots \mathsf{pub}_{a_n}\ R'$; therefore, using the definition of $\sim'$: $(w, R') \sim'_{a_1} \ldots \sim'_{a_n} (w', R')$, and thus $(w, R') \sim'_{\mathbf{A}} (w', R')$. Similarly for $(w', R') \sim'_{\mathbf{A}} (w', R'')$. We now have $(w, R') \sim'_{\mathbf{A}} (w', R') \sim'_{\mathbf{A}} (w', R'')$, and therefore, as $\sim'_{\mathbf{A}}$ is an equivalence relation, $(w, R') \sim'_{\mathbf{A}} (w', R'')$. ■

Example 18 (Executing $\mathsf{show}^{2,-}_{1,r}$)
We apply definition 14 to the knowledge game state $(\mathsf{hexa}, \mathsf{rwb})$ and the game action $\langle 2, \{\{rwb, rbw\}, \{wrb, wbr\}, \{brw, bwr\}\}, 1, \{rwb, rbw\}, \mathsf{show}\rangle$ also abbreviated as $\mathsf{show}^{2,-}_{1,r}$. Indeed the computations then result in the knowledge game state on the right in figure 2.2 on page 17. In figure 2.3 we visualize the construction. In the figure, we abbreviate $\{rwb, rbw\}$ as r, $\{wrb, wbr\}$ as w, and $\{brw, bwr\}$ as b. E.g. in the middle figure we have that $(rbw, r) \sim_3 (brw, b)$ because $rbw \sim_3 brw$ and $\mathsf{show}_3(r, b)$. In the figure on the extreme right, we follow the convention that we name worlds by the deals that atomically characterize them.

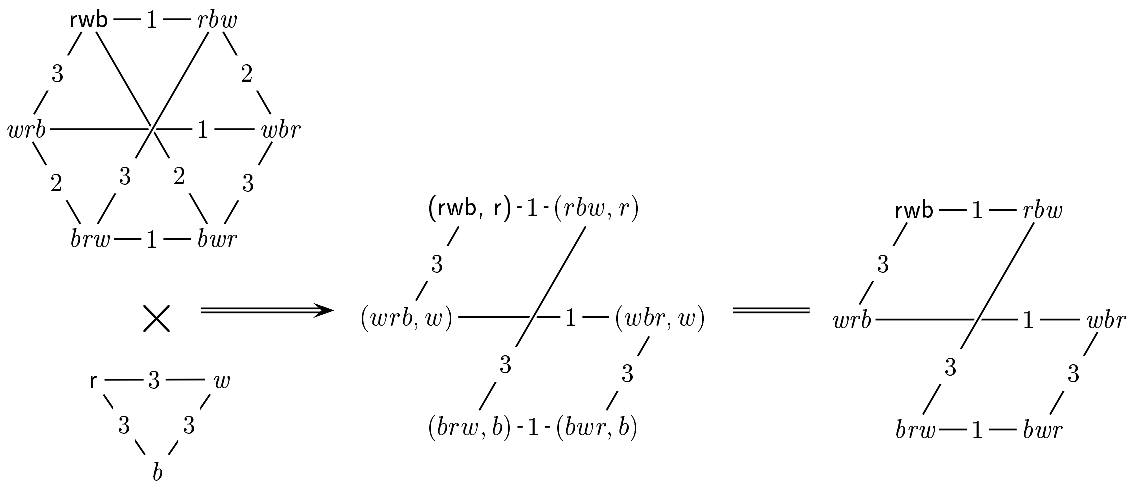


Figure 2.3: Executing action $\mathsf{show}^{2,-}_{1,r}$ in state $(\mathsf{hexa}, \mathsf{rwb})$

These matters are discussed in greater detail in chapters 4 and 5. We already mentioned in subsection 2.4.2 that the knowledge action $L_{123}(!L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1)$ corresponds to the game action $\mathsf{show}^{2,-}_{1,r}$. We can now explain what that correspondence is: the knowledge action $L_{123}(!L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1)$ is interpreted as a state transformer, i.e. as a relation between states. The pointed frame for the game action $\mathsf{show}^{2,-}_{1,r}$ is a semantic object. However, its execution induces exactly the same relation between knowledge states, as that of the 'corresponding' knowledge action. The notion of an action as a semantic object is similar to that in [Bal99]. See chapter 7.

2.6 Playing knowledge games

In the previous sections we have defined what a state of the game is, what an action in the game is, and how to compute the next state of the game. We have not actually played knowledge games. It will be clear that only with all this groundwork covered we can start to think about optimal strategies for playing knowledge games. To investigate this is relevant for game theory, because playing knowledge games is nothing but proceduralized information exchange in groups of competitors, where the value of questions and answers depends on their content and on the group members that receive that information.

First, we have to define the *rules* of the game. Then, we can describe strategies. Only then, we can compute optimal strategies. In this section, we will touch upon these different topics. Beyond this section, we will, regrettably, not pursue them further in our research.

Rules for playing knowledge games The game actions that occur in knowledge games are of the sort: show, noshow, win and nowin. A player can ask another player for one of some cards, after which the other player replies by saying that he doesn't have them, or by showing to the requesting player one of the requested cards. And a player can declare to have won the game, or, by finishing his move, 'declare' that he cannot win yet.

The order protocol restricts the order of actions in a game. These restrictions are computed from the history of requesting players, i.e. from the first argument of game actions. A play of the game consists of an alternating sequence of either show or noshow actions followed by nowin actions, where the last of the show or noshow actions is followed by a win action, the final action in the play. This means that, after a show or noshow action, we always allow the requesting player to announce that he has won. We could have chosen a different protocol, where he also is allowed to *guess*, see below.

It is randomly determined who starts the game, i.e. who asks the first question. The table is, naturally, *not* allowed to ask a question. Drawing a card from the table is not permitted as an action.

Guessing We only allow players to announce that they *know* the deal of cards (or to announce knowledge of another winning condition). Instead, we may allow players to *guess* the deal of cards. This is just as in the real Cluedo game. A player may only guess once during the game; if his guess is wrong, he lost. We can define a new kind of knowledge game: a player may *either* ask for a card, in which case a show or noshow action results, *or* he may guess the deal of cards, in which case a win or nowin action results.[5] He may not do both during his turn, as we allowed before. Therefore it is no longer the case, that after a show or noshow action, a player performs a nowin action, unless he wins: now a player may know, but he is only allowed to announce that knowledge in his next turn. In the 'guessing game' a show or noshow action is always *preceded* by a nowin action, instead of *almost* always followed by it. This suggests that we can redefine the game actions show and noshow by adding an extra constraint on the question.

[5] Suggested by Ariel Rubinstein, personal communication

Winning the game In the example of three players and three cards, winning was publicly announcing knowledge of the deal of cards. We can define weaker criteria for knowledge required to win, e.g. knowledge of the cards of one particular player. An example is knowledge of the cards on the table, as in Cluedo.

Game theory Knowledge games are competitive games of imperfect information, where the only final outcomes are that players can win or lose. (See e.g. [OR94, Bin92] for a general introduction.) The value of such a game is the probability for the starting player to win, given that players follow optimal strategies. These are mixed strategies. For very simple knowledge games we can draw the entire game tree and by backwards induction compute the value of the game. It will be obvious that the value of a hexa game is 1: the beginner can always win in the first move. A slightly more complex knowledge game is that for two players and five cards, where each player holds two cards and one card is lying on the table. Winning is announcing knowledge of the card lying on the table. Assume that players are not allowed to ask the same question twice, so that the game tree is finite. Both when a player is only allowed to ask for one card, and when a player is only allowed to ask for one of two cards, the value of the game is $\frac{7}{9}$. Does the first player always have the highest probability to win a knowledge game? What is the value of Cluedo? In general, we have not answered these questions. The individual preference relation for a player on the different questions that he can ask, depends on the answers that he expects, on the probability distribution of these answers, and on how they refine the partitions for all players. This is quite hard to compute, if possible at all, see [Koo00].

2.7 Conclusion

We have defined the concepts of knowledge game, deal of cards, knowledge game state, game action, and action execution. Questions and answers in games can be modelled as game actions. We now can describe in mathematical detail actual card games and card requests and responses in those games. Only given this precise definition of game states and game actions, can we start to think about optimal strategies for playing such games. Our results are relevant to the analysis of communication in groups.

Chapter 3

Descriptions of game states

In chapter 2 we have defined knowledge games as a concrete example of games where information change may be rather complex and its description therefore of interest both to logicians and game theorists. From a given knowledge game state and a game action that is executable in that state we can compute the next game state. Therefore, we can compute any game state from an initial knowledge game state and a game action sequence. This illustrates the need for a logical description of initial game states. Although it seems to be rather clear what the players know in an initial game state, their ignorance is less transparent, as one easily overlooks game features. In this chapter we provide descriptions of initial knowledge game *states*. The descriptive language is multiagent epistemic logic. We assume a working knowledge of epistemic logic, see appendix A for an overview, or [MvdH95, FHMV95, BdRV00, HC84] for a general introduction. We generally use notation as in [MvdH95]. The description of game *actions* will be postponed to chapter 6. First we need to define a language for dynamic epistemics, in chapter 4.

Why does a model not suffice but do we need its characterization, i.e. its logical description? The model hexa models three players each holding a card.[1] Many have never seen this model and may not even know how to interpret such a relational structure, but can probably play a hexa game perfectly well. They will even consider the game trivial: just ask about any card that you don't know and you will win. The model hexa encodes what players know about their cards and about each other. But when they play, i.e. when they reason about their knowledge, players do not use hexa but use a *description* of hexa. The description lists properties of hexa such as 'player 1 holds (at least) one card': $r_1 \vee w_1 \vee b_1$. That players use descriptions of models is even more obvious when we consider a non-trivial knowledge game as Cluedo.

In section 3.1 we present the theory $\mathsf{33}$ that describes hexa. We prove that indeed $\mathsf{33}$ *describes* hexa: all $S5_3$ models of $\mathsf{33}$ are bisimilar to hexa. From $\mathsf{33}$ follow various other formulas, that describe *properties* of the players' knowledge.

In section 3.2 we continue with the general case: the knowledge game for a deal $\mathsf{d} \in \mathbf{A}^{\mathbf{C}}$ of $|\mathbf{C}|$ cards over $|\mathbf{A}| = n$ players. The $S5_n$ model I_{d} models its initial

[1] See figure 3.1 on page 30, or chapter 2

game state. We present the theory kgames for parameter d that describes I_{d}: all $S5_n$ models of kgames are bisimilar to I_{d}. From kgames follow various other descriptions of the players' knowledge, in particular, different ways to express ignorance, and different ways to express knowledge of your own cards.

In section 3.3 we discuss the game state where the cards have been dealt but where players haven't picked up and looked into their own cards. It has a simpler intended model $preI_{\mathsf{d}}$ and a simpler description prekgames. Again, we prove that $preI_{\mathsf{d}}$ is unique. The model I_{d} results from $preI_{\mathsf{d}}$ by executing the action look of 'turning cards', thus providing an indirect proof that kgames describes I_d. Details are given in chapter 6.

In section 3.4, we compute the descriptions of I_{d} and $preI_{\mathsf{d}}$ by a fixed point construction for finite models as in [vB98, BM96], and we show how the results relate to kgames and prekgames, respectively.

3.1 Description of hexa

In this section, we present the theory 33, that describes the $S5_3$ model hexa. The knowledge state (hexa, rwb) has been introduced in chapter 2. It is the initial state of the knowledge game for three players each holding a card, where 1 holds red, 2 holds white, and 3 holds blue. Figure 3.1 pictures the model hexa.

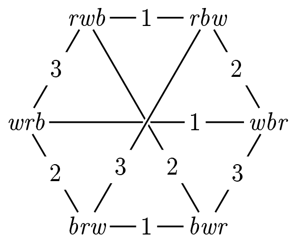


Figure 3.1: The model hexa for three players each holding a card

What information do the players have in this model, regardless of the actual deal of cards? They know how many cards there are, namely three. They know that the cards are all different, namely one red, one white and one blue. They know that each of them holds one card. Beyond that, if they hold a card, they know it, and if they don't hold a card, they also know that they do not hold it. All this is publicly known. They don't know anything else, and there seem to be two sides of that ignorance. First, a player doesn't know that another player holds a specific card. Second, with the exception of his own card, a player can imagine any card to be in possession of another player. All these constraints are satisfied by the theory 33 (which can be regarded as either a set of formulas or as the conjunction of these formulas).

Definition 15 (33)
Theory 33 consists of the following three formulas:

$$\begin{array}{rcl} \mathsf{see33} & := & \bigwedge_{a\in\{1,2,3\}} \bigwedge_{c\in\{r,w,b\}} (c_a \rightarrow K_a c_a) \\ \mathsf{deals33} & := & \delta_{rwb} \vee \delta_{rbw} \vee \delta_{wrb} \vee \delta_{wbr} \vee \delta_{brw} \vee \delta_{bwr} \\ \mathsf{dontknowthat33} & := & \bigwedge_{a\neq a'\in\{1,2,3\}} \bigwedge_{c\in\{r,w,b\}} \neg K_a c_{a'} \end{array}$$

See33 expresses that every agent can see his own card. Deals33 expresses that there are only six deals of size 1|1|1. For δ_{abc}, read $a_1 \wedge \neg b_1 \wedge \neg c_1 \wedge \neg a_2 \wedge b_2 \wedge \neg c_2 \wedge \neg a_3 \wedge \neg b_3 \wedge c_3$. This is the (atomic) description of (world) deal abc. The concept 'description of a deal' will be introduced for the more general case in definition 18 in section 3.2. Dontknowthat33 expresses that players do not know the cards of other players. In section 3.1.1 we discuss various properties of agent knowledge in hexa that can be derived from the concise formulation in 33.

When we say that 33 describes hexa, we have implicitly quantified over all its worlds. This has to be made explicit when we describe one of its states. Any state (hexa, d) is described by the conjunction of its atomic description and common knowledge of the theory 33. For example, $\delta_{rwb} \wedge C_{123}33$ describes the initial state of the game where 1 holds red, 2 holds white and 3 holds blue.

Fact 2
$\mathsf{hexa} \models 33$

Proof Obvious. ■

We cannot substantially *weaken* the theory (by deleting formulas), because it then would model structures of different game states. We now show that we do not need to *strengthen* the theory, because in a technical sense hexa already is its only model (33 defines the bisimulation class of hexa). This shows that we have chosen the right model, and the right description, for the game state of three players each holding a card.

Proposition 4
Let $M = \langle W, \{\sim_1, \sim_2, \sim_3\}, V\rangle$ be an $S5_3$ model of 33, i.e. $M \models 33$. Then M is bisimilar to hexa.

Proof Write $\mathsf{hexa} = \langle W^h, \{\sim_1^h, \sim_2^h, \sim_3^h\}, V^h\rangle$, where $W^h = \{rwb,\ rbw,\ brw,\ bwr,\ wrb,\ wbr\}$, $\sim_1^h = \{\{rwb, rbw\},\ \{brw, bwr\},\ \{wrb, wbr\}\}$, $\sim_2^h = \{\{rwb, bwr\},\ \{rbw, wbr\},\ \{wrb, brw\}\}$, $\sim_3^h = \{\{rwb, wrb\},\ \{wbr, bwr\},\ \{rbw, brw\}\}$, $V^h_{ijk} = V_{ijk}$ such that: $V_{ijk}(i_1) = V_{ijk}(j_2) = V_{ijk}(k_3) = 1$ and $V_{ijk}(p) = 0$ for all other (six) atomic propositions p.

First an observation on valuations of worlds in M. Because $M, w \models \mathsf{deals33}$, and because each one of the six exclusive alternatives in deals33 correspond to a valuation, any world $w \in M$ has one of six different valuations V_{rwb}, V_{rbw}, V_{brw}, V_{bwr}, V_{wbr}, V_{wrb}.

Now define relation $\mathfrak{R} \subseteq (W \times W^h)$ ($\mathfrak{R} \subseteq (\mathcal{D}(M) \times \mathcal{D}(\mathsf{hexa}))$) as follows:

$$\forall w \in M : \forall w^h \in \mathsf{hexa} : \mathfrak{R}(w, w^h) \Leftrightarrow V_w = V^h_{w^h}$$

We prove that $\mathfrak{R}$ is a bisimulation between M and hexa.

Forth: Let $w, w' \in M$, let $w^h \in \mathsf{hexa}$. Suppose $w \sim_1 w'$ and $\mathfrak{R}(w, w^h)$. We find an $\mathfrak{R}$-image of w' for every valuation V_w on w. First suppose $V_w = V_{rwb}$. From $\mathfrak{R}(w, w^h)$ follows $V^h_{w^h} = V_w = V_{rwb}$. Therefore $w^h = rwb$.

As M is a model of 33, $M \models \mathsf{see33}$. From $M, w \models \mathsf{see33}$ follows $M, w \models r_1 \rightarrow K_1 r_1$. From $V_w(r_1) = V_{rwb}(r_1) = 1$ and $M, w \models r_1 \rightarrow K_1 r_1$ follows $M, w \models K_1 r_1$. From $M, w \models K_1 r_1$ and $w \sim_1 w'$ follows $M, w' \models r_1$. Therefore $V_{w'} = V_{rwb}$ or $V_{w'} = V_{rbw}$. If $V_{w'} = V_{rwb}$, choose rwb as the $\mathfrak{R}$-image of w' in hexa: obviously $rwb \sim^h_1 rwb$ and also $\mathfrak{R}(w', rwb)$. If $V_{w'} = V_{rbw}$, choose rbw as the $\mathfrak{R}$-image of w' in hexa: we now have $rwb \sim^h_1 rbw$ and $\mathfrak{R}(w', rbw)$.

Similarly for the five other valuations V_w on w. Similarly for $i = 2$ and $i = 3$.

Back: Let $w^h, w^h_* \in \mathsf{hexa}$, let $w \in M$. Suppose $w^h \sim^h_1 w^h_*$ and $\mathfrak{R}(w, w^h)$. We find an $\mathfrak{R}$-original of w^h_* for every valuation $V^h_{w^h}$ on w^h. First suppose $V^h_{w^h} = V_{rwb}$. Obviously $w^h = rwb$.

From $rwb \sim^h_1 w^h_*$ follows $w^h_* = rwb$ or $w^h_* = rbw$. If $w^h_* = rwb$ choose w itself as the required $\mathfrak{R}$-original of w^h_*. As M is an *S5* model, $w \sim_1 w$, and we already assumed $\mathfrak{R}(w, rwb)$. Otherwise $w^h_* = rbw$. We derive a contradiction from the assumption that there is *no* $w' \in M$ such that $w \sim_1 w'$ and $V_{w'} = V_{rbw}$.

Suppose so. In other words: for all $w' \in M : w \sim_1 w' \Rightarrow V_{w'} \neq V_{rbw}$. Suppose $w \sim_1 w'$. As before, from see33 follows $M, w \models K_1 r_1$ and from that and $w \sim_1 w'$ follows $M, w' \models r_1$ and therefore $V_{w'} = V_{rwb}$ or $V_{w'} = V_{rbw}$. From that and the assumption follows $V_{w'} = V_{rwb}$, thus $M, w' \models w_2$, and thus, as w' is an arbitrary 1-accessible world from $w \in M$, $M, w \models K_1 w_2$. However, also $M \models \mathsf{dontknowthat33}$, thus $M, w \models \neg K_1 w_2$. Contradiction.

Therefore there is a $w' \in M$ such that $w \sim_1 w'$ and $V_{w'} = V_{rbw}$. By definition we have $\mathfrak{R}(w', rbw)$. So we have found the required $\mathfrak{R}$-original of rbw.

Similarly for the five other valuations $V^h_{w^h}$ on w^h. Similarly for $i = 2$ and $i = 3$. ■

We also present a more general version of this proposition, namely for any number of players and cards (proposition 5 in section 3.2). Compared to that proof, the proof for hexa is more explicit in *what* part of the description of hexa is (only) needed in *what* direction of the bisimulation: see33 is essential in the *forth* part of the proof, dontknowthat33 is essential in the *back* part of the proof. We present both proofs, because this distinction cannot be made in the proof of proposition 5, as see33 and dontknowthat33 are combined into one formula for the general case.

3.1.1 Derived characteristics of hexa

One can define various other characteristics of hexa. We list a few:

Definition 16 (Other properties of agents' knowledge)

$$\begin{array}{rcl}
 & & \textit{players only see their own cards} \\
\mathsf{dontsee33} & := & \bigwedge_{a\in\{1,2,3\}} \bigwedge_{c\in\{r,w,b\}} (\neg c_a \to K_a \neg c_a) \\
 & & \textit{there is at most one card of each colour} \\
\mathsf{atmost33} & := & \bigwedge_{a\neq b\in\{1,2,3\}} \bigwedge_{c\in\{r,w,b\}} \neg(c_a \wedge c_b) \\
 & & \textit{there is at least one card per player} \\
\mathsf{atleast33} & := & \bigwedge_{a\in\{1,2,3\}} (r_a \vee w_a \vee b_a) \\
 & & \textit{there is exactly one card per player} \\
\mathsf{exactly33} & := & \bigwedge_{a\in\{1,2,3\}} (r_a \triangledown w_a \triangledown b_a) \\
 & & \textit{players can imagine that others hold other cards} \\
\mathsf{dontknownot33} & := & \bigwedge_{a\neq b\in\{1,2,3\}} \bigwedge_{c\in\{r,w,b\}} (\neg c_a \to \neg K_a \neg c_b) \\
 & & \textit{players do not know that others hold other cards} \\
\mathsf{dontknowother33} & := & \bigwedge_{a\neq b\in\{1,2,3\}} \bigwedge_{c\in\{r,w,b\}} (\neg c_a \to \neg K_a c_b)
\end{array}$$

In exactly33, $\triangledown$ is 'exclusive or'. All of these hold in 33. If property φ holds in hexa, then $C_{123}33 \models_{S5EC_3} C_{123}\varphi$.[2] Because the three constituents of 33 and the derived characteristics hold in all worlds of hexa, there is an implicit common knowledge operator C_{123}, that we have to make explicit for logical entailment.

Definition 17 (Equally strong)
Let Σ be a set of $\mathcal{L}_n^C$ formulas (given a set $\mathbf{A}$ of n agents). Then 'φ is equally strong as ψ in Σ' (or 'just as strong as'), notation $\varphi \Leftrightarrow_\Sigma \psi$, if both $C_\mathbf{A}(\Sigma-\psi+\varphi) \models_{S5EC_n} C_\mathbf{A}\psi$ and $C_\mathbf{A}(\Sigma-\varphi+\psi) \models_{S5EC_n} C_\mathbf{A}\varphi$.

We can show that:

$$\begin{array}{rcl}
\mathsf{atmost33} \wedge \mathsf{atleast33} & \Leftrightarrow_{33} & \mathsf{deals33} \\
\mathsf{exactly33} & \Leftrightarrow_{33} & \mathsf{deals33} \\
\mathsf{see33} & \Leftrightarrow_{33} & \mathsf{dontsee33} \\
\mathsf{dontknowthat33} & \Leftrightarrow_{33} & \mathsf{dontknowother33} \\
\mathsf{dontknowthat33} & \Leftrightarrow_{33} & \mathsf{dontknownot33}
\end{array}$$

Although the proofs need some combinatorial juggling, they are rather basic and have been omitted. The proof of proposition 6 in section 3.2 gives an example.

[2] By $\varphi \models_\mathcal{M} \psi$ we mean that all models in class $\mathcal{M}$ that satisfy φ also satisfy ψ; $S5EC_3$ is the class of $S5_3$ models plus access computed on these models for general (E) and common (C) knowledge operators. See appendix A or [MvdH95].

3.2 Description of initial game states

In the previous section we have described hexa. In this section we generalize our results for any number of players and cards. Let $\mathsf{d} \in \mathbf{A}^{\mathbf{C}}$ be an actual deal of a set $\mathbf{C}$ of cards over a set $\mathbf{A}$ of n players. The state of the game where these cards have been dealt and where everybody has (only) looked in his cards is modelled by the pointed $S5_n$ model $(I_{\mathsf{d}}, \mathsf{d}) = (\langle D_{\sharp \mathsf{d}}, (\sim_a)_{a \in \mathbf{A}}, V \rangle, \mathsf{d})$, where (for all a, c, d', d'') $d' \sim_a d'' \Leftrightarrow d'^{-1}(a) = d''^{-1}(a)$ and $V_{d'}(c_a) = 1 \Leftrightarrow d'(c) = a$. The set $D_{\sharp \mathsf{d}}$ is the set of all deals of size d. See also chapter 2. In this section we describe the underlying model I_{d}. First, we introduce the concept of *description of a deal.*

Definition 18 (Description of a deal of cards)
Let $d \in \mathbf{A}^{\mathbf{C}}$. $\mathbf{P}$ is the set of all atoms c_a, for $a \in \mathbf{A}$ and $c \in \mathbf{C}$. Define, for all $c_a \in \mathbf{P}$: $sign_d(c_a) := c_a$ if $d(c) = a$ and $sign_d(c_a) := \neg c_a$ if $d(c) \neq a$. Then:

$$\begin{array}{rcl} \delta_d & := & \bigwedge_{c_a \in \mathbf{P}} sign_d(c_a) \\ \delta_d^a & := & \bigwedge_{c \in \mathbf{C}} sign_d(c_a) \end{array}$$

Formula δ_d is called the description of deal of cards d. Formula δ_d^a is called the description of the cards of player a. In a model where a world w is characterized by a deal d, δ_d is *the atomic description of the world w.* The following useful equivalences hold:

$$\begin{array}{rcl} \delta_d & \leftrightarrow & \bigwedge_{a \in \mathbf{A}} \delta_d^a \\ \delta_d^a & \leftrightarrow & \bigvee_{d' \sim_a d} \delta_{d'} \end{array}$$

We now present the theory kgames:

Definition 19 (kgames)
The theory kgames for deal $\mathsf{d} \in \mathbf{A}^{\mathbf{C}}$ consists of the two formulas:

$$\begin{array}{rcl} \mathsf{deals} & := & \bigvee_{d' \in D_{\sharp \mathsf{d}}} \delta_{d'} \\ \mathsf{seedontknow} & := & \bigwedge_{a \in \mathbf{A}} \bigwedge_{d' \in D_{\sharp \mathsf{d}}} (K_a \delta_{d'}^a \leftrightarrow M_a \delta_{d'}) \end{array}$$

Deals expresses that every world is atomically characterized by a deal of size $\sharp \mathsf{d}$. $\mathsf{Seedontknow}$ expresses that agents only consider deals that correspond to what they know of (the description of) their own cards. Similarly as for hexa, an initial state $(I_{\mathsf{d}}, \mathsf{d})$ is described by $\delta_{\mathsf{d}} \wedge C_{\mathbf{A}} \mathsf{kgames}$. Because we have that $K_a \delta_d^a \rightarrow \delta_d^a$ $(S5)$, $\mathsf{seedontknow}$ is equivalent to the conjunction of the two formulas $\mathsf{seedeal} := \bigwedge_{a \in \mathbf{A}} \bigwedge_{d \in D_{\sharp \mathsf{d}}} (\delta_d^a \rightarrow K_a \delta_d^a)$ and $\mathsf{dontknow} := \bigwedge_{a \in \mathbf{A}} \bigwedge_{d \in D_{\sharp \mathsf{d}}} (\delta_d^a \leftrightarrow M_a \delta_d)$. In a way, $\mathsf{seedeal}$ expresses the *private knowledge* of a player, $\mathsf{dontknow}$ expresses his *private ignorance*, and deals expresses his *public knowledge.* In section 3.2.1 we discuss these and various other characterizations of the players' knowledge in I_{d}, and in what respect they are generalizations of constituents of 33. We allow ourselves the abus de language to call both $\mathsf{seedeal}$ and $\mathsf{dontknow}$ also constituents of kgames, so that we can speak of other formulas as being equally strong as $\mathsf{seedeal}$ or $\mathsf{dontknow}$ in kgames.

Fact 3
I_{d} is a model of kgames.

Proof Obvious. ■

Just as for the case of three persons and three cards, we cannot substantially *weaken* the theory. We now show that we also do not need to *strengthen* the theory, because I_{d} is its only model. Together, this shows that we have chosen the right model and the right description for the game state where a finite number of cards are dealt over a finite number of players.

Proposition 5
Let M be an $S5_n$ model of theory kgames for deal d. Then M is bisimilar to I_{d}.

Proof Write $M = \langle W^M, (\sim_a^M)_{a\in\mathbf{A}}, V^M\rangle$. We have that $M \models \mathsf{kgames}$. Write $I_{\mathsf{d}} = \langle D_{\sharp\mathsf{d}}, (\sim_a)_{a\in\mathbf{A}}, V\rangle$, for the intended initial model I_{d}. Observe that, because $M \models \mathsf{deals}$, each world $w \in M$ has a valuation $V_w = V_d$ for some $d \in D_{\sharp\mathsf{d}}$. Define relation $\mathfrak{R} \subseteq (W^M \times D_{\sharp\mathsf{d}})$ as follows:

$$\forall w \in M : \forall d \in D_{\sharp\mathsf{d}} : \mathfrak{R}(w,d) \Leftrightarrow V_w = V_d$$

We prove that $\mathfrak{R}$ is a bisimulation between M and I_{d}.

Forth: Let $w, w' \in M$, let $d \in D_{\sharp\mathsf{d}}$. Suppose that $\mathfrak{R}(w,d)$ and that, for an arbitrary $a \in \mathbf{A}$: $w \sim_a w'$. We find an $\mathfrak{R}$-image of w', in $D_{\sharp\mathsf{d}}$, as follows:

Observe that $I_{\mathsf{d}}, d \models \delta_d$. As $V_w = V_d$, also $M, w \models \delta_d$. Therefore $M, w \models M_a\delta_d$. From that and $M, w \models \mathsf{seedontknow}$ follows $M, w \models K_a\delta_d^a$. From that and $w \sim_a^M w'$ follows $M, w' \models \delta_d^a$, i.e.: $M, w' \models \bigvee_{d'\sim_a d}\delta_{d'}$. Therefore there is a $d' \sim_a d$ such that $M, w' \models \delta_{d'}$. *That* d' is the required $\mathfrak{R}$-image of w': note that $d \sim_a d'$, and that $V_{w'}^M = V_{d'}$, because also, obviously, $I_{\mathsf{d}}, d' \models \delta_{d'}$.

Back: Let $d, d' \in I_{\mathsf{d}}$, let $w \in M$. Suppose that $\mathfrak{R}(w,d)$ and that, for an arbitrary $a \in \mathbf{A}$, $d \sim_a d'$. We find an $\mathfrak{R}$-original of d', in M, as follows:

$M, w \models \delta_d$
$\Rightarrow$ reflexivity
$M, w \models M_a\delta_d$
$\Leftrightarrow$ from $\mathsf{seedontknow}$
$M, w \models K_a\delta_d^a$
$\Leftrightarrow$
$\forall w'' \sim_a w : M, w'' \models \delta_d^a$
$\Leftrightarrow$
$\forall w'' \sim_a w : M, w'' \models \bigvee_{d''\sim_a d}\delta_{d''}$
$\Leftrightarrow$
$\forall w'' \sim_a w : \exists d'' \sim_a d : M, w'' \models \delta_{d''}$

$\Leftrightarrow$ as $d \sim_a d'$
$\forall w'' \sim_a w : \exists d'' \sim_a d' : M, w'' \models \delta_{d''}$
$\Leftrightarrow$
$\forall w'' \sim_a w : M, w'' \models \bigvee_{d'' \sim_a d'} \delta_{d''}$
$\Leftrightarrow$
$\forall w'' \sim_a w : M, w'' \models \delta^a_{d'}$
$\Leftrightarrow$
$M, w \models K_a \delta^a_{d'}$
$\Leftrightarrow$ from seedontknow
$M, w \models M_a \delta_{d'}$
$\Rightarrow$
$\exists w' \sim_a w : M, w' \models \delta_{d'}$

Any w' satisfying the last statement is an $\mathfrak{R}$-original of d', as $M, w' \models \delta_{d'}$ says that $V_{w'} = V_{d'}$.

Note that in the 'forth' part of the proof, we have only essentially used that, for any agent a and deal d', $M_a \delta_{d'} \rightarrow K_a \delta^a_{d'}$, whereas in the 'back' part of the proof, we have also essentially used the reverse: $K_a \delta^a_{d'} \rightarrow M_a \delta_{d'}$. Further, note that, in the proof, we only use reflexivity of models; the proposition therefore holds for all T_n (merely reflexive) models. ■

Instead of this direct proof, there is also an indirect proof. The indirect proof uses that the model I_{d} can be constructed by executing an action in a simpler model for card games. See section 3.3.

3.2.1 Derived characteristics of initial knowledge game states

We already mentioned that in $S5$, seedontknow is equivalent to the conjunction of seedeal and dontknow. We discuss various other derived characteristics. We list six formulas describing what players 'see', that are all equally strong in kgames.

Let $\mathsf{d} \in \mathbf{A}^{\mathbf{C}}$, $a, b \in \mathbf{A}$. We write $\sharp a$ for $|\mathsf{d}^{-1}(a)|$, $\sharp\neg a$ for $|\mathbf{C}| - |\mathsf{d}^{-1}(a)|$, and (to be used later) $\sharp\neg ab$ for $|\mathbf{C}| - |\mathsf{d}^{-1}(a)| - |\mathsf{d}^{-1}(b)|$. As $\sharp\mathsf{d}$ is the size of deal d, that lists the number of cards per player, we have that $(\sharp\mathsf{d})a = \sharp a$.

Definition 20 (Seeing cards)

$$\begin{array}{rcl} \mathsf{see} & := & \bigwedge_{a\in\mathbf{A}} \bigwedge_{c\in\mathbf{C}} (c_a \to K_a c_a) \\ \mathsf{dontsee} & := & \bigwedge_{a\in\mathbf{A}} \bigwedge_{c\in\mathbf{C}} (\neg c_a \to K_a \neg c_a) \\ \mathsf{seeall} & := & \bigwedge_{a\in\mathbf{A}} \bigwedge_{c^1\neq\ldots\neq c^{\sharp a}\in\mathbf{C}} (\bigwedge_{i=1}^{\sharp a} c_a^i \to K_a \bigwedge_{i=1}^{\sharp a} c_a^i) \\ \mathsf{dontseeall} & := & \bigwedge_{a\in\mathbf{A}} \bigwedge_{c^1\neq\ldots\neq c^{\sharp\neg a}\in\mathbf{C}} (\bigwedge_{i=1}^{\sharp\neg a} \neg c_a^i \to K_a \bigwedge_{i=1}^{\sharp\neg a} \neg c_a^i) \\ \mathsf{seedeal} & := & \bigwedge_{a\in\mathbf{A}} \bigwedge_{d\in D_{\sharp\mathsf{d}}} (\delta_d^a \to K_a \delta_d^a) \\ \mathsf{dontseedeal} & := & \bigwedge_{a\in\mathbf{A}} \bigwedge_{d\in D_{\sharp\mathsf{d}}} (\neg\delta_d^a \to K_a \neg\delta_d^a) \end{array}$$

See is the straightforward generalization of see33, and dontsee of dontsee33: for every agent and for every single card, if a player holds it, he knows that, and if he doesn't hold it, he knows that too. Instead, seeall and dontseeall express that, if a player holds his given number of cards, he knows them all, and that if he does not hold any other card, he knows that too. Seedeal (for parameter deal d) expresses that, if a player holds his given number of cards and does not hold any other card, he knows that. This is another way of saying that he knows his *local state*: for a given player a, every atom c_a or $\neg c_a$ in δ_{d}^a corresponds to the value of a 'local state variable for that player'. Dontseedeal expresses that, if a player is *not* in a given local state, he knows that too. Somewhat surprisingly, all six forms of seeing are equally strong in kgames. The proofs are simple and use deals. See is the most straightforward of all six, but does not combine nicely with dontknow into dontknowdeal, the actual constituent of kgames.

Similarly to the case for hexa, there are different ways to express how many cards players hold:

Definition 21 (How many cards)

$$\begin{array}{rcl} & & \textit{all cards are different} \\ \mathsf{atmost} & := & \bigwedge_{a\neq b\in\mathbf{A}} \bigwedge_{c\in\mathbf{C}} \neg(c_a \wedge c_b) \\ & & \textit{each player has (at least) a known number of cards} \\ \mathsf{atleast} & := & \bigwedge_{a\in\mathbf{A}} \bigvee_{c^1\neq\ldots\neq c^{\sharp a}\in\mathbf{C}} \bigwedge_{i=1}^{\sharp a} c_a^i \\ & & \textit{each player has exactly a known number of cards} \\ \mathsf{exactly} & := & \bigwedge_{a\in\mathbf{A}} \bigtriangledown_{c^1\neq\ldots\neq c^{\sharp a}\in\mathbf{C}} \bigwedge_{i=1}^{\sharp a} c_a^i \end{array}$$

As in section 3.1, we have that deals is just as strong in kgames as atmost and atleast together, and is just as strong as exactly.

Ignorance What you *know* in the initial state of a knowledge game is rather straightforward: in hexa you know your own card, in the general case you know your own *cards*. What you *don't know* in the initial state of the game, in the general case, is much less clear. We start with two examples.

Example 19
Consider the initial game state for the game for three players $1, 2, 3$ each holding two cards, with actual deal $kl|mn|op$. How ignorant is player 1 in this state of the game? Clearly, it is not strong enough that 1 does not know that 2 holds a specific combination of two cards: for all $c \neq c'$, $\neg K_1(c_2 \wedge c'_2)$. Player 1 also does not know that 2 holds any single card, which is stronger: for all c, $\neg K_1 c_2$. However, even that is not strong enough:

Suppose 2 has told the others that he holds one of m and n. After that, it still holds that 1 doesn't know any of 2's cards: $\neg K_1 m_2$ and $\neg K_1 n_2$. However, 1 is less ignorant than before, because he *now* knows that 2 holds one of *two* cards: $K_1(m_2 \vee n_2)$. Initially, he didn't know that, indeed: for all $c \neq c' \in \mathbf{C}$: $\neg K_1(c_2 \vee c'_2)$.

It appears that this is exactly the limit of his ignorance, because for some combinations of *three* cards he *does* know that 2 holds one of them, e.g. $K_1(m_2 \vee n_2 \vee o_2)$. Suppose not, then $M_1(\neg m_2 \wedge \neg n_2 \wedge \neg o_2)$, or in other words: it would be conceivable for player 1 that player 2 did not have any of these three cards m, n, o. Because player 1 holds two *other* cards himself, k, l, there would then be only one card left for player 2 to hold: card p. That player 2 holds only one card, contradicts deals.

The ignorance in example 19 is generally described in a formula $\mathsf{dontknowthat}$. There are also two other interesting ways to express ignorance in the initial state of a knowledge game:

Definition 22 (Ignorance)

$$\begin{array}{rcl} \mathsf{dontknowthat} & := & \bigwedge_{a \neq b \in \mathbf{A}} \bigwedge_{c^1 \neq \ldots \neq c^{\sharp\neg ab} \in \mathbf{C}} M_a \bigwedge_{i=1}^{\sharp\neg ab} \neg c_b^i \\ \mathsf{dontknownot} & := & \bigwedge_{a \neq b \in \mathbf{A}} \bigwedge_{c^1 \neq \ldots \neq c^{\sharp b} \in \mathbf{C}} (\bigwedge_{i=1}^{\sharp b} \neg c_a^i \rightarrow M_a \bigwedge_{i=1}^{\sharp b} c_b^i) \\ \mathsf{dontknow} & := & \bigwedge_{a \in \mathbf{A}} \bigwedge_{d' \in D_{\sharp \mathsf{d}}} (\delta_{d'}^a \leftrightarrow M_a \delta_{d'}) \end{array}$$

First, we explain what kind of ignorance the formulas express, by generalizing on the ignorance in hexa.

Dontknowthat In hexa it holds that $\neg K_a c_b$, for all cards c and players $a \neq b$. Each player only had one card. In order to generalize this to more than one card, we have to demand that $\neg K_a(c_b^1 \vee \ldots \vee c_b^r)$, for some $r > 1$. What is r, or, in other words: how large is a's ignorance? The extent of player a's ignorance is, that he doesn't know that another player b has one from any of $|\mathbf{C}| - |d^{-1}(a)| - |d^{-1}(b)|$ cards: $\bigwedge_{a \neq b \in \mathbf{A}} \bigwedge_{c^1 \neq \ldots \neq c^{\sharp\neg ab} \in \mathbf{C}} \neg K_a \bigvee_{i=1}^{\sharp\neg ab} c_b^i$. This formula is obviously equivalent to $\mathsf{dontknowthat}$.

Dontknownot In hexa, $\mathsf{dontknownot33}$, expressing that a player can imagine others to hold other cards, was just as strong as $\mathsf{dontknowthat33}$. In the general case, $\neg c_a \rightarrow M_a c_b$ for all cards c and players $a \neq b$, is not strong enough: we cannot derive that a can imagine b to hold a combination of two cards, e.g. What is the largest conjunction of *other* cards that still must be conceivable? In other words: for which r do we have to require that $(\neg c_a^1 \wedge \ldots \wedge \neg c_a^r) \rightarrow M_a(c_b^1 \wedge \ldots \wedge c_b^r)$? Obviously, we cannot imagine a player b to hold *more* than $\sharp b$ cards. Indeed, $r = \sharp b$ is the required maximum.

Dontknow Dontknownot and dontknowthat are unsatisfactory, because they are defined too much in terms of relations *between* players. In dontknow we directly refer to the actual deal of cards: a player considers a deal of cards if, and only if, it corresponds to his own cards. Put in different terms: a player can imagine a global state if, and only if, it corresponds to his local state (cf. seedeal, in definition 20). The following explanation may help to make it appear plausible: Imagine the state of the game where the cards have been dealt but are still lying face down on the table, so that the players do not know their own cards yet. In that state, all players can imagine *all* deals in $D_{\sharp \mathsf{d}}$: $\bigwedge_{a \in \mathbf{A}} \bigwedge_{d \in D_{\sharp \mathsf{d}}} M_a \delta_{d'}$. Looking at cards corresponds to revising that maximum ignorance by conditionalizing on your own cards: $\bigwedge_{a \in \mathbf{A}} \bigwedge_{d \in D_{\sharp \mathsf{d}}} (\delta_d^a \leftrightarrow M_a \delta_d)$. This is dontknow. See also section 3.3.

If there are only two players or there is just one card, all players have full knowledge of the deal of cards and there is no ignorance (the ignorance formulas have become trivial, or have disappeared). Therefore, we assume that there are at least two players and more than one card. It turns out that the three ways to describe ignorance are not equally strong (in kgames), but that dontknow is strongest. The following example[3] clearly shows that. Further, dontknowthat and dontknownot are equally strong (in $\{\mathsf{deals}, \mathsf{see}\}$, i.e. in $\mathsf{kgames} \setminus \mathsf{dontknow}$). This is surprising, because dontknownot and dontknowthat appear to describe complementary kinds of ignorance.

Example 20
Consider the initial game state for the game for four players $1, 2, 3, 4$ each holding one card, with actual deal $k|l|m|n$ ($klmn$). Suppose an outsider tells player 1 that $kmln$ is not the actual deal of cards, so that $K_1 \neg \delta_{kmln}$, i.e. $\neg M_1 \delta_{kmln}$. Deal $kmln$ is consistent with the local state, k, of player 1 in actual deal $klmn$, so that $M_1 \delta_{kmln}$ should follow from dontknow: a contradiction. However, dontknowthat holds: for any two cards, 1 can imagine that 2 does not have both. (Similarly, dontknownot holds.)

[3]Suggested by Erik Krabbe.

Proposition 6 (Ignorance)
Let $\mathsf{d} \in \mathbf{A}^{\mathbf{C}}$. Then:

$$\begin{array}{rcl} \mathsf{dontknow} & \Rightarrow_{\{\mathsf{deals},\mathsf{see}\}} & \mathsf{dontknownot} \\ \mathsf{dontknownot} & \Rightarrow_{\{\mathsf{deals},\mathsf{see}\}} & \mathsf{dontknowthat} \\ \mathsf{dontknowthat} & \Rightarrow_{\{\mathsf{deals},\mathsf{see}\}} & \mathsf{dontknownot} \end{array}$$

$\mathsf{dontknow} \Rightarrow_{\{\mathsf{deals},\mathsf{see}\}} \mathsf{dontknownot}$:

Proof Suppose player a doesn't have any of the cards $c^1, ..., c^{\sharp b}$: $\bigwedge_{i=1}^{\sharp b} \neg c_a^i$. Instead, a has cards $c^{\sharp b+1}, ..., c^{\sharp b+\sharp a}$. Let d^* be a deal of cards where a has those cards and such that b has all the cards $c^1, ..., c^{\sharp b}$, thus $\bigwedge_{i=1}^{\sharp b} c_b^i$. Formula $c_a^{\sharp b+1} \wedge ... \wedge c_a^{\sharp b+\sharp a}$ is the subformula of $\delta_{d^*}^a$ consisting of all positive atoms. Therefore, from $c_a^{\sharp b+1} \wedge ... \wedge c_a^{\sharp b+\sharp a}$ and deals follows $\delta_{d^*}^a$. From that and $\mathsf{dontknow}$ follows $M_a \delta_{d^*}$. Because $\bigwedge_{i=1}^{\sharp b} c_b^i$ is a subformula of the conjunction δ_{d^*}, and because from $\varphi \rightarrow \psi$ follows $\Diamond\varphi \rightarrow \Diamond\psi$, it follows that $M_a \bigwedge_{i=1}^{\sharp b} c_b^i$. Therefore $\bigwedge_{i=1}^{\sharp b} \neg c_a^i \rightarrow M_a \bigwedge_{i=1}^{\sharp b} c_b^i$. As the cards $c^1, ..., c^{\sharp b}$ were arbitrary, we have shown that $\mathsf{dontknownot}$.

$\mathsf{dontknownot} \Rightarrow_{\{\mathsf{deals},\mathsf{see}\}} \mathsf{dontknowthat}$:

Proof Suppose not. Then there are players a, b and cards $c^1, ..., c^{\sharp\neg ab}$ such that $K_a \bigvee_{i=1}^{\sharp\neg ab} c_b^i$. Regardless of whether a holds some of the cards $c^1, ..., c^{\sharp\neg ab}$ himself, there must be at least $\sharp b$ other cards that a doesn't hold, suppose: $ca^1, ..., ca^{\sharp b}$. In other words, we have that: $\neg ca_a^1 \wedge ... \wedge \neg ca_a^{\sharp b}$. Applying $\mathsf{dontknownot}$ we get $M_a \bigwedge_{i=1}^{\sharp b} ca_b^i$. Formula $M_a \bigwedge_{i=1}^{\sharp b} ca_b^i$ means that a can imagine that b holds the $\sharp b$ cards $ca^1, ..., ca^{\sharp b}$. From $K_a \bigvee_{i=1}^{m} c_b^i$ it follows that a knows that b holds at least one more card, namely one of the (other!) cards $c^1, ..., c^m$. Therefore, a can imagine that b holds more than $\sharp b$ cards. From deals (that is known by a) follows that b holds exactly $\sharp b$ cards. Contradiction.

$\mathsf{dontknowthat} \Rightarrow_{\{\mathsf{deals},\mathsf{see}\}} \mathsf{dontknownot}$:

Proof Let $a \neq b$, $c^1 \neq ... \neq c^{\sharp b} \in \mathbf{C}$, and $\bigwedge_{i=1}^{\sharp b} \neg c_a^i$. Further, suppose $ca^1 \neq ... \neq ca^{\sharp a} \in \mathbf{C}$ and $\bigwedge_{i=1}^{\sharp a} ca_a^i$ (so that all c^i and ca^j are different too). Note that there are $\sharp\neg ab$ other cards left. If b does not hold all these other cards, and given that a holds $ca^1 \neq ... \neq ca^{\sharp a}$, b must necessarily hold the cards $c^1, ..., c^{\sharp b}$. From that, and because a can imagine b not to hold these other cards (which follows from $\mathsf{dontknowthat}$), and because a knows his own cards (which follows from see), a can imagine b to hold the cards $c^1, ..., c^{\sharp b}$: $M_a \bigwedge_{i=1}^{\sharp b} c_b^i$. Therefore $\bigwedge_{i=1}^{\sharp b} \neg c_a^i \rightarrow M_a \bigwedge_{i=1}^{\sharp b} c_b^i$. ■

3.3 Description of the pre-initial state

We have described the initial state of a knowledge game, where the cards have been dealt and where players have looked at their cards. Now imagine that the cards have been dealt but that the players have not looked at their cards yet. Assume that everybody can see how many cards lie face down in front of each player. Then players know the *size* ($\sharp\mathsf{d}$) of the actual deal: for each player they know how many cards he has. They do not know anything else: they consider every deal of that size a possibility. We call this state the pre-initial state. (Note that it is *not* a knowledge game state, because it does not fullfil the requirement that players know their own cards, see chapter 2.)

Definition 23 (Pre-initial state)
In the state $(preI_{\mathsf{d}}, \mathsf{d})$, *all* deals of size $\sharp\mathsf{d}$ are possible for all players, i.e. each player's access on $D_{\sharp\mathsf{d}}$ is the universal relation:

$$preI_{\mathsf{d}} = \langle D_{\sharp\mathsf{d}}, (\sim_a)_{a\in\mathbf{A}}, V\rangle$$

where

$$\begin{array}{rc} \forall a \in \mathbf{A} : \forall d'', d' \in D_{\sharp\mathsf{d}} : & d'' \sim_a d' \\ \forall d' \in D_{\sharp\mathsf{d}} : \forall c_a \in \mathbf{P} : & V_{d'}(c_a) = 1 \text{ iff } d'(c) = a \end{array}$$

The theory $\mathsf{prekgames}$ for deal d describes the model $preI_{\mathsf{d}}$:

Definition 24 (Theory prekgames for parameter deal d)
Let $\mathsf{d} \in \mathbf{A}^{\mathbf{C}}$ be a deal of cards. The theory $\mathsf{prekgames}$ (for deal d) consists of the following formulas:

$$\begin{array}{rcl} \mathsf{deals} & := & \bigvee_{d'\in D_{\sharp\mathsf{d}}} \delta_{d'} \\ \mathsf{dontknowany} & := & \bigwedge_{a\in\mathbf{A}} \bigwedge_{d'\in D_{\sharp\mathsf{d}}} M_a\delta_{d'} \end{array}$$

Fact 4
$preI_{\mathsf{d}}$ is a model of $\mathsf{prekgames}$.

Proof Obvious. ■

First we prove that the theory describes this model: all other models of $\mathsf{prekgames}$ are bisimilar to $preI_{\mathsf{d}}$. Then we show that the model I_{d}, describing the initial state of a knowledge game, can be constructed from $preI_{\mathsf{d}}$ by executing a knowledge action type (as defined in chapter 4) in $preI_{\mathsf{d}}$. Using that, we can prove in an indirect way that I_{d} is uniquely described by kgames.

Proposition 7
Let $\mathsf{d} \in \mathbf{A}^{\mathbf{C}}$ be a deal of cards. Let M be a model of $\mathsf{prekgames}$ for parameter d. Then M is bisimilar to $preI_{\mathsf{d}}$.

Proof Let M be a model of $\mathsf{prekgames}$ for parameter d. Write $M = \langle W^M, (\sim_a^M)_{a\in\mathbf{A}}, V^M\rangle$. We remind the reader that $preI_\mathsf{d} = \langle D_{\sharp\mathsf{d}}, (\sim_a)_{a\in\mathbf{A}}, V\rangle$. First observe that, because $M \models \mathsf{deals}$, each world $w \in M$ has a valuation $V_w = V_d$ for some $d \in D_{\sharp\mathsf{d}}$. Define relation $\mathfrak{R} \subseteq (M \times preI_\mathsf{d})$ as follows:

$$\forall w \in M : \forall d \in D_{\sharp\mathsf{d}} : \mathfrak{R}(w, d) \Leftrightarrow V_w = V_d$$

We prove that $\mathfrak{R}$ is a bisimulation between M and $preI_\mathsf{d}$.

Forth: Let $w, w' \in M$. Let $d \in D_{\sharp\mathsf{d}}$. Suppose that $\mathfrak{R}(w, d)$ and that, for an arbitrary agent $a \in \mathbf{A}$, $w \sim_a^M w'$. From our observation on valuations in M, it follows that there is a deal $d' \in D_{\sharp\mathsf{d}}$ such that $V_{w'} = V_{d'}$. This deal d' is our required $\mathfrak{R}$-image in $D_{\sharp\mathsf{d}}$: because $\sim_a$ is universal on $D_{\sharp\mathsf{d}}$ it trivially holds that $d \sim_a d'$, and because $V_{w'} = V_{d'}$ we have that $\mathfrak{R}(w', d')$.

Back: Let $d, d' \in D_{\sharp\mathsf{d}}$. Let $w \in M$. Suppose that $\mathfrak{R}(w, d)$ and that, for an arbitrary agent $a \in \mathbf{A}$, $d \sim_a d'$.

Suppose there is no $w' \in M$ such that $V_{w'} = V_{d'}$. In other words: there is no $w' \in M$ such that $M, w' \models \delta_{d'}$. Then, in particular there is no $w' \in M$ such that $w \sim_a^M w'$ and $M, w' \models \delta_{d'}$, thus $M, w \not\models M_a\delta_{d'}$, thus $M, w \not\models \mathsf{dontknowany}$. Contradiction.

Therefore there is such a $w' \in M$, and, as we have shown, there is even a w' that is $\sim_a^M$-related to w. This world w' is our required $\mathfrak{R}$-original in M: we have that $w \sim_a^M w'$, and because $V_{w'} = V_{d'}$ we have that $\mathfrak{R}(w', d')$. ■

Turning the cards In chapter 4, see also [vD99], we present a dynamic epistemic language with dynamic modal operators $[\pi]$ for knowledge actions $\pi \in \mathsf{KA}$ and knowledge action types $\pi \in \mathsf{KT}$. KT action types and KA actions have an interpretation $[\![\cdot]\!]$, that is a relation between $S5$ models. If the relation is functional, we can use $[\![\cdot]\!]$ as a postfix unary operator. The action of all players looking at (turning) their cards is described by the knowledge action type $\mathsf{look}_\mathbf{A}$ (that will be defined in chapter 6) that has a functional interpretation. We now have that $preI_\mathsf{d}[\![\mathsf{look}_\mathbf{A}]\!] = I_\mathsf{d}$ (i.e.: the model resulting from executing $\mathsf{look}_\mathbf{A}$ in $preI_\mathsf{d}$ is identical, disregarding some trivial renaming of worlds, to I_d). This presents us with an 'indirect' proof of proposition 5, on page 35, that I_d is the unique model of kgames:

Indirect proof of proposition 5 Let M be a model of kgames. For every agent a, add access for a between all worlds in M that are not a-related. The resulting model M' is a model of $\mathsf{prekgames}$. It holds that $M'[\![\mathsf{look}_\mathbf{A}]\!] = M$. Because M' is bisimilar to $preI_\mathsf{d}$, and because bisimilarity is preserved under execution of action types with a functional interpretation (as shown in chapter 4), $M'[\![\mathsf{look}_\mathbf{A}]\!]$ is bisimilar to $preI_\mathsf{d}[\![\mathsf{look}_\mathbf{A}]\!]$. From that and $preI_\mathsf{d}[\![\mathsf{look}_\mathbf{A}]\!] = I_\mathsf{d}$ follows that M is bisimilar to I_d. ■

We suggest that this indirect proof method may be of more general interest, e.g. in cases where the bisimulation needed to establish a direct proof for the uniqueness of some model M is more complex than the bisimulation needed for a much simpler model from which M can be constructed by an action sequence. Note that the proof of proposition 7 is shorter than the proof of proposition 5, which illustrates our point.

3.4 Modal fixed points

We have described some finite $S5_n$ models and their states by way of proving that a suggested description (33, kgames, prekgames) indeed defines the bisimulation class of these models. These descriptions were 'merely' the most intelligible outcome of a gradual process of generalizing properties of players' knowledge. The description (or 'characteristic formula', or 'descriptive formula') of finite modal models and states can also be directly computed, by a fixed point construction. See [vB98], relating to [BM96]. We apply the construction in [vB98], chapter 5 ('Modal Fixed Points and Bisimulation'), to the initial state $(I_{\mathsf{d}}, \mathsf{d})$ of a knowledge game for deal d (and also, at the end of the section, to the pre-initial state $(preI_{\mathsf{d}}, \mathsf{d})$). A description operator E for modal models is defined. For the fixed-point description $E(I_{\mathsf{d}})$, we can compute a solution, because I_{d} is finite. We compute a specific solution $E^{\delta}(I_{\mathsf{d}})$ (where δ stands for 'choose atomic descriptions'). State $(I_{\mathsf{d}}, \mathsf{d})$ is then described by $\delta_{\mathsf{d}} \wedge C_{\mathbf{A}} E^{\delta}(I_{\mathsf{d}})$. We compare this to our own description. Now in more detail:

A fixed-point construction defines a template description for a given model M:

$$E(M) = \bigwedge_{w \in M} E(M, w) = \bigwedge_{w \in M} (p_w \rightarrow (\delta_w \wedge \bigwedge_{Rwv} \Diamond p_v \wedge \Box \bigvee_{Rwv} p_v))$$

Here δ_w is the atomic description of world w, and all p_w are fresh atoms. The idea behind the construction is the following: we give each world w in the model a fixed reference p_w, and then sum up for all worlds what is actually the case there, conditionalizing on this reference p_w: that w is atomically characterized by δ_w, that anything in a world accessible from w is possibly the case, and that anything shared by all worlds that are accessible from w must necessarily be the case.

If M is finite, we can replace the atomic variables p_w by a unique modal definition Δ_w of w in M. Initial (and pre-initial) states $(I_{\mathsf{d}}, \mathsf{d})$ of knowledge games are finite. In the initial model, different worlds are different deals of cards, i.e. they have a different *atomic* description. So $\delta_{d'}$ serves as a unique modal definition $\Delta_{d'}$ of worlds $d' \in I_{\mathsf{d}}$. We compute a solution of $E(I_{\mathsf{d}})$ by replacing all p_w by δ_w (i.e. $p_{d'}$ by $\delta_{d'}$). Use a multiagent epistemic language:

$E(I_{\mathsf{d}})[\bigwedge_{d' \in D_{\sharp\mathsf{d}}} (p_{d'} := \delta_{d'})] = E^\delta(I_{\mathsf{d}}) =$
$\bigwedge_{d' \in I_{\mathsf{d}}} \bigwedge_{a \in \mathbf{A}} (\delta_{d'} \rightarrow (\delta_{d'} \wedge \bigwedge_{d' \sim_a d''} M_a \delta_{d''} \wedge K_a \bigvee_{d' \sim_a d''} \delta_{d''}))$

As $\delta^a_{d'} \leftrightarrow \bigvee_{d' \sim_a d''} \delta_{d''}$, and as $\delta_{d'}$ in the consequent is superfluous, we get:

$$E^\delta(I_{\mathsf{d}}) = \bigwedge_{d' \in I_{\mathsf{d}}} \bigwedge_{a \in \mathbf{A}} (\delta_{d'} \rightarrow (\bigwedge_{d' \sim_a d''} M_a \delta_{d''} \wedge K_a \delta^a_{d'}))$$

The implicit quantification over all worlds of I_{d} in the formula $E^\delta(I_{\mathsf{d}})$, is made explicit with a common knowledge operator in the description (characteristic formula) of state $(I_{\mathsf{d}}, \mathsf{d})$:

$$\delta_{\mathsf{d}} \wedge C_{\mathbf{A}} E^\delta(I_{\mathsf{d}})$$

This is an alternative description to the one in section 3.2:

$$\delta_{\mathsf{d}} \wedge C_{\mathbf{A}} \mathsf{kgames}$$

Obviously, as they describe the same state, the two descriptions are equivalent.[4]

Example 21
The state (hexa, rwb) is described by $\delta_{rwb} \wedge C_{123} E^\delta(\mathsf{hexa})$. E.g. conjunct $E^\delta(\mathsf{hexa}, rwb)$ of $E^\delta(\mathsf{hexa})$ is equivalent to $\delta_{rwb} \rightarrow (M_1 \delta_{rbw} \wedge M_2 \delta_{bwr} \wedge M_3 \delta_{wrb} \wedge K_1 r_1 \wedge K_2 w_2 \wedge K_3 b_3)$. Note that in the consequent of that formula, we have deleted subformulas expressing reflexivity, such as $M_1 \delta_{rwb}$.

In section 3.1 we have shown that (hexa, rwb) is also described by $\delta_{rwb} \wedge C_{123} \mathsf{33}$. Note that in hexa any two worlds can linked by a $\{1,2,3\}$-path ($\bigcup_{i \in \{1,2,3\}} \sim_i$-path) of at most length 2. A still 'shorter' description is therefore $\delta_{rwb} \wedge E_{123} E_{123} E^\delta(\mathsf{hexa})$ – where E_{123} is the general knowledge operator.

Even though $\delta_{\mathsf{d}} \wedge C_{\mathbf{A}} E^\delta(I_{\mathsf{d}}) \Leftrightarrow \delta_{\mathsf{d}} \wedge C_{\mathbf{A}} \mathsf{kgames}$, it does not hold that that $E^\delta(I_{\mathsf{d}}) \Leftrightarrow \mathsf{kgames}$: note that $E^\delta(I_{\mathsf{d}})$ does not imply $\delta_{d'}$ for some $d' \in I_{\mathsf{d}}$. Therefore, the constituent $\mathsf{deals} = \bigvee_{d' \in D_{\sharp\mathsf{d}}} \delta_{d'}$ of kgames does not follow from it.

In the description $\delta_{\mathsf{d}} \wedge C_{\mathbf{A}} E^\delta(I_{\mathsf{d}})$, because δ_{d} holds in a state, and because $E^\delta(I_{\mathsf{d}})$, some $\delta_{d'}$ must hold in all worlds that are $\mathbf{A}$-accessible from the point of that state, so that deals holds there too; δ_{d} starts the ball rolling, so to speak.

Indeed, deals is *just* what we are short to establish a correspondence:

Proposition 8 (Comparison of descriptions)
In kgames, $E^\delta(I_{\mathsf{d}})$ is equally strong as $\mathsf{seedontknow}$:

$\bigwedge_{e \in I_{\mathsf{d}}} \bigwedge_{a \in \mathbf{A}} (\delta_e \rightarrow (\bigwedge_{e \sim_a d'} M_a \delta_{d'}) \wedge K_a \delta^a_e)$
$\Leftrightarrow_{\mathsf{kgames}}$
$\bigwedge_{e \in I_{\mathsf{d}}} \bigwedge_{a \in \mathbf{A}} (K_a \delta^a_e \leftrightarrow M_a \delta_e)$

[4] Incidentally, as we have shown in chapter 2 that any two worlds in the initial game state can be linked by a path of length 3, even 'shorter' descriptions would be – where $E_{\mathbf{A}}$ is the 'general knowledge' operator – $\delta_{\mathsf{d}} \wedge E^3_{\mathbf{A}} E^\delta(I_{\mathsf{d}})$ and $\delta_{\mathsf{d}} \wedge E^3_{\mathbf{A}} \mathsf{kgames}$.

Proof
$\Rightarrow$ Let $a \in \mathbf{A}, e \in D_{\sharp \mathsf{d}}$. First, we prove that $K_a \delta^a_e \rightarrow M_a \delta_e$: Assume $K_a \delta^a_e$. Then $(S5)$ δ^a_e. From that and deals follows $\bigvee_{d' \sim_a e} \delta_{d'}$. Let $d' \sim_a e$ be an arbitrary deal such that $\delta_{d'}$. From that and $E^\delta(I_\mathsf{d})$ follows $\bigwedge_{d'' \sim_a d'} M_a \delta_{d''}$. Because $e \sim_a d'$ it now follows that $M_a \delta_e$. Therefore $K_a \delta^a_e \rightarrow M_a \delta_e$.

Next, we prove that $M_a \delta_e \rightarrow K_a \delta^a_e$. Assume $M_a \delta_e$. Either δ_e or $\neg \delta_e$. If δ_e, then apply $E^\delta(I_\mathsf{d})$ and $K_a \delta^a_e$ follows. If $\neg \delta_e$, then from deals it follows that $\delta_{d''}$ for some $d'' \neq e$. Again with $E^\delta(I_\mathsf{d})$, follows $K_a \delta^a_{d''}$. We can have either $d'' \sim_a e$ or $d'' \not\sim_a e$. If $d'' \sim_a e$ then $\delta^a_{d''} \leftrightarrow \delta^a_e$ and from that and $K_a \delta^a_{d''}$ follows $K_a \delta^a_e$. If $d'' \not\sim_a e$, then:

$\delta^a_{d''}$
$\Leftrightarrow$
$\bigvee_{d^* \sim_a d''} \delta_{d^*}$
$\Leftrightarrow$ because deals is an exclusive disjunction
$\neg \bigvee_{d^* \not\sim_a d''} \delta_{d^*}$
$\Leftrightarrow$
$\bigwedge_{d^* \not\sim_a d''} \neg \delta_{d^*}$
$\Rightarrow$
$\neg \delta_e$

Therefore, using that deals is commonly known, from $K_a \delta^a_{d''}$ follows $K_a \neg \delta_e$, which is equivalent to $\neg M_a \delta_e$: contradiction with our assumption $M_a \delta_e$. Therefore $M_a \delta_e \rightarrow K_a \delta^a_e$.

$\Leftarrow$ Suppose δ_e. Then $M_a \delta_e$. From that and seedontknow follows $K_a \delta^a_e$. From that, and because for all $d' \sim_a e$: $\delta^a_{d'} \leftrightarrow \delta^a_e$, follows that for all $d' \sim_a e$: $K_a \delta^a_{d'}$. Using seedontknow for all $d' \sim_a e$, we get: for all $d' \sim_a e$: $M_a \delta_{d'}$. Thus $\bigwedge_{d' \sim_a e} M_a \delta_{d'}$. ∎

Prekgames Similarly as for I_d, we can compute the solution $E^\delta(preI_\mathsf{d})$ for a fixed point description of the pre-initial model $preI_\mathsf{d}$. The solution $E^\delta(preI_\mathsf{d})$ is equivalent to dontknowany.

3.5 Further observations

Describing other game states We have described only the initial state of a knowledge game, and the pre-initial state. Similarly, we may describe other game states, resulting from the execution of game actions in the initial game state. We can then expect that subgroup common knowledge operators will occur in these descriptions. E.g. in hexa, $C_{12} r_1$ is a postcondition of the show action where 1 shows red to 2.

Knowledge revision Computing the description of each resulting game state all over again after execution of every single game action, seems a rather roundabout way to proceed.[5] A process of knowledge revision, where we directly 'update' or revise the theories describing the models underlying these states instead, is more direct and therefore preferable. First, an example of what we mean by knowledge revision:

Example 22 (Revising ignorance)
Formula

$$\mathsf{dontknowany} = \bigwedge_{a \in \mathbf{A}} \bigwedge_{d' \in D_{\sharp \mathsf{d}}} M_a \delta_{d'}$$

describes that all deals are relevant when the cards have been dealt but not turned. In the transition from $preI_{\mathsf{d}}$ to I_{d}, players turn and look at their cards. As a consequence, we have to revise the ignorance as expressed in $\mathsf{dontknowany}$: some deals are no longer conceivable to some players. We can revise ignorance by conditionalizing on conceivable deals: after execution of the action $\mathsf{look}_{\mathbf{A}}$, a deal is conceivable, and only conceivable, if your known 'local state' corresponds to it:

$$\mathsf{seedontknow} = \bigwedge_{a \in \mathbf{A}} \bigwedge_{d' \in D_{\sharp \mathsf{d}}} (K_a \delta^a_{d'} \leftrightarrow M_a \delta_{d'})$$

This revision should be a function of the action $\mathsf{look}_{\mathbf{A}}$. We imagine $\mathsf{seedontknow}$ being computed from something like $\mathsf{dontknowany}[\mathsf{look}_{\mathbf{A}}]$, where $\varphi[\alpha]$ stands for: 'the revision of φ as a consequence of the execution of action α'. The process of revising the theory $\mathsf{prekgames}$ would then be:

$$\begin{aligned}
\mathsf{prekgames}_{\mathsf{d}}[\mathsf{look}_{\mathbf{A}}] &= (\mathsf{deals} \wedge \mathsf{dontknowany})[\mathsf{look}_{\mathbf{A}}] \\
&= \mathsf{deals}[\mathsf{look}_{\mathbf{A}}] \wedge \mathsf{dontknowany}[\mathsf{look}_{\mathbf{A}}] \\
&= \mathsf{deals} \wedge \mathsf{seedontknow}
\end{aligned}$$

This kind of knowledge revision is treated in [vB00b], where the process is called *syntactic relativization.* A procedure is given for the special case of actions that are public announcements. The action $\mathsf{look}_{\mathbf{A}}$ is not a public announcement. We have not pursued this fascinating topic further.

Describing other multiagent states The topic of characterizing multiagent $S5$ models has also been fruitfully pursued for *hypercubes*, that represent interpreted systems. See [Lom99, LvdMR00, LR98a]. This will be summarily discussed in chapter 7.

[5] The indirect proof of proposition 5 illustrates that point: even though a unique model M results from executing $\mathsf{look}_{\mathbf{A}}$ in M', we have to show that this is a model of kgames. In the proof we guaranteed that by *assuming* that M is a model of kgames and by *constructing* M' from M.

3.6 Conclusion

We have described two different states for card games. We showed that the model hexa for the initial knowledge game state of three players each holding a card is described by the theory $\mathsf{33}$. Given a deal $\mathsf{d} \in \mathbf{A}^{\mathbf{C}}$, the model I_{d} underlying an initial game state for $|\mathbf{C}|$ cards over $|\mathbf{A}|$ players, is described by the theory kgames. In particular we have described various equally strong formulas that express that a player knows the cards that he holds, and we have presented three different expressions of ignorance. Prior to the state where players have picked up their cards from the table, is the state $(preI_{\mathsf{d}}, \mathsf{d})$ where cards have been dealt over players but where they haven't yet turned their cards. The model $preI_{\mathsf{d}}$ is described by the theory $\mathsf{prekgames}$. Our results correspond to those of fixed point computations of the description of modal models. Descriptions of other game states may be fruitfully pursued from a viewpoint of knowledge revision.

Chapter 4

Update by local interpretation

The area of dynamic epistemics, how to update models for reasoning about knowledge, has come to the full attention of the research community by the treatment of public announcements in the famous 'Muddy Children Problem' [FHMV95]. An integrated approach including announcements to subgroups has been put forward in [GG97]. Gerbrandy's thesis, [Ger99], presents this dynamic epistemics in more generality. Gerbrandy's approach is based on non-well-founded set theory, a non-standard semantics. Based on a standard semantics, [BMS99] treats announcements. This is currently being extended to an entire framework for epistemic dynamics [Bal99]. Our research should probably be seen as a special case of the more general framework as presented by Gerbrandy and under development by Baltag. Part of its interest lies in the detailed description of new sorts of epistemic action, namely actions in games. We restrict ourselves to $S5$ models and states. We base ourselves on standard Kripke semantics. Our contribution is to interpret programs for updates by a process of – what we call – *local interpretation*, by which we can elegantly describe and interpret combined updates for different subgroups. Apart from the usual programming constructs: test, sequential execution, and nondeterministic choice [Har84, Gol92], we introduce as well: *learning* and *local choice.* Learning is related to truthful (factive) updating in [Ger99]. Local choice is needed to describe actions where different subgroups learn different things. We start with some examples to illustrate the need for such an operation.

The following actions can be executed in the $S5$ state (hexa, rwb), that models three players (1, 2, 3) each holding one card (red, white, blue), see chapter 2:

Example 23 (tell)
Player 1 puts the red card (face up) on the table.

Example 24 (show)
Player 1 shows (only) player 2 the red card. Player 3 cannot see the face of the shown card, but notices that a card is being shown.

Example 25 (whisper)
Player 2 asks player 1 to tell him a card that he (1) doesn't have. Player 1 whispers in 2's ear "I don't have white". Player 3 notices that the question is

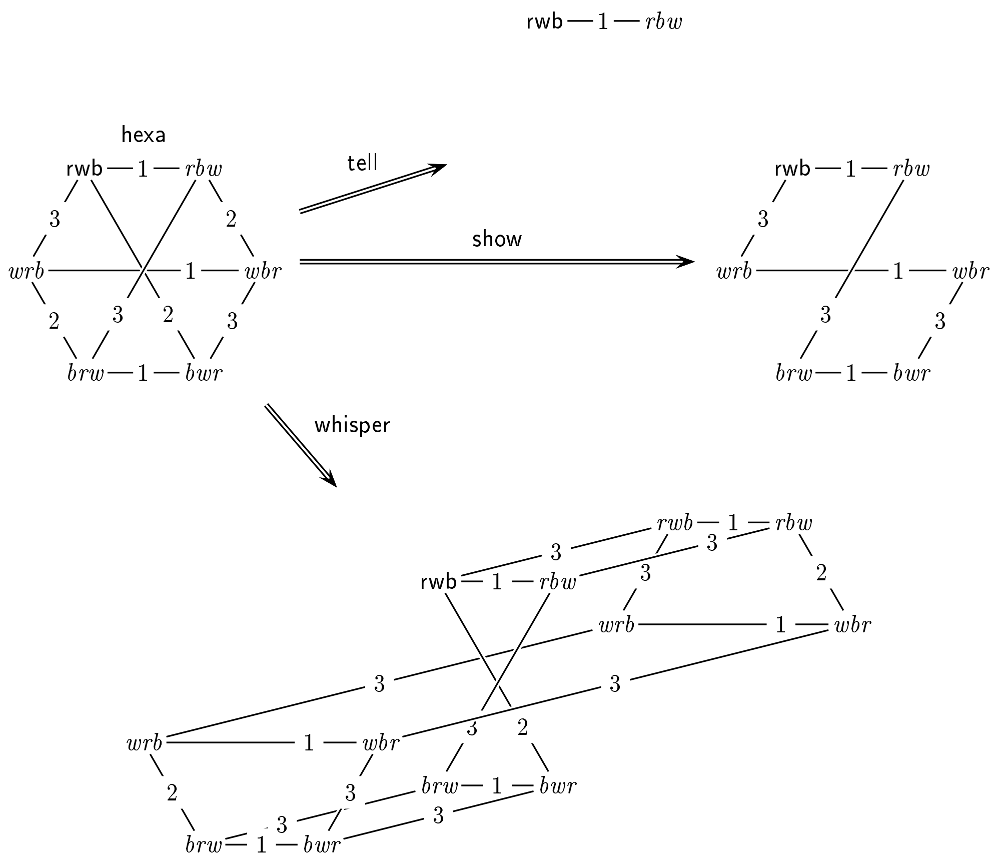


Figure 4.1: The results from three actions in a state where 1 holds red, 2 holds white and 3 holds blue. The points of the states are in sans serif. Worlds are named by the deals that (atomically) characterize them. Assume reflexivity and transitivity of access. More explanations are given throughout the text.

answered, but cannot hear the answer.

We assume that only the truth is told. Therefore tell has the same effect as when 1 tells the others that he has red. In show and whisper, we assume that it is publicly known what 3 can and cannot see or hear. Figure 4.1 pictures the states that result from updating the current game state (hexa, rwb) with the information contained in the three actions. In tell it suffices to eliminate some worlds: after 1's action, the four deals of cards where 1 does not hold red are eliminated. It is publicly known that they are no longer accessible. This update is a public announcement. In show we cannot eliminate any world. After this action, e.g., 1 can imagine that 3 can imagine that 1 has shown red, but also that 1 has shown white, or blue. However, some *links* between worlds have now been severed: whatever the actual deal of cards, 2 cannot imagine any alternatives after execution of show. In whisper player 1 can *choose* whether to say "not white" or "not blue", and the resulting game state has twice as many worlds as the current state, because for each deal of cards this choice can be imagined to have been made.

We can paraphrase some more of the structure of the actions. In tell, all three players *learn* that player 1 holds the red card. In show, 1 and 2 *learn* that 1 holds red, whereas the group consisting of 1, 2 and 3 learns that 1 and 2 learn which card 1 holds, or, in other words: that either 1 and 2 learn that 1 holds red, or that 1 and 2 learn that 1 holds white, or that 1 and 2 learn that 1 holds blue. The choice made by subgroup $\{1, 2\}$ from the three alternatives is *local*, i.e. known to them only, because it is hidden from player 3. *Local interpretation* of 1 and 2 learning that 1 holds red, is indeed made without any reference to 3. Here, 'local' means 'not public' – 'for a subgroup of the public' – and does not mean 'private' – 'for one agent only'. The 'action' of player 1 showing player 2 *his card* is called the *type* of the action show where player 1 shows player 2 *the red card*. There are two other actions of that type: showing white and showing blue. Similarly, there are three actions of the type of whisper, corresponding to the answers "not red", "not white", and "not blue" to the question.

In section 4.1 we define the logical language $\mathcal{L}_n^\Box$, the knowledge action types KT and the knowledge actions KA. In section 4.2 we define local interpretation. In section 4.3 we present some theoretical results. In section 4.4 we discuss possible extensions of $\mathcal{L}_n^\Box$.

4.1 Knowledge actions

To a standard multiagent epistemic language $\mathcal{L}_n$ (as in [MvdH95, FHMV95], or see appendix A) we add dynamic model operators for programs that describe actions.

Definition 25 (Language of dynamic epistemic logic)
Given are a set of atomic propositions $\mathbf{P}$ and a set of agents $\mathbf{A}$, where $|\mathbf{A}| = n$. $\mathcal{L}_n^\Box$ is the smallest set closed under:

$$\begin{array}{rll}
p \in \mathcal{L}_n^\Box & \text{if} & p \in \mathbf{P} \\
\neg\varphi \in \mathcal{L}_n^\Box & \text{if} & \varphi \in \mathcal{L}_n^\Box \\
\varphi \wedge \psi \in \mathcal{L}_n^\Box & \text{if} & \varphi, \psi \in \mathcal{L}_n^\Box \\
K_a\varphi \in \mathcal{L}_n^\Box & \text{if} & a \in \mathbf{A} \text{ and } \varphi \in \mathcal{L}_n^\Box \\
C_B\varphi \in \mathcal{L}_n^\Box & \text{if} & B \subseteq \mathbf{A} \text{ and } \varphi \in \mathcal{L}_n^\Box \\
[\pi]\varphi \in \mathcal{L}_n^\Box & \text{if} & \pi \in \mathsf{KT} \cup \mathsf{KA} \text{ and } \varphi \in \mathcal{L}_n^\Box
\end{array}$$

KT is the set of *knowledge action types* (for $\mathbf{P}$ and $\mathbf{A}$). KA is the set of *knowledge actions* (for $\mathbf{P}$ and $\mathbf{A}$). 'Program' is the generic term that we use for both types and actions. The parameter set of agents $\mathbf{A}$ is called the *public*. We introduce the usual abbreviations (let $p \in \mathbf{P}$): $\varphi \vee \psi := \neg(\neg\varphi \wedge \neg\psi)$, $\varphi \to \psi := \neg\varphi \vee \psi$,

$\varphi \leftrightarrow \psi := (\varphi \to \psi) \wedge (\psi \to \varphi)$, $E_B\varphi := \bigwedge_{a\in B} K_a\varphi$, $\top := p \vee \neg p$, $\bot := p \wedge \neg p$. Next, we define the class of knowledge action types:

Definition 26 (Knowledge action types – KT)
Given a set of agents $\mathbf{A}$ and a set of atoms $\mathbf{P}$, KT is the smallest set closed under:

$$\begin{array}{rll} ?\varphi \in \mathsf{KT} & \text{if} & \varphi \in \mathcal{L}_n^\Box \\ L_B\tau \in \mathsf{KT} & \text{if} & \tau \in \mathsf{KT} \text{ and } la(\tau) \subseteq B \\ \tau \ ; \ \tau' \in \mathsf{KT} & \text{if} & \tau, \tau' \in \mathsf{KT} \\ \tau \cup \tau' \in \mathsf{KT} & \text{if} & \tau, \tau' \in \mathsf{KT} \end{array}$$

L_B is the 'learn' operator. $L_B\tau$ stands for 'group B learn that τ'. Instead of $L_{\{1,2,\dots,i\}}$ write $L_{12\dots i}$. In $la(\tau) \subseteq B$, operator la stands for 'learning agents'. These are the agents occurring in the learning operators of τ. Without this constraint, it would be possible that an agent a learns something about an agent b, without b being aware of that: after that, b may have false beliefs about a's knowledge, so that the resulting state is not $S5$. We restrict ourselves to $S5$.

Definition 27 (Learning agents)
The (set of) learning agents of a knowledge action type is defined by inductive cases: $la(?\varphi) = \emptyset$, $la(L_B\tau) = B \cup la(\tau)$, $la(\tau \ ; \ \tau') = la(\tau) \cup la(\tau')$, $la(\tau \cup \tau') = la(\tau) \cup la(\tau')$.

Naturally, $\mathcal{L}_n^\Box$, KT and la are supposed to be simultaneously defined. We now give some examples of action types, related to the model hexa:

Example 26 (Knowledge action type for tell)
Player 1 puts the red card on the table:

$$L_{123}?r_1$$

Example 27 (Knowledge action type for show)
Player 1 shows (only) player 2 his card:

$$L_{123}(L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1)$$

To understand the description, greater precision is needed: 'player 1 shows (only) player 2 his card' is the same as 'players 1, 2 and 3 learn that player 1 shows player 2 his card', which is the same as 'players 1, 2 and 3 learn that 1 shows the red card to 2, or the white card, or the blue card' which is the same as 'players 1, 2 and 3 learn (that 1 and 2 learn that 1 holds red, or that 1 and 2 learn that 1 holds white, or that 1 and 2 learn that 1 holds blue)'. The action type is nondeterministic. Note that it does *not* describe the *action* show where the *red* card was shown. It is the *type* of that action. The action type can be executed in a game state where 1 holds blue, the action show can *not* be executed in that game state. Assume associativity for the operator $\cup$ (see proposition 10, later).

Example 28 (Knowledge action type for whisper)
Player 1 whispers in 2's ear a card that he (1) doesn't have:

$$L_{123}(L_{12}?\neg r_1 \cup L_{12}?\neg w_1 \cup L_{12}?\neg b_1)$$

The three options are *not* having a card, instead of having a card. This *action type* does not describe the *action* whisper, that is just one of three actions of that type.

Knowledge actions We continue by defining *knowledge actions*. We extend our language with the operation of 'local choice', to make that possible. In 'nondeterministic choice' choice means 'making a choice possible', in local choice it means 'actually choosing'. We can only 'actually choose' after a choice has been made possible. Because actual choice is local, a new program is created.

A knowledge action can be formed from an action type τ by a mapping operation $!_I\tau$, where $!_I$ determines *local choice* in τ. Index I is a *bundle*: (an abstraction of) a subtree in the structural tree of action type τ. The bundle determines *where* the local choice is made, the choice itself is *indicated* by an exclamation mark '!'. In that way we can describe the action show as the *knowledge action* $L_{123}(!L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1)$, given the type $L_{123}(L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1)$ ('showing your card') of that action.

Definition 28 (Bundle)
The *labelled structure* $ls(\tau)$ of a knowledge action type τ is inductively defined as follows: $ls(?\varphi) = ()$, $ls(L_B\tau) = (ls(\tau))$, $ls(\tau \cup \tau') = x(ls(\tau), ls(\tau'))$ for some fresh variable x, $ls(\tau \ ; \ \tau') = (ls(\tau), ls(\tau'))$. Let $fv(ls(\tau))$ be the set of variables in $ls(\tau)$. Let val be a valuation in $\{0,1\}^{fv(ls(\tau))}$. Extend val to apply to $ls(\tau)$ in the obvious way. Define an equivalence relation $=_{bu}$ on the set $val(ls(\tau))$ of evaluated labelled structures by inductive cases: $=_{bu}$ is '=' except for $0(I,J) =_{bu} 0(K,L) \Leftrightarrow I =_{bu} K$, and $1(I,J) =_{bu} 1(K,L) \Leftrightarrow J =_{bu} L$. The set $bu(\tau)$ of *bundles* of τ is defined as:

$$bu(\tau) := \{[I]_{=_{bu}} \mid I \in val(ls(\tau)), val \in \{0,1\}^{fv(ls(\tau))}\}$$

Now choose representants of the $=_{bu}$ equivalence classes, and write, par abus de langage, $I = J$ for $[I]_{=_{bu}} = [J]_{=_{bu}}$.

A bundle defines a rooted subtree of the labelled structure of a knowledge action type.[1] Value 0 selects the 'left' arc of the current subtree, whereas 1 selects the 'right' arc. The subtree that is generated by the node that is reached by the arc that is not chosen, is pruned. The effect of $=_{bu}$ is that all choices are identified that are made further down in a pruned subtree.

[1]It can be seen as 'a couple of branches' (a branch is a path from a root to the leaf of the tree), hence its name. We copy the similar use of the term 'bundle' in modal sequent calculus.

Given a knowledge action $\tau \in \mathsf{KT}$ and a bundle $I \in bu(\tau)$, $!_I\tau$ defines a *knowledge action*: a specific action of type τ. Every bundle corresponds to an action. The set $\{!_I\tau \mid I \in bu(\tau)\}$ is the set of actions of type τ. KA is the class of all syntactic objects thus formed. We write α for an arbitrary knowledge action in KA.

Definition 29 (Knowledge actions – KA)
Let $\mathbf{A}$ be a set of agents, let $\mathbf{P}$ be a set of atoms.

$$\mathsf{KA} = \{!_I\tau \mid \tau \in \mathsf{KT} \text{ and } I \in bu(\tau)\}$$

The notation with bundles prefixed to action types is a bit hard to read. We therefore define a notational equivalent for knowledge actions:

Definition 30 (Notational equivalent for knowledge actions)
By inductive cases:

$$\begin{array}{rcl} !_{()}(?\varphi) & := & !?\varphi \\ !_{(I)}(L_B\tau) & := & !L_B\ !_I\tau \\ !_{0(I,J)}(\tau \cup \tau') & := & !_I\tau \cup \tau' \\ !_{1(I,J)}(\tau \cup \tau') & := & \tau \cup\ !_J\tau' \\ !_{(I,J)}(\tau\ ;\ \tau') & := & !_I\tau\ ;\ !_J\tau' \end{array}$$

The notational equivalent is not the primitive notation, because the interpretation of an action (see definition 34, later) would then not be compositional.

If there is nothing to choose, we can delete exclamation marks in the notational equivalent. Their only meaningful position is in constructs $\pi' \cup\ !\pi$ and $!\pi \cup \pi'$: if $\pi = L_B\pi''$; this means that only the agents in B know that they have selected π'' for execution and not the alternatives expressed in π'. See also the examples, below. We change their usual order:

Example 29 (Knowledge action for show)
The type of the action show is $L_{123}(L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1)$. We must be more precise and choose it to be: $L_{123}((L_{12}?r_1 \cup L_{12}?w_1) \cup L_{12}?b_1)$. The bundle in the labelled structure of this action type that corresponds to choosing red, is:

$$(0(0((()),(())),(())))$$

The knowledge action

$$!_{(0(0((()),(())),(())))}L_{123}((L_{12}?r_1 \cup L_{12}?w_1) \cup L_{12}?b_1)$$

is written in the notational equivalent as

$$!L_{123}((!L_{12}!?r_1 \cup L_{12}?w_1) \cup L_{12}?b_1)$$

and can be simplified, also assuming associativity again, to

$$L_{123}(!L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1)$$

We now have described the action 'player 1 shows (only) player 2 the red card'. The other two actions of this type are $L_{123}(L_{12}?r_1\cup\ !L_{12}?w_1 \cup L_{12}?b_1)$ (1 shows white to 2) and $L_{123}(L_{12}?r_1 \cup L_{12}?w_1\cup\ !L_{12}?b_1)$ (1 shows blue to 2). Note that the bundle for the choice 'blue' corresponds to an equivalence class consisting of two valued labelled structures: $(1(0(((()),(())),(()))))$ is the same ('$=_{bu}$') bundle as $(1(1(((()),(())),(()))))$; once we have chosen the *right* subtree, we can ignore choices further down in the pruned *left* subtree.

Example 30 (Knowledge action for whisper)
It will be clear that the KA action describing whisper is

$$L_{123}(L_{12}?\neg r_1\cup\ !L_{12}?\neg w_1 \cup L_{12}?\neg b_1)$$

Example 31 (Knowledge action for tell)
There is no choice involved. The only bundle of tell is $(())$. $L_{123}?r_1$, strictly $!L_{123}!?r_1$, is the only knowledge action of type $L_{123}?r_1$.

The class of actions KA is *not* closed under action type constructing operations. E.g., $L_{123}(!L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1) \cup L_{123}(L_{12}?r_1\cup\ !L_{12}?w_1 \cup L_{12}?b_1)$ is *neither* an action, *nor* an action type. In particular, it is not an action of the same type as $L_{123}(L_{12}?r_1 \cup L_{12}?w_1)$, showing either red or white.

4.2 Local interpretation

One is used to the set of agents of a model as a background parameter that is fixed throughout one's manipulations with a logic, and that therefore does not need a *name*. However, because the local interpretation of an action type may relate models and states for different sets of agents, we have to be very explicit about it. The 'group' $gr(Y)$ of a semantic object Y (model, state, frame) is the set of agents for which access is defined, i.e. the set of agents 'occurring in it'. Similarly, the group $gr(X)$ of an expression X (program, formula) is the set of agents occurring in modal operators K_a, C_B and $[\pi]$ in that expression.

Definition 31 (Group)
Let M be an $S5$ model. The group of $M = \langle W, (\sim_a)_{a\in A}, V\rangle$ is A: $gr(M) = A$. Similarly for states and frames. The group of a syntactic expression is defined by inductive cases for formulas and for knowledge action types, and directly for knowledge actions: $gr(p) = \emptyset$, $gr(\neg\varphi) = gr(\varphi)$, $gr(\varphi \wedge \psi) = gr(\varphi) \cup gr(\psi)$, $gr(K_a\varphi) = gr(\varphi)\cup\{a\}$, $gr(C_B\varphi) = gr(\varphi)\cup B$, $gr([\tau]\varphi) = gr(\varphi)\cup gr(\tau)$; $gr(?\varphi) = gr(\varphi)$, $gr(L_B\tau) = B\cup gr(\tau)$, $gr(\tau\ ;\ \tau') = gr(\tau)\cup gr(\tau')$, $gr(\tau\cup\tau') = gr(\tau)\cup gr(\tau')$; $gr(!_I\tau) = gr(\tau)$.

If $gr(M) = A$, we say that M is an A *model*, or that M is a *model for group* A. Similarly, if $gr(\pi) = A$, we say that π is an A *program*. Similarly for other semantic and syntactic objects.

Definition 32 (Semantics of $\mathcal{L}_n^\Box$)
Given are a set of (n) agents $\mathbf{A}$ and a set of atoms $\mathbf{P}$. Let $M = \langle W, (\sim_a)_{a\in\mathbf{A}}, V\rangle$ be an $S5_n$ model. Let $w \in M$, let $\psi \in \mathcal{L}_n^\Box$. The interpretation of ψ in M, w is defined by inductive cases; let $p \in \mathbf{P}$, $a \in \mathbf{A}$, $B \subseteq \mathbf{A}$, $\pi \in \mathsf{KT} \cup \mathsf{KA}$.

$$
\begin{array}{lll}
M, w \models p & \text{iff} & V_w(p) = 1 \\
M, w \models \neg\varphi & \text{iff} & M, w \not\models \varphi \\
M, w \models \varphi \wedge \psi & \text{iff} & M, w \models \varphi \text{ and } M, w \models \psi \\
M, w \models K_a\varphi & \text{iff} & \forall w' \in M : w' \sim_a w \Rightarrow M, w' \models \varphi \\
M, w \models C_B\varphi & \text{iff} & \forall w' \in M : w' \sim_B w \Rightarrow M, w' \models \varphi \\
M, w \models [\pi]\varphi & \text{iff} & \forall M' \in S5_{\leq n} : \forall w' \in M' : (M, w)[\![\pi]\!](M', w') \Rightarrow M', w' \models \varphi
\end{array}
$$

A program π can be either an action type $\tau \in \mathsf{KT}$ or an action $\alpha \in \mathsf{KA}$. For the local interpretation $[\![\tau]\!]$ of action types, see definition 33. For the local interpretation $[\![\alpha]\!]$ of actions, see definition 34. Both for actions and types, the interpretation $[\![\pi]\!]$ is a state transformer and can be seen as a binary relation between $S5$ states. If $(M, w)[\![\pi]\!](M', w')$, the group (gr) of M' is contained in the group of M. This will be explained in definition 33. Therefore, we write $M' \in S5_{\leq n}$ in that clause of the interpretation: M' may be an $S5_m$ model for any $m \leq n$.

Local interpretation of a knowledge action type To interpret a B action type τ in an A model 'locally', you 'forget' about the agents in $A \setminus B$ (contrary to [Ger99], where it is assumed that agents not in B have learnt *nothing* from τ). How τ affects these other agents may be interpreted 'later', namely when τ is a subprogram of another action type.

The local interpretation of an action type τ on an $S5$ model M is a relation $[\![\tau]\!]$ between $S5$ models and between their worlds (states). We 'overload' the notation $[\![\cdot]\!]$ by using it both between models and between between states.

Notation In the definition below, we use the following notations for infix binary relations R such as $[\![\cdot]\!]$: $[aR] := \{b \mid aRb\}$, as in clause 33.b.2, and $aRbRc :=$ (aRb and bRc), as in clause 33.c.3. In the clauses under (e), write $M'' = \langle W''$, $(\sim''_a)_{a\in gr(M'')}$, $V''\rangle$ for an arbitrary $M'' \in [M[\![\tau]\!]]$; in 33.e.2 and 33.e.4 we thus refer to access and valuation in M''. In 33.e.3 we use the following shorthand, given a context of two models: $\rightarrow_\tau^{-1}(w) := \iota v.(M, v)[\![\tau]\!](M'', w)$. In other words, $\rightarrow_\tau^{-1}(w)$ is the unique world in M that is the $\rightarrow$-origin of world w in M''. (As proven in proposition 9, on page 65.)

Definition 33 (Local interpretation of a knowledge action type) Given are a set of agents $\mathbf{A}$ and a set of atoms $\mathbf{P}$. Let (M, w) be an $S5_n$ state, where $M = \langle W, (\sim_a)_{a \in \mathbf{A}}, V \rangle$. Let (M', w') be an $S5_{\leq n}$ state. Let $\tau \in \mathsf{KT}$. The local interpretation $[\![\cdot]\!]$ of τ in M is simultaneously defined as a relation between $S5$ models and between their worlds, by inductive cases:

$$
\begin{array}{lrcl}
33.a.1 & (M,w)[\![\tau]\!](M',w') & \text{iff} & M[\![\tau]\!]M' \text{ and } w \rightarrow_\tau w' \\
\\
33.b.1 & M[\![?\varphi]\!]M' & \text{iff} & M' = \langle W', \emptyset, V \restriction W' \rangle \\
 & & & \quad \text{where } W' = \{w \in W \mid M, w \models \varphi\} \\
33.b.2 & M[\![L_B \tau'']\!]M' & \text{iff} & M' = \bigoplus_B [M[\![\tau'']\!]] \text{ as defined below} \\
33.b.3 & M[\![\tau'' \; ; \; \tau']\!]M' & \text{iff} & \exists M'' : M[\![\tau'']\!]M''[\![\tau']\!]M' \\
33.b.4 & M[\![\tau'' \cup \tau']\!]M' & \text{iff} & M[\![\tau'']\!]M' \text{ or } M[\![\tau']\!]M' \\
\\
33.c.1 & w \rightarrow_{?\varphi} w' & \text{iff} & w = w' \\
33.c.2 & w \rightarrow_{L_B \tau''} w' & \text{iff} & \exists M'' \in [M[\![\tau'']\!]] : \exists w'' \in M'' : \\
 & & & w' = (M'', w'') \text{ and } w \rightarrow_{\tau''} w'' \\
33.c.3 & w \rightarrow_{\tau'';\tau'} w' & \text{iff} & \exists M'' : M[\![\tau'']\!]M''[\![\tau']\!]M', \exists w'' \in M'' : \\
 & & & w \rightarrow_{\tau''} w'' \rightarrow_{\tau'} w' \\
33.c.4 & w \rightarrow_{\tau'' \cup \tau'} w' & \text{iff} & w \rightarrow_{\tau''} w' \text{ or } w \rightarrow_{\tau'} w' \\
\\
33.d.1 & \bigoplus_B [M[\![\tau'']\!]] & = & \langle W', (\sim'_a)_{a \in B}, V' \rangle, \text{ where:} \\
\\
33.e.1 & W' & = & \{(M'', w'') \mid w'' \in M'' \in [M[\![\tau'']\!]]\} \\
33.e.2 & & & \forall a \in B : \forall (M'', v), (M'', v') \in W' : \\
 & (M'', v) \sim'_a (M'', v') & \text{iff} & v \sim''_a v' \\
33.e.3 & & & \forall a \in B : \forall (M'', v), (M^*, v') \in W' : \\
 & & & a \notin gr(M'') \cup gr(M^*) \Rightarrow \\
 & (M'', v) \sim'_a (M^*, v') & \text{iff} & \rightarrow^{-1}_{\tau''}(v) \sim_a \rightarrow^{-1}_{\tau''}(v') \\
33.e.4 & \forall (M'', v) \in W' : & & V'_{(M'',v)} = V''_v
\end{array}
$$

Ad (33.a) To interpret an action type τ in a state (M, w), one has to interpret τ in the model M *and* one has to determine the images of w under that interpretation. The operator $\rightarrow_\tau$ determines these images.

Ad (33.b) In part (b) of definition 33, the interpretation of action types on models is defined. In clause $33.b.1$, the group of the resulting model M' must be empty: nobody is aware of a test being executed. Also in $33.b.1$, it is implicit that $W' \neq \emptyset$: an $S5$ model has a nonempty domain. In clause $33.b.2$, in the resulting model M' all in B are aware of (know) the alternatives resulting from interpreting τ, but they cannot (necessarily) distinguish between them. Further, note that in $33.b$ the group of a resulting model M' is always (equal to or strictly) contained in the group of M, and that the group of a program to be interpreted is always contained in the group of M. The last is implicit in the assumptions of definition 33: they are given relative to $\mathbf{P}$ and $\mathbf{A}$, so that τ can be at most an $\mathbf{A}$ knowledge type, from which follows that $gr(\tau) \subseteq gr(M)$. This implies, in clause 33.b.3, that the interpretation of a sequence τ'' ; τ' is undefined if $gr(\tau'') \subset gr(\tau')$.

Ad (33.c) In part (c) of definition 33, the interpretation of action types on states is defined. Clause 33.c.2 states that world w' is an image of world w for type $L_B\tau$, if w' is a state (M'', w'') that results from interpreting τ on (M, w).

Ad (33.d) and (33.e) In parts (d) and (e) of definition 33, we define the model $\bigoplus_B[M[\![\tau]\!]]$, the result of interpreting an action type $L_B\tau$. The model $\bigoplus_B[M[\![\tau]\!]]$ is the direct sum $\bigoplus[M[\![\tau]\!]]$ plus for any two models in that set, access added for agents $b \in B$ that do not occur in these models. (So that, indeed, B is just the parameter that we need.) Note that clause 33.e.2 has an implicit condition $a \in gr(M'')$. Another issue: without the constraint $la(\tau) \subseteq B$ for types $L_B\tau$ in definition 26 of KT, the model $\bigoplus_B[M[\![\tau]\!]]$ could still be constructed, but would be incorrect; the knowledge resulting from our intuitive interpretation of such an 'action' does not correspond to the knowledge encoded in the constructed model.

Direct sum To construct the direct sum $\bigoplus\mathcal{M}$ of a set $\mathcal{M}$ of models (see appendix A), we cannot just take the union of their domains: if we do that, we may incorrectly identify worlds with the same name. We have to pair a world $w \in M_i \in \mathcal{M}$ to some index i in order to determine the model M_i it originates from. The pair (w, i) is then a world in the sum model. In our case, it suffices to take the model *itself* as index.[2] Thus we get: (w, M_i). Because we may as well write (M_i, w), this amounts to taking *states* for models from $\mathcal{M}$ as *worlds* in the direct sum $\bigoplus\mathcal{M}$.

Unless confusion results, in the examples we *still* simply take the union of the domains of models in $\bigoplus_B[M[\![\tau]\!]]$. In the examples we name worlds by card deals, that correspond to the atomic descriptions of those worlds. Even when different worlds have the same name, we can distinguish them by the different access they have to other worlds.

Notation If there is a unique model M' such that $M[\![\tau]\!]M'$, we may write $M[\![\tau]\!]$ for that model. If there is a unique state (M', w') such that $(M, w)[\![\tau]\!](M', w')$, we may write $(M, w)[\![\tau]\!]$ for that state.

We illustrate definition 33 by computing the interpretation of the knowledge action types of tell, show and $\mathsf{whisper}$ on the model hexa. For more examples of

[2] In general, this does not do, because two models may have the same name. In our case, the *set* $[M[\![\tau]\!]]$ does not contain multiple occurrences of models, see also clause 33.b.4: they are not being produced anyway.

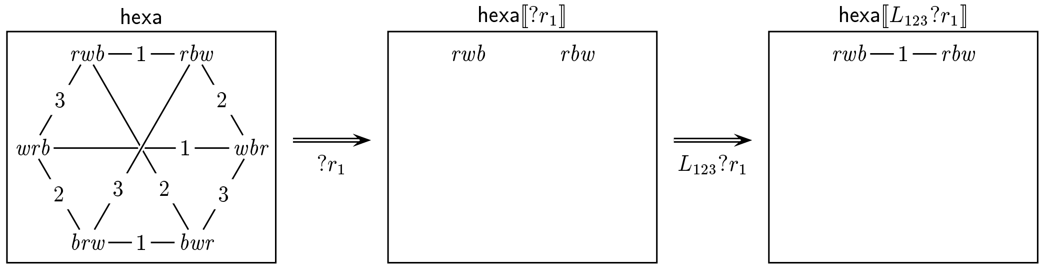


Figure 4.2: Computing $\mathsf{hexa}[\![L_{123}?r_1]\!]$

action types in hexa, see chapter 6.

Example 32 (Local interpretation of the type of tell)
Player 1 puts the red card on the table: $L_{123}?r_1$.

We apply definition 33 stepwise:

$\mathsf{hexa}[\![L_{123}?r_1]\!]M$
$\Leftrightarrow$ definition 33.b.2
$M = \bigoplus_{123}[\mathsf{hexa}[\![?r_1]\!]]$

$\mathsf{hexa}[\![?r_1]\!]M'$
$\Leftrightarrow$ definition 33.b.1
$M' = \langle\{rwb, rbw\}, \emptyset, V\restriction\{rwb, rbw\}\rangle$

Because M' is unique, we may write $M' = \mathsf{hexa}[\![?r_1]\!]$ and we have that $[\mathsf{hexa}[\![?r_1]\!]] = \{\mathsf{hexa}[\![?r_1]\!]\}$. We still have to compute the relations between worlds in hexa and worlds in $\mathsf{hexa}[\![?r_1]\!]$. According to clause 33.c.1: $rwb \rightarrow_{?r_1} rwb$ and $rbw \rightarrow_{?r_1} rbw$. We now compute $M = \mathsf{hexa}[\![L_{123}?r_1]\!]$. Write $M = \langle W^M, \{\sim_1^M, \sim_2^M, \sim_3^M\}, V^M\rangle$. According to clause 33.e.1, $W^M = \{(\mathsf{hexa}[\![?r_1]\!], rwb), (\mathsf{hexa}[\![?r_1]\!], rbw)\}$. As $gr(?r_1) = \emptyset$, clause 33.e.2 doesn't compute any access. For the other agents, i.e. for all agents 1,2 and 3, we continue as follows. An example:

$(\mathsf{hexa}[\![?r_1]\!], rwb) \sim_1^M (\mathsf{hexa}[\![?r_1]\!], rbw)$
$\Leftrightarrow$ definition 33.e.3
$\rightarrow_{?r_1}^{-1}(rwb) \sim_1 \rightarrow_{?r_1}^{-1}(rbw)$
$\Leftrightarrow$ definition 33.b.1 and 33.c.1
$rwb \sim_1 rbw$

The six reflexive links in $\mathsf{hexa}[\![L_{123}?r_1]\!]$ are similarly computed, i.e. for 1, 2, and 3, and for both $(\mathsf{hexa}[\![?r_1]\!], rwb)$ and $(\mathsf{hexa}[\![?r_1]\!], rbw)$. Further, clause 33.$e$.4 states that the valuation V^M on $(\mathsf{hexa}[\![?r_1]\!], rwb)$ and $(\mathsf{hexa}[\![?r_1]\!], rbw)$ is the same as V

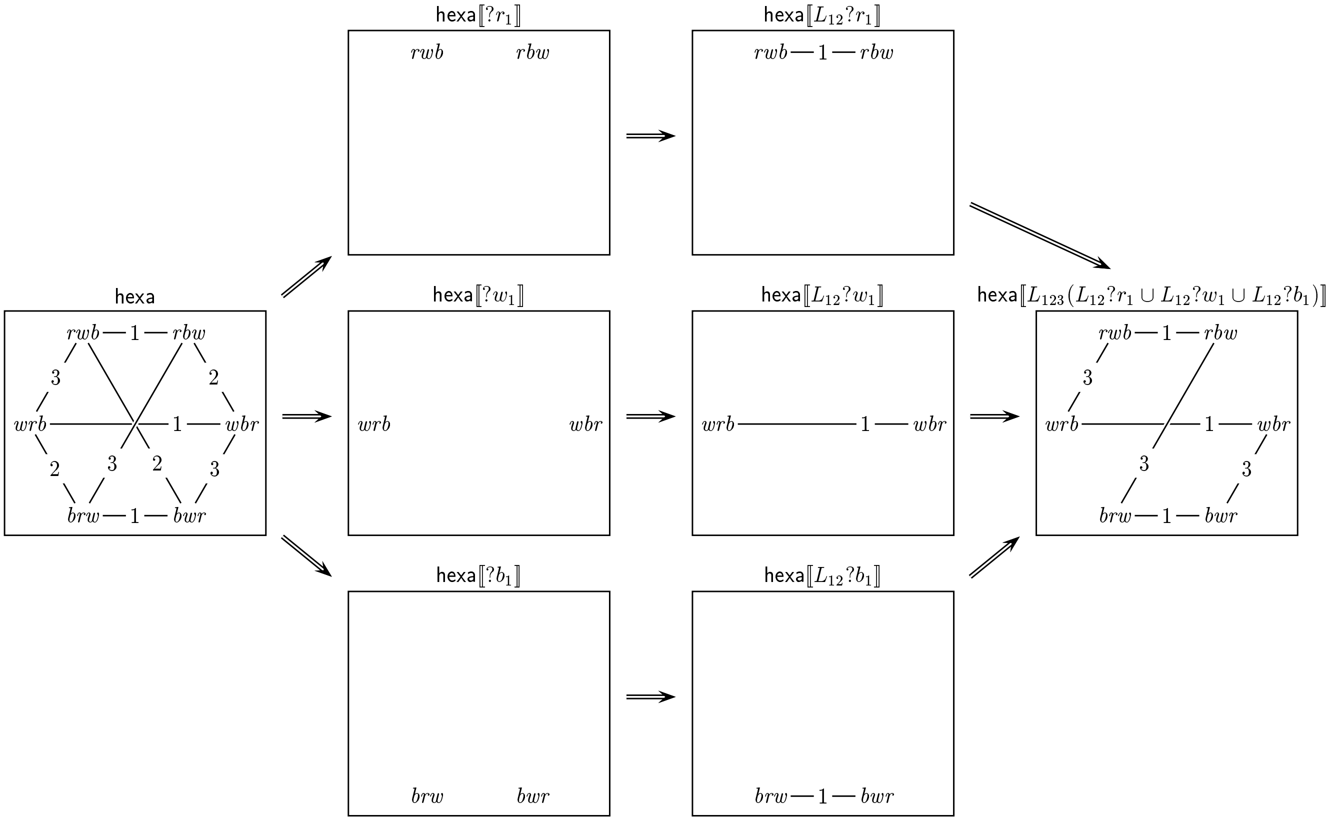


Figure 4.3: Computing $\mathsf{hexa}[\![L_{123}(L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1)]\!]$

on rwb and rbw, respectively. We simplify the notation of domain W^M of model $\mathsf{hexa}[\![L_{123}?r_1]\!]$, as suggested in the remarks on constructing the direct sum, and write rwb for $(\mathsf{hexa}[\![?r_1]\!], rwb)$ and rbw for $(\mathsf{hexa}[\![?r_1]\!], rbw)$. Figure 4.2 gives an overview of our computations.

Example 33 (Local interpretation of the type of show**)**
Player 1 shows his card (only) to player 2: $L_{123}(L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1)$.

We apply definition 33 stepwise, see also figure 4.3:

$\mathsf{hexa}[\![L_{123}(L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1)]\!]M'$
$\Leftrightarrow$ definition 33.b.2
$M' = \bigoplus_{123}[\mathsf{hexa}[\![L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1]\!]]$

$\mathsf{hexa}[\![L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1]\!]M''$
$\Leftrightarrow$ definition 33.b.4
$\mathsf{hexa}[\![L_{12}?r_1]\!]M''$ or $\mathsf{hexa}[\![L_{12}?w_1]\!]M''$ or $\mathsf{hexa}[\![L_{12}?b_1]\!]M''$

Assume that $\cup$ is associative (see proposition 10). There are three different models M''. We compute the first one.

$\mathsf{hexa}[\![L_{12}?r_1]\!]M''$
$\Leftrightarrow$ definition 33.b.2
$M'' = \bigoplus_{12}[\mathsf{hexa}[\![?r_1]\!]]$

Similarly to the computation in example 32, we get $M'' = \mathsf{hexa}[\![L_{12}?r_1]\!] =$

$$\langle\{(\mathsf{hexa}[\![?r_1]\!], rwb), (\mathsf{hexa}[\![?r_1]\!], rbw)\}, \{\sim_1^{M''}, \sim_2^{M''}\}, V^{M''}\rangle$$

Apart from reflexive access for both 1 and 2 in both worlds, we have

$$(\mathsf{hexa}[\![?r_1]\!], rwb) \sim_1^{M''} (\mathsf{hexa}[\![?r_1]\!], rbw)$$

Access for player 3 is not computed. The resulting $\{1, 2\}$ model pictures the result of 1 and 2 learning that 1 holds red, without any assumptions on what 3 has learnt about that. Similarly we compute $\mathsf{hexa}[\![L_{12}?w_1]\!]$ and $\mathsf{hexa}[\![L_{12}?b_1]\!]$. We finish the computation by determining

$$\begin{array}{rcl} M' & = & \mathsf{hexa}[\![L_{123}(L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1)]\!] \\ & = & \bigoplus_{123} \mathsf{hexa}[\![L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1]\!] \\ & = & \mathsf{hexa}[\![L_{12}?r_1]\!] \bigoplus_{123} \mathsf{hexa}[\![L_{12}?w_1]\!] \bigoplus_{123} \mathsf{hexa}[\![L_{12}?b_1]\!] \end{array}$$

The domain W' of M' is the set

$$\begin{array}{rcl} W' & = & \{(\mathsf{hexa}[\![L_{12}?r_1]\!], (\mathsf{hexa}[\![?r_1]\!], rwb)), \\ & & (\mathsf{hexa}[\![L_{12}?r_1]\!], (\mathsf{hexa}[\![?r_1]\!], rbw)), \\ & & (\mathsf{hexa}[\![L_{12}?w_1]\!], (\mathsf{hexa}[\![?w_1]\!], wrb)), \\ & & (\mathsf{hexa}[\![L_{12}?w_1]\!], (\mathsf{hexa}[\![?w_1]\!], wbr)), \\ & & (\mathsf{hexa}[\![L_{12}?b_1]\!], (\mathsf{hexa}[\![?b_1]\!], bwr)), \\ & & (\mathsf{hexa}[\![L_{12}?b_1]\!], (\mathsf{hexa}[\![?b_1]\!], brw))\} \end{array}$$

Again, for convenience we simplify it to $W' = \{rwb, rbw, wrb, wbr, bwr, brw\}$. Access for 1 and 2 remains as it is, using definition 33.e.2. Similarly to the computations for access in example 32, we add reflexive access for player 3 to all worlds. Apart from that we can compute in M' that $rwb \sim_3 wrb$, $rbw \sim_3 brw$, and $bwr \sim_3 wbr$. Figure 4.3 visualizes the resulting model, and how we have constructed it. Note that in any world of the resulting model, player 2 knows the deal of cards. Player 1 doesn't know the cards of 2 and 3, although he knows that 2 knows it. Player 3 knows that 2 knows the deal of cards.

Example 34 (Local interpretation of the type of whisper)
Player 1 whispers in 2's ear a card that he (1) doesn't have: $L_{123}(L_{12}?\neg r_1 \cup L_{12}?\neg w_1 \cup L_{12}?\neg b_1)$.

This is a case of true nondeterminism: in any state of hexa, 1 can choose what to whisper. The model resulting from the execution of this action type is more complex than hexa. We do not perform the computations in detail. Figure 4.1 on page 50 pictures $\mathsf{hexa}[\![L_{123}(L_{12}?\neg r_1 \cup L_{12}?\neg w_1 \cup L_{12}?\neg b_1)]\!]$. We assume access to be transitive. Just as before, we have kept the worlds' original names. Otherwise, e.g., the world wrb 'in front' would be named

$$(\mathsf{hexa}[\![L_{12}?\neg r_1]\!], (\mathsf{hexa}[\![?\neg r_1]\!], wrb))$$

and the world wrb 'at the back' would be named

$$(\mathsf{hexa}[\![L_{12}?\neg b_1]\!], (\mathsf{hexa}[\![?\neg b_1]\!], wrb))$$

Note that we can still distinguish the two worlds named wrb from each other, because player 2 is less informed in front, than at the back.

After a test on your knowledge, it is no longer there! That a type τ can only be interpreted in a model M if $gr(\tau) \subseteq gr(M)$, see the remark ad clause 33.b.3 of definition 33, has consequences that we better make as explicit as possible. E.g. $[?K_1 r_1]K_1 r_1$ can *not* be interpreted on hexa. We have that $\mathsf{hexa}, rwb \models [?K_1 r_1]K_1 r_1 \Leftrightarrow \mathsf{hexa}[\![?K_1 r_1]\!], rwb \models K_1 r_1$. The model $\mathsf{hexa}[\![?K_1 r_1]\!]$ consists of the worlds rwb and rbw without access defined for any agent: it is an $\emptyset$ model. Obviously we cannot interpret the formula $K_1 r_1$ on that model. So in general, the interpretation of $[?K_a\varphi]K_a\varphi$ is undefined. But this is exactly what *must* be! After an action in which it is not specified what 1 has learnt, of course it *must* be the case that we cannot evaluate propositions about 1's knowledge. This is different from the approach in [Ger99], where it is assumed that 1 learns nothing, so that 1 therefore still knows that he holds red after that test on his knowledge. To conclude: never have your knowledge tested – $?K_1\varphi$ – without your consent, either do it yourself – $L_1?K_1\varphi$, or at least *suspect* it – $L_1(?K_1\varphi \cup ?\top)$. The topic of suspicion will be discussed in chapter 6.

Local interpretation of knowledge actions We have defined the local interpretation of a knowledge action type, we now define the local interpretation of a knowledge action of that type. To interpret a knowledge action $\alpha = !_I\tau$ on an $S5$ state (M, w) we first determine the interpretation of τ on M. From the set $[M[\![\tau]\!]]$ we then select *one* model M' and in that model M' *one* world w'. Both are truly selections, as $[M[\![\tau]\!]]$ can contain more than one model, and M' can contain more than one τ-image of w. What M' and w' are, is determined by the bundle I. The world w' is the $!_I\tau$ image of w, to be defined below as a relation $\mapsto_{!_I\tau}$, and M' is simply the model that contains w'. (M', w') is then the required $S5$ state.

Definition 34 (Local interpretation of a knowledge action)
Given are a set of agents $\mathbf{A}$ and a set of atoms $\mathbf{P}$. Let (M, w) be an $S5_n$ state. Let (M', w') be an $S5_{\leq n}$ state. Let $\tau \in \mathsf{KT}$ be an action type. Let $I \in bu(\tau)$. The local interpretation $[\![\cdot]\!]$ of $!_I\tau$ in M is a functional relation between $S5$ states,

defined by inductive cases:

$$\begin{array}{lrll}
34.a.1 & (M,w)[\![!_I\tau]\!](M',w') & \text{iff} & M[\![\tau]\!]M' \text{ and } w \mapsto_{!_I\tau} w' \\
\\
34.b.1 & w \mapsto_{!_{()}?\varphi} w' & \text{iff} & w = w' \\
34.b.2 & w \mapsto_{!_{(I)}L_B\tau'} w' & \text{iff} & \exists M'' \in [M[\![\tau']\!]] : \exists w'' \in M'' : \\
 & & & w' = (M'', w'') \text{ and } w \mapsto_{!_I\tau'} w'' \\
34.b.3 & w \mapsto_{!_{(I,J)}(\tau'' \;;\; \tau')} w' & \text{iff} & \exists M'' : \exists w'' \in M'' : M[\![\tau'']\!]M''[\![\tau']\!]M' \\
 & & & \text{and } w \mapsto_{!_I\tau''} w'' \mapsto_{!_J\tau'} w' \\
34.b.4 & w \mapsto_{!_{0(I,J)}(\tau''\cup\tau')} w' & \text{iff} & w \mapsto_{!_I\tau''} w' \\
34.b.5 & w \mapsto_{!_{1(I,J)}(\tau''\cup\tau')} w' & \text{iff} & w \mapsto_{!_J\tau'} w'
\end{array}$$

Notation Because the local interpretation of an action is a functional relation, instead of $(M,w)[\![!_I\tau]\!](M',w')$ we write $(M',w') = (M,w)[\![!_I\tau]\!]$. If there is only one M' such that $M[\![\tau]\!]M'$, so that $M' = M[\![\tau]\!]$, and if we keep the names that worlds have in M, we can simply write $(M[\![\tau]\!], w)$. Here, w is actually the $\mapsto_{!_I\tau}$-image w' of $w \in M$. We have chosen the $\mapsto$ symbol for its functional connotation: in definition 33.*c* the expression $w \to_\tau w'$ relates w to *one of* its τ-images, whereas in definition 34.*b* the expression $w \mapsto_{!_I\tau} w'$ relates w to its *unique* $!_I\tau$-image.[3]

Example 35 (Local interpretation of action show)
We illustrate definition 34 by computing the interpretation of action show in state (hexa, rwb).

$(\mathsf{hexa}, rwb)[\![L_{123}(!L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1)]\!](M',w')$
$\Leftrightarrow$ definition 34.*a*.1
$\mathsf{hexa}[\![L_{123}(L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1)]\!]M'$ and
$rwb \mapsto_{L_{123}(!L_{12}?r_1\cup L_{12}?w_1\cup L_{12}?b_1)} w'$

The model M' has been computed in example 33 (see figure 4.3). The image of rwb in that model is 'computed' as follows.

$rwb = rwb$
$\Leftrightarrow$ definition 34.*b*.1
$rwb \mapsto_{?r_1} rwb$
$\Leftrightarrow$ definition 34.*b*.2
$rwb \mapsto_{L_{12}?r_1} (\mathsf{hexa}[\![?r_1]\!], rwb)$

[3] We considered two alternative although equivalent forms of definition 34. First, clause 34.*a*.1 could have been: $(M,w)[\![!_I\tau]\!](M',w')$ iff $(M,w)[\![\tau]\!](M',w')$ and $w \mapsto_{!_I\tau} w'$. We preferred a form similar to that of definition 33. Second, the bundle I already contains sufficient information to select the $!_I\tau$-image, so we can write $\mapsto_I$ instead $\mapsto_{!_I\tau}$ – plus some changes in the conditions 34.*b*.2 and 34.*b*.3. As we generally use the notational equivalent for actions, as in example 35, this would have been impractical. Also, our use of the preferred notation $\mapsto_{!_I\tau}$ is similar to that of $\to_\tau$ in definition 33.

$\Leftrightarrow$ definition 34.b.4, twice
$rwb \mapsto_{!L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1} (\mathsf{hexa}[\![?r_1]\!], rwb)$
$\Leftrightarrow$ definition 34.b.2
$rwb \mapsto_{L_{123}(!L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1)} (\mathsf{hexa}[\![L_{12}?r_1]\!], (\mathsf{hexa}[\![?r_1]\!], rwb))$

See also figure 4.1 on page 50. Instead of $(\mathsf{hexa}[\![L_{12}?r_1]\!], (\mathsf{hexa}[\![?r_1]\!], rwb))$ we write rwb.

Compositionality Of course, the semantics of knowledge actions is compositional: the interpretation of an action $!_I\tau$ is a function of I and τ. However, it *appears* to be not compositional when applied to the notational equivalent (definition 30). For example, the interpretation of $!L_{12}?r_1$ in $L_{123}(!L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1)$ is *not* a function of the interpretation of '!' and $L_{12}?r_1$ (whatever the first is supposed to mean, anyway). Instead, it a function of the interpretation of $L_{12}?r_1 = \pi$ and that of the context $L_{123}(\pi \cup L_{12}?w_1 \cup L_{12}?b_1)$ in which it appears.

4.3 Action type properties

We define and prove some properties of interpretations of knowledge actions and knowledge action types.

Definition 35 (Executable)
An action type τ is *executable* in an $S5$ model M, if the local interpretation of τ on M is defined and is not the empty relation. An action type τ is *executable* in an $S5$ state (M, w), if the local interpretation of τ in (M, w) is defined and is not the empty relation: i.e. if it is executable in M and if w has a τ-image under that interpretation. A knowledge action $!_I\tau$ is *executable* in an $S5$ state (M, w), if the local interpretation of $!_I\tau$ in (M, w) is defined and is not the empty relation.

That an action type is executable in a model, does not imply that every action of that type is executable in a state of that model. An example: the action type $L_{123}(L_{12}?r_1 \cup\ ?\bot)$ is executable in hexa, but the action $L_{123}(L_{12}?r_1 \cup\ !?\bot)$ of that type is not executable in any state (hexa, w).

Definition 36 (Equivalence)
Let $\pi, \pi' \in \mathsf{KT} \cup \mathsf{KA}$. Program π is *equivalent* to program π', notation $\pi = \pi'$, if they have the same local interpretation:

$$\pi = \pi' \Leftrightarrow [\![\pi]\!] = [\![\pi']\!]$$

Definition 37 (Public interpretation)
Let M be an $S5$ model, $\tau \in \mathsf{KT}$. If τ is executable in M and if $gr(\tau) = gr(M) = gr(M')$ for all M' such that $M[\![\tau]\!]M'$, then $[\![\tau]\!]$ is the *public interpretation* of τ in M. We also say that τ can be publicly interpreted in M. An action $!_I\tau$ can be

publicly interpreted in a state (M, w) (where $[\![!_I\tau]\!]$ is the public interpretation), if $!_I\tau$ is executable in (M, w) and if τ can be publicly interpreted in M.

An action (type) can be publicly interpreted, if in its informal description the roles of all the agents are fully specified and are not ambiguous. The interpretations of the actions tell, show and whisper in hexa, and of their types, are all public. The interpretation of the action $L_{12}?r_1$ in hexa is not public, i.e. it is 'local' in the stricter meaning of 'not public': only for the subgroup consisting of 1 and 2, and not for the *public* $\{1, 2, 3\}$ of that model. What 3 learns in this action is (entirely unknown and therefore) ambiguous.

Proposition 9 (Every world image has a unique origin)
Let M be an $S5$ model. Let $\tau \in \mathsf{KT}$. Then:

$$\forall M' : M[\![\tau]\!]M' \Rightarrow \forall w' \in M' : \exists! w^* \in M : (M, w^*)[\![\tau]\!](M', w')$$

Proof Induction on τ. E.g. case $L_B\tau$: Suppose $M[\![L_B\tau]\!]M'$. Let $w' \in M'$. As $M' = \bigoplus_B[M[\![\tau]\!]]$, it holds that $w' = (M'', w'')$ for some (exactly one) $w'' \in M'' \in [M[\![\tau]\!]]$. By induction there must be exactly one $w \in M$ such that $(M, w)[\![\tau]\!](M'', w'')$. From definition 33.$a$.1 follows $w \rightarrow_\tau w''$. From that and from definition 33.c.2 follows $w \rightarrow_{L_B\tau} w'$, i.e. $w \rightarrow_{L_B\tau} (M'', w'')$. ■

Therefore, we can write $\rightarrow_\tau^{-1} (w')$ for the unique origin of a world w', as in clause e.3 of definition 33.

Proposition 10 (Associativity)
Let $\tau, \tau', \tau^* \in \mathsf{KT}$. Then:

$$\begin{array}{lrcl} (a) & (\tau \cup \tau') \cup \tau^* & = & \tau \cup (\tau' \cup \tau^*) \\ (b) & (\tau \ ; \ \tau') \ ; \ \tau^* & = & \tau \ ; \ (\tau' \ ; \ \tau^*) \end{array}$$

Proof (a) Suppose $M[\![(\tau \cup \tau') \cup \tau^*]\!]M'$. Then, by definition 33.b.4, either $M[\![\tau \cup \tau']\!]M'$ or $M[\![\tau^*]\!]M'$. If $M[\![\tau \cup \tau']\!]M'$, then, again by definition 33.b.4, either $M[\![\tau]\!]M'$ or $M[\![\tau']\!]M'$. From $M[\![\tau']\!]M'$ or $M[\![\tau^*]\!]M'$ follows $M[\![\tau' \cup \tau^*]\!]M'$. From $M[\![\tau]\!]M'$ or $M[\![\tau' \cup \tau^*]\!]M'$ follows $M[\![\tau \cup (\tau' \cup \tau^*)]\!]M'$. ■
(b) Similar to (a), by decomposing and recomposing according to definition 33.b.3. ■

We have not further investigated algebraic properties of action type operators.

Fact 5 (S5 preservation)
The class of $S5$ models is closed under execution of knowledge action types and execution of knowledge actions.

Obvious. It follows immediately from the constructions of agent access in clauses 33.b.1, 33.e.2, and 33.e.3 of definition 33. ■

Proposition 11.a states that the interpretation of an action is indeed contained in the interpretation of its type. Given how we defined local interpretation for types and for actions, this is not trivial. Proposition 11.b states that, if τ is executable on M, there is a bundle $I \in bu(\tau)$, and a world $w \in M$, such that $!_I\tau$ is executable in state (M, w):

Proposition 11 (Relation between actions and action types)
Let $(M, w), (M', w')$ be $S5$ models, let $!_J\tau \in \mathsf{KA}$. Then:

$$\begin{array}{ll} (a) & (M,w)[\![!_J\tau]\!](M',w') \Rightarrow (M,w)[\![\tau]\!](M',w') \\ (b) & (M,w)[\![\tau]\!](M',w') \Rightarrow \exists I \in bu(\tau) : (M,w)[\![!_I\tau]\!](M',w') \end{array}$$

Proof (a) Induction. For each case, we have to prove that $w \mapsto_{!_J\tau} w' \Rightarrow w \rightarrow_\tau w'$. It then follows that: $(M,w)[\![!_J\tau]\!](M',w') \Leftrightarrow M[\![\tau]\!]M'$ and $w \mapsto_{!_J\tau} w' \Rightarrow M[\![\tau]\!]M'$ and $w \rightarrow_\tau w' \Leftrightarrow (M,w)[\![\tau]\!](M',w')$. We do two cases. Case $?\varphi$: $w \mapsto_{!_{()}?\varphi} w' \Leftrightarrow w = w' \Leftrightarrow w \rightarrow_\varphi w'$. Case $!_{(I)}L_B\tau$:

$w \mapsto_{!_{(I)}L_B\tau} w'$
$\Leftrightarrow$
$\exists M'' \in [M[\![\tau]\!]] : \exists w'' \in M'' : w' = (M'', w'')$ and $w \mapsto_{!_I\tau} w''$
$\Rightarrow$
$\exists M'' \in [M[\![\tau]\!]] : \exists w'' \in M'' : w' = (M'', w'')$ and $w \rightarrow_\tau w''$
$\Leftrightarrow$
$w \rightarrow_{L_B\tau} w'$. ■

(b) Instead of (b) we prove: $(M,w)[\![\tau]\!](M',w') \Rightarrow \exists I \in bu(\tau) : w \mapsto_{!_I\tau} w'$, by induction on τ; (b) then follows immediately, from definition 34. We construct a bundle for τ using definition 34.b. A typical case: Let $(M,w)[\![\tau \cup \tau']\!](M',w')$. Then either $(M,w)[\![\tau]\!](M',w')$ or $(M,w)[\![\tau']\!](M',w')$. Suppose $(M,w)[\![\tau]\!](M',w')$. By induction $\exists I \in bu(\tau)$ such that $w \mapsto_{!_I\tau} w'$. For any $J \in bu(\tau')$, bundle $0(I,J)$ is the required bundle, as $w \mapsto_{!_{0(I,J)}\tau\cup\tau'} w' \Leftrightarrow w \mapsto_{!_I\tau} w'$. ■

We may expect that bisimilarity of models is preserved under execution of actions and types. This is indeed the case:

Proposition 12 (Preservation of bisimilarity)
Let M, M' be $S5$ models, and let $\tau \in \mathsf{KT}$. For every $S5$ model M^* there is an $S5$ model $M^\bullet$ such that:

$$M \underline{\leftrightarrow} M' \text{ and } M[\![\tau]\!]M^* \Rightarrow M'[\![\tau]\!]M^\bullet \text{ and } M^* \underline{\leftrightarrow} M^\bullet$$

We prove something stronger, from which proposition 12 follows immediately:

Lemma for proposition 12: Let M, M' be $S5$ models, and let $\tau \in \mathsf{KT}$. For every $S5$ model M^* there is an $S5$ model $M^\bullet$ such that:

$$\begin{array}{rcl}
\forall\mathfrak{R} : \exists\mathfrak{R}^\tau : & & \\
\mathfrak{R} : M \underline{\leftrightarrow} M' \text{ and } M[\![\tau]\!]M^* & \Rightarrow & M'[\![\tau]\!]M^\bullet \text{ and } \mathfrak{R}^\tau : M^* \underline{\leftrightarrow} M^\bullet \\
& & \text{and } (i) \text{ and } (ii) \\
\text{where } (i) \text{ is } : & & \\
\forall w^* \in M^* : \forall w^\bullet \in M^\bullet : & & \\
\mathfrak{R}^\tau(w^*, w^\bullet) & \Rightarrow & \mathfrak{R}(\rightarrow_\tau^{-1}(w^*), \rightarrow_\tau^{-1}(w^\bullet)) \\
 & & \\
\text{and where } (ii) \text{ is } : & & \\
\forall w \in M : \forall w' \in M' : \forall w^* \in M^* : \exists w^\bullet \in M^\bullet : & & \\
\mathfrak{R}(w, w') \text{ and } w \rightarrow_\tau w^* & \Rightarrow & w' \rightarrow_\tau w^\bullet \text{ and } \mathfrak{R}^\tau(w^*, w^\bullet)
\end{array}$$

Proof: The proof is by induction on the structure of action types, and consists of constructing a proper bisimulation $\mathfrak{R}^\tau$ from a given bisimulation $\mathfrak{R}$, for each inductive case. It can be easily checked that conditions (i) and (ii) are satisfied for all inductive cases. We will therefore skip these parts in the proof. However, they are essentially needed as inductive assumptions in the case $L_B\tau$ of the proof.

Case $?\varphi$: Suppose $\mathfrak{R} : M \underline{\leftrightarrow} M'$ and $M[\![?\varphi]\!]M^*$. Type $?\varphi$ is also executable on M', because if $\mathfrak{R}(w, w')$ then $M, w \models \varphi \Leftrightarrow M', w' \models \varphi$. Let $M^\bullet$ be that model $M'[\![?\varphi]\!]$. Define for all worlds $w \in M^*$ and $v \in M'[\![?\varphi]\!]$: $\mathfrak{R}^{?\varphi}(w, v) \Leftrightarrow \mathfrak{R}(w, v)$. Relation $\mathfrak{R}^{?\varphi}$ is a bisimulation between $M[\![?\varphi]\!]$ and $M'[\![?\varphi]\!]$, because both $M[\![?\varphi]\!]$ and $M'[\![?\varphi]\!]$ are $\emptyset$ models (models without access) and because $\mathfrak{R}^{?\varphi}(w, v)$ implies that $\mathfrak{R}(w, v)$, which implies that $V_w = V_v$.

Case $L_B\tau$: Suppose $\mathfrak{R} : M \underline{\leftrightarrow} M'$ and $M[\![L_B\tau]\!]M^*$. From definition 33 follows that $M^* = \bigoplus_B[M[\![\tau]\!]]$. We show that the (unique) model $M^\bullet = \bigoplus_B[M'[\![\tau]\!]]$ is the *required model*. By induction, for each $M_i \in [M[\![\tau]\!]]$ we have an $M'_i \in [M'[\![\tau]\!]]$ such that $M_i \underline{\leftrightarrow} M'_i$; suppose $\mathfrak{R}_i : M_i \underline{\leftrightarrow} M'_i$. We show that $M^\bullet = \bigoplus_B[M'[\![\tau]\!]]$ is the required model. For all $w \in M_i \in [M[\![\tau]\!]]$, $w' \in M'_i \in [M'[\![\tau]\!]]$, define $\mathfrak{R}^\oplus((M_i, w), (M'_i, w')) \Leftrightarrow \mathfrak{R}_i(w, w')$. Obviously, $\mathfrak{R}^\oplus : \bigoplus[M[\![\tau]\!]] \underline{\leftrightarrow} \bigoplus[M'[\![\tau]\!]]$. We show that also $\mathfrak{R}^\oplus : \bigoplus_B[M[\![\tau]\!]] \underline{\leftrightarrow} \bigoplus_B[M'[\![\tau]\!]]$, so that $\mathfrak{R}^\oplus$ is the *required bisimulation* $\mathfrak{R}^{L_B\tau}$.

We do *forth*, *back* is similar: Suppose that $v, v'' \in \bigoplus_B[M[\![\tau]\!]]$, where $v = (M_1, v_1)$ such that $v_1 \in M_1 \in [M[\![\tau]\!]]$ and $v'' = (M_2, v''_2)$ such that $v''_2 \in M_2 \in [M[\![\tau]\!]]$, and that $w \in \bigoplus_B[M'[\![\tau]\!]]$, where $w = (M'_1, w_1)$ such that $w_1 \in M'_1 \in [M'[\![\tau]\!]]$, and that $\mathfrak{R}^\oplus(v, w)$, and that $v \sim_b v''$. We have to find a w'' with $\mathfrak{R}^\oplus(v'', w'')$ and $w \sim_b w''$. Distinguish case $M_1 = M_2$ from case $M_1 \neq M_2$.

If $M_1 = M_2$, then (by definition 33): $v \sim_b v'' \Leftrightarrow (M_1, v_1) \sim_b (M_1, v''_2) \Leftrightarrow v_1 \sim_b v''_2$. As $\mathfrak{R}_1(v_1, w_1)$, by induction there is a $w''_2 \in M'_1$ such that $\mathfrak{R}_1(v''_2, w''_2)$ and $w_1 \sim_b w''_2$. Therefore $\mathfrak{R}^\oplus((M_1, v''_2), (M'_1, w''_2))$ and $(M'_1, w_1) \sim_b (M'_1, w''_2)$, in other words: $w'' = (M'_1, w''_2)$ is the required world.

If $M_1 \neq M_2$, then (by definition 33) $v \sim_b v''$ can only be the case if $b \neq gr(M_1) \cup gr(M_2)$, and we have that $v \sim_b v'' \Leftrightarrow (M_1, v_1) \sim_b (M_2, v_2'') \Leftrightarrow \rightarrow_\tau^{-1} (v_1) \sim_b \rightarrow_\tau^{-1} (v_2'')$. Further, $\mathfrak{R}^\oplus(v, w) \Leftrightarrow \mathfrak{R}^\oplus((M_1, v_1), (M_1', w_1)) \Leftrightarrow \mathfrak{R}_1(v_1, w_1)$. Applying (i) for τ, by induction, from $\mathfrak{R}_1(v_1, w_1)$ follows $\mathfrak{R}(\rightarrow_\tau^{-1}(v_1), \rightarrow_\tau^{-1}(w_1))$. From $\rightarrow_\tau^{-1}(v_1) \sim_b \rightarrow_\tau^{-1}(v_2'')$ and $\mathfrak{R}(\rightarrow_\tau^{-1}(v_1), \rightarrow_\tau^{-1}(w_1))$ follows that there is an $x \in M'$ such that $\rightarrow_\tau^{-1}(w_1) \sim_b x$ and $\mathfrak{R}(\rightarrow_\tau^{-1}(v_2''), x)$. From $\mathfrak{R}(\rightarrow_\tau^{-1}(v_2''), x)$, $\rightarrow_\tau^{-1}(v_2'') \rightarrow_\tau v_2''$ and (ii) (for case τ) follows that there must be a $w_2'' \in M_2'$ such that $x \rightarrow_\tau w_2''$ and $\mathfrak{R}_2(v_2'', w_2'')$. Let $w'' := (M_2', w_2'')$. We now have that $w \sim_b w''$, because again by definition 33 it holds that $w \sim_b w'' \Leftrightarrow (M_1', w_1) \sim_b (M_2', w_2'') \Leftrightarrow \rightarrow_\tau^{-1}(w_1) \sim_b \rightarrow_\tau^{-1}(w_2'')$, and we have that $\mathfrak{R}^\oplus(v'', w'')$, because $\mathfrak{R}_2(v_2'', w_2'')$.

Case $\tau\ ;\ \tau'$: Suppose $M \underline{\leftrightarrow} M'$ and $M[\![\tau\ ;\ \tau']\!]M^*$. By definition 33, there must be an M_1 such that $M[\![\tau]\!]M_1$ and $M_1[\![\tau']\!]M^*$. By induction, there is an M_1' such that $M'[\![\tau]\!]M_1'$ and $M_1 \underline{\leftrightarrow} M_1'$. Because $M_1 \underline{\leftrightarrow} M_1'$ and $M_1[\![\tau']\!]M^*$, again by induction, there must be an $M^\bullet$ such that $M_1'[\![\tau']\!]M^\bullet$ and $M^* \underline{\leftrightarrow} M^\bullet$.

Case $\tau \cup \tau'$: Suppose $M \underline{\leftrightarrow} M'$ and $M[\![\tau \cup \tau']\!]M^*$. By definition 33.*b*.4: $M[\![\tau]\!]M^*$ or $M[\![\tau']\!]M^*$. If $M[\![\tau]\!]M^*$, then by induction, there is an $M^\bullet$ such that $M'[\![\tau]\!]M^\bullet$ and $M^* \underline{\leftrightarrow} M^\bullet$. From $M'[\![\tau]\!]M^\bullet$ follows, again by definition 33.*b*.4, that $M'[\![\tau \cup \tau']\!]M^\bullet$. If $M[\![\tau']\!]M^*$, then by induction, there is an M'' such that $M[\![\tau']\!]M''$ and $M^* \underline{\leftrightarrow} M''$. From $M'[\![\tau']\!]M''$ follows, again by definition 33.*b*.4, that $M'[\![\tau \cup \tau']\!]M''$. ■

The following are direct consequences of proposition 12:

Proposition 13
(*a*) Let M, M' be $S5$ models, and $\tau \in \mathsf{KT}$. If $|[M[\![\tau]\!]]| = 1$, then:

$$M \underline{\leftrightarrow} M' \Rightarrow M[\![\tau]\!] \underline{\leftrightarrow} M'[\![\tau]\!]$$

(*b*) Let $(M, w), (M', w')$ be $S5$ states, and $\alpha \in \mathsf{KA}$. If α is executable in (M, w), then:

$$(M, w) \underline{\leftrightarrow} (M', w') \Rightarrow (M, w)[\![\alpha]\!] \underline{\leftrightarrow} (M', w')[\![\alpha]\!]$$

4.4 Further observations

Simultaneous execution The program class KT of action types might be extended by adding the operation of simultaneous execution. We give an example:

Example 36
There are three players (1,2,3) and four cards north, east, south and west (n, e, s, w). Player 1 holds north, player 2 east and player 3 south and west. Player 3 shows his south card (only) to player 1, with his left hand, and (simultaneously) his west card (only) to player 2, with his right hand.

We add a new action type construction operator: simultaneous execution $\cap$ (e.g. see [Gol92]). The action in example 36 can be described in more detail as: "(1 and 3 learn that 3 holds south) and simultaneously (2 and 3 learn that 3 holds west) and (simultaneously) (1, 2 and 3 learn that (1 and 3 learn a card of 3) and simultaneously (2 and 3 learn the other card of 3))". This is expressed by the 'knowledge action':

$$L_{123}(!(L_{13}?s_3 \cap L_{23}?w_3) \cup \bigcup_{i \neq j, (i,j) \neq (s,w)} (L_{13}?i_3 \cap L_{23}?j_3))$$

The incorporation of an operator for simultaneous execution would be a valuable extension to our language. We haven't further investigated this. Other epistemic action languages ([Bal99]) contain such an operation.

History-related actions The interpretation of real actions in games may require not just the current game state, but also the action *history*. We give an example:

Example 37
Player 3 shows his south card (only) to player 1; next, player 2 asks player 3 to show him his other card; player 3 shows his west card (only) to player 2

In every game state resulting from the first part of the action, the information is lost which card has been shown, so that the reference in the second part of the action cannot be deduced from this state. To describe that, would require a very different action language. We also haven't further investigated that.

Axiomatization We did not validate our semantics by an axiomatization of the dynamic language $\mathcal{L}_n^\Box$ for a given set of atoms $\mathbf{P}$ and agents $\mathbf{A}$. We expect the usual axioms for a dynamic logic and for the learning operator we expect axioms such as $[L_B\tau]p \leftrightarrow p$ and $[L_B\tau]\neg\varphi \leftrightarrow \neg[L_B\tau]\varphi$. In chapter 7 we define embeddings of subclasses of $\mathsf{KA} \cup \mathsf{KT}$ into the languages of other researchers ([Ger99, Bal99]). They have given axiomatizations. This may provide further clues.

4.5 Conclusion

We proposed a dynamic epistemic language $\mathcal{L}_n^\Box$, that includes a language KT of action types and a derived language KA of knowledge actions. Basic to our approach is the concept of local interpretation of an action type in a model: the interpretation for a subgroup of agents only. We performed detailed computations on some example knowledge actions, to illustrate the language and its interpretation. In chapter 7 we compare our approach to that of other researchers. Future research should include an axiomatization to validate our approach. We also consider extensions of the action language. We have not encountered the notion of local interpretation in the literature and regard it as our contribution to epistemic semantics.

Chapter 5

Descriptions of game actions

In chapter 2 we defined a format for game actions. In chapter 4 we defined a programming language KA for knowledge actions. We now show that every game action is described by a knowledge action.

A KA action is interpreted as a state transformer by *local interpretation*. Game actions are themselves semantic objects. In order to relate knowledge actions to game actions, in section 5.1 we define an alternative semantics for knowledge actions, called *product interpretation* (related to [Bal99], see also chapter 7). The product interpretation uses a knowledge action frame, which is a semantic object quite similar to a game action frame. We show that execution of knowledge actions produces bisimilar states for both notions of interpretation. In section 5.2, we give the precise relation between game action frames and knowledge action frames, and we define a procedure to construct from a given game action a knowledge action that describes it. In section 5.3 we treat an entirely different topic related to knowledge game states: A measure for the complexity of a finite $S5$ state is the number of its (nonsimilar) worlds. As a result of action execution, the complexity decreases in more cases than one may think at first sight.

We start with an example that illustrates the relation between knowledge actions and game actions, see also figure 5.1. In the initial model (hexa, rwb) of the hexa game, the game action where player 2 asks player 1 for his card, and player 1 responds by showing his card, is:

$$\mathsf{show}^{2,-}_{1,r} = \langle 2, \{\{rwb, rbw\}, \{wrb, wbr\}, \{brw, bwr\}\}, 1, \{rwb, rbw\}, \mathsf{show}\rangle$$

where show_1 and show_2 are the identity and show_3 is the universal relation on the question. See chapter 2. If we abstract from its game parameters, this game action is represented as the pointed frame $(\mathsf{show}^{2,-}_{1,r})^- = (\langle\{\{rwb, rbw\}, \{wrb, wbr\}, \{brw, bwr\}\}, \mathsf{show}\rangle, \{rwb, rbw\})$. The knowledge action expressing the same as this game action is (see chapter 4):

$$\alpha_r = L_{123}(!L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1)$$

Knowledge action α_r can be transformed into game action frame $(\mathsf{show}^{2,-}_{1,r})^-$ as follows: The set of actions of this type not only consists of α_r but also of the different actions α_w and α_b where instead 1 shows white or blue. Although 1

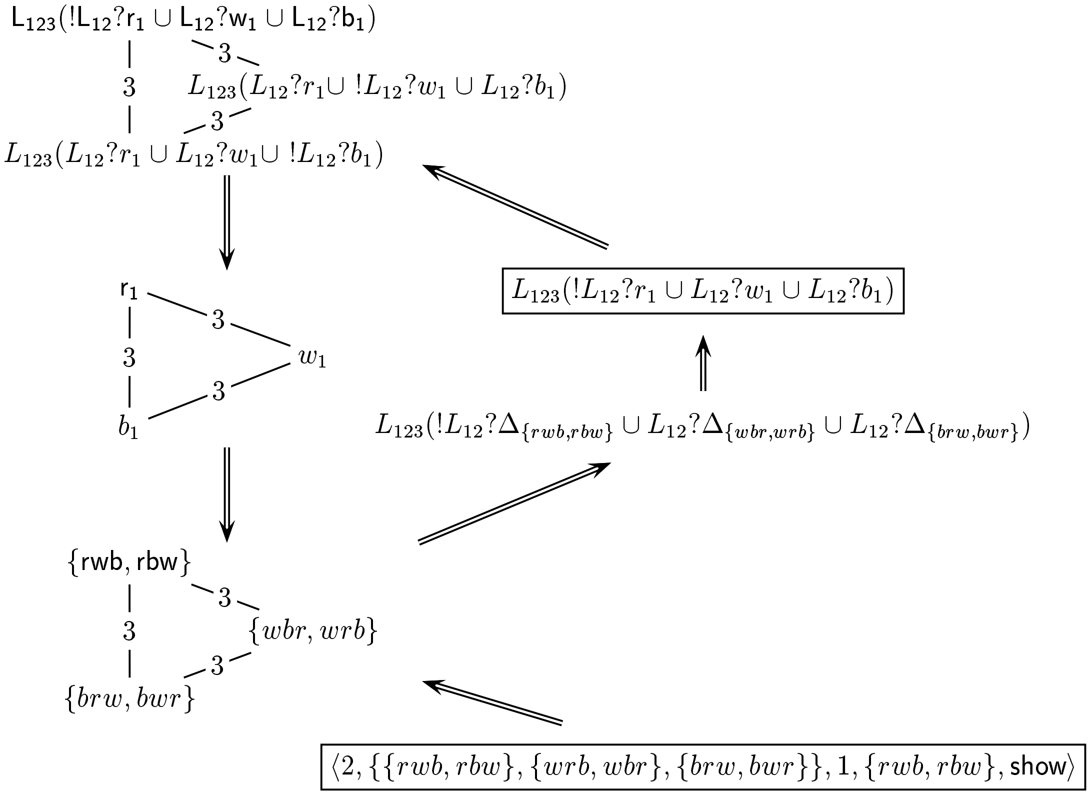


Figure 5.1: From knowledge actions to game actions, and vice versa

and 2 can tell all three actions apart, 3 cannot distinguish between them. This defines access between these knowledge actions. Therefore, to α_r corresponds a pointed $S5$ frame on the set of actions of this type, with point α_r. This is called the *knowledge action frame* $[\![\alpha_r]\!]^{\oplus}$. It provides us with an alternative notion of interpretation: the product interpretation of action α_r in state (hexa, rwb) computes the model that consists of pairs of worlds (w, α_c), with $c = r, w, b$ such that the *precondition* of α_c is satisfied in (hexa, w), and two world-action pairs are accessible for an agent if both the worlds and the actions are accessible for that agent. An example: The precondition of α_r is r_1, which is satisfied in worlds rwb and rbw of hexa. Therefore, both (rwb, α_r) and (rbw, α_r) are in the product interpretation. Also, they are 1-accessible to each other, because $rwb \sim_1 rbw$ and $\alpha_r \sim_1 \alpha_r$. If we replace the actions in $[\![\alpha_r]\!]^{\oplus}$ by their preconditions, and these preconditions by their extensions in hexa, we get the game action frame $(\mathsf{show}^{2,-}_{1,r})^{-}$.

From the game action $\mathsf{show}^{2,-}_{1,r}$ we can also construct a knowledge action that describes it. Take as preconditions of actions the characterizations of possible answers, which is simply the disjunction of the state descriptions of the worlds in that answer. E.g., given state descriptions $\Delta_{rwb} := \delta_{rwb} \wedge C_{123}\mathsf{33}$ and $\Delta_{rbw} := \delta_{rbw} \wedge C_{123}\mathsf{33}$ for rwb and rbw, respectively, the possible answer $\{rwb, rbw\}$ is characterized by $\Delta_{\{rwb,rbw\}} := \Delta_{rbw} \vee \Delta_{rbw}$. These preconditions are the test formulas in our programs. Access on the game action frame is matched by operators for learning and choice, and the point of the frame by local choice. We postpone illustrating that, to section 5.2.

We now continue with the general definitions.

5.1 Product interpretation

In this section we define the *product interpretation* of knowledge actions and types. We show that it corresponds to *local interpretation*, as defined in chapter 4. First we introduce some useful concepts. The product interpretation of a knowledge action in a given state, results from applying a *knowledge action frame* to that state. This frame is defined on the *set of knowledge actions* of the same type as the particular action, based on a notion of *accessibility between actions* of the same type; in order to determine access, we introduce the operation of *actually learning agent* (*ala*).

Definition 38 (Set of actions of a given type)
Let $\tau \in \mathsf{KT}$ be a knowledge action type. Then:

$$\mathsf{KA}(\tau) := \{!_I \tau \mid I \in bu(\tau)\}$$

is the set of knowledge actions of type τ.

Definition 39 (Actually learning agents)
The actually learning agents ala are defined on knowledge actions, by inductive cases: $ala(!_{()}?\varphi) = \emptyset$, $ala(!_{(I)}L_B\tau) = B$, $ala(!_{0(I,J)}(\tau \cup \tau')) = ala(!_I\tau)$, $ala(!_{1(I,J)}(\tau \cup \tau')) = ala(!_J\tau')$, $ala(!_{(I,J)}(\tau \ ; \ \tau')) = ala(!_J\tau')$.

Compare this to the definition of the learning agents of a knowledge action type, $la(\tau)$, in chapter 4. Note that $ala(!_I\tau) \subseteq la(\tau)$; we have e.g. that $la(L_{12}?r_1 \cup L_{23}?w_2) = \{1,2,3\}$ whereas $ala(!L_{12}?r_1 \cup L_{23}?w_2) = \{1,2\}$ and $ala(L_{12}?r_1 \cup !L_{23}?w_2) = \{2,3\}$. The operation ala gives us *just* the finer structure needed to distinguish between actions of the same type:

Definition 40 (Accessibility between knowledge actions)
We define by inductive cases whether two actions are indistinguishable for a player a:

$$\begin{array}{rll}
!_{()}?\varphi \sim_a !_{()}?\varphi & & \text{never} \\
!_{(I)}L_B\tau \sim_a !_{(J)}L_B\tau & \text{iff} & !_I\tau \sim_a !_J\tau \text{ or } a \in B \setminus (ala(!_I\tau) \cup ala(!_J\tau)) \\
!_{(I,J)}(\tau \ ; \ \tau') \sim_a !_{(K,L)}(\tau \ ; \ \tau') & \text{iff} & !_I\tau \sim_a !_K\tau \text{ and } !_J\tau' \sim_a !_L\tau' \\
!_{i(I,J)}(\tau \cup \tau') \sim_a !_{j(K,L)}(\tau \cup \tau') & \text{iff} & i = 0 \text{ and } !_I\tau \sim_a !_K\tau \text{ or} \\
& & i = 1 \text{ and } !_J\tau' \sim_a !_L\tau'
\end{array}$$

Note that actions can only be the same for an agent a if he is learning something in them: if $a \in la(\tau)$. Therefore even identical tests cannot be the same for any agent: $la(?\varphi) = \emptyset$. However, two actions $!_{(I)}L_B\tau$ and $!_{(J)}L_B\tau$ *may* be the same for an agent $a \in B$, namely if he is not *actually* learning anything in either $!_I\tau$ or $!_J\tau$: if $a \notin ala(!_I\tau) \cup ala(!_J\tau')$.

We now define the frame corresponding to a knowledge action type as follows:

Definition 41 (Knowledge action type frame)
Let $\tau \in \mathsf{KT}$. The *knowledge action type frame* of τ is:

$$[\![\tau]\!]^{\otimes} = \langle \mathsf{KA}(\tau), (\sim_a)_{a\in\mathbf{A}}\rangle$$

where for all $a \in \mathbf{A}$, $\sim_a$ is access as in definition 40.

Definition 42 (Knowledge action frame)
Let $!_I\tau \in \mathsf{KA}$. The *knowledge action frame* of $!_I\tau$ is the pointed frame:

$$[\![!_I\tau]\!]^{\otimes} = ([\![\tau]\!]^{\otimes}, !_I\tau)$$

Frames for knowledge actions and knowledge action types may not be $S5$ (see example 40). Because it is a semantical object, such frames may be called the interpretation of the corresponding knowledge action (type). Therefore we write $[\![\tau]\!]^{\otimes}$. The notion $[\![\cdot]\!]^{\otimes}$ is called *product interpretation* to distinguish it from the notion $[\![\cdot]\!]$ (no superscript) of *local interpretation*. We prefer to think of both notions of interpretation as state transformers, in the case of $[\![\tau]\!]^{\otimes}$ this is induced by definition 44, that we will now present. In that definition we need the concept of *precondition* of an action:[1]

Definition 43 (Precondition of a knowledge action)
The precondition $pre(\alpha)$ of a knowledge action α is defined by inductive cases:

$$\begin{array}{rcl} pre(!_{()}?\varphi) & = & \varphi \\ pre(!_{(I)}L_B\tau) & = & pre(!_I\tau) \\ pre(!_{(I,J)}(\tau\ ;\ \tau')) & = & pre(!_I\tau) \wedge [!_I\tau]pre(!_J\tau') \\ pre(!_{0(I,J)}(\tau \cup \tau')) & = & pre(!_I\tau) \\ pre(!_{1(I,J)}(\tau \cup \tau')) & = & pre(!_J\tau') \end{array}$$

Definition 44 (Product interpretation of an action type)
Given are a set $\mathbf{P}$ of atoms and a set $\mathbf{A}$ of agents. Let $M = \langle W, (\sim_a)_{a\in\mathbf{A}}, V\rangle$ be an $S5$ model. Let $\tau \in \mathsf{KT}$ and $[\![\tau]\!]^{\otimes} = \langle \mathsf{KA}(\tau), (\sim_a)_{a\in B}\rangle$. The *product interpretation* of τ in M is the model:

$$M \otimes [\![\tau]\!]^{\otimes} := \langle W', (\sim'_a)_{a\in B}, V'\rangle$$

[1] In [Bal99], the semantic object comparable to a knowledge action frame is not - just - defined on alternative actions but on abstract 'tokens' that are characterized by the preconditions of those actions.

where

$$W' = \{(w, !_I\tau) \in W \times \mathsf{KA}(\tau) \mid M, w \models pre(!_I\tau)\}$$

and $\forall (w, !_I\tau), (w', !_J\tau) \in M \otimes [\![\tau]\!]^\otimes : \forall a \in B$:

$$\begin{array}{rcl} (w, !_I\tau) \sim'_a (w', !_J\tau) & \Leftrightarrow & w \sim_a w' \text{ and } !_I\tau \sim_a !_J\tau \\ V'_{(w,!_I\tau)} & = & V_w \end{array}$$

Definition 45 (Product interpretation of a knowledge action)
The product interpretation of a knowledge action $!_I\tau \in \mathsf{KA}$ in an $S5$ state (M, v) is the state

$$(M, v) \otimes [\![!_I\tau]\!]^\otimes = (M \otimes [\![\tau]\!]^\otimes, (v, !_I\tau))$$

The product interpretation $M \otimes [\![\tau]\!]^\otimes$ is not necessarily an $S5$ model! It is not an $S5$ model, if different actions of type τ have different groups of actually learning agents. See e.g. example 40. Still, in such a case the resulting model consists of disconnected components of $S5$ models ($S5_{\leq n}$ models, if M is an $S5_n$ model).

The notions of local interpretation $[\![\cdot]\!]$ and product interpretation $[\![\cdot]\!]^\otimes$ are *almost* the same. The differences are inessential: the resulting models will still be bisimilar. First, we give an example where local and product interpretation correspond: see example 38. Then we give two examples where they are different. The differences are due to the fact that $[\![\cdot]\!]$ is *not* functional but that $[\![\cdot]\!]^\otimes$ is functional. If τ is executable on M, $M \otimes [\![\tau]\!]^\otimes$ may be the direct sum $\bigoplus[M[\![\tau]\!]]$ of all models in $[M[\![\tau]\!]]$, as in example 39. In that case, however, the resulting models are still bisimilar. If $[M[\![\tau]\!]]$ consists of models for different groups of agents (or, differently said, if the group of actually learning agents – $ala(!_I\tau)$ – is *not* the same for all actions of that type τ), the resulting models are also not the same, but, again, are bisimilar, as the connected components containing related points correspond. See example 40.

Example 38
The product interpretation of $L_{123}(L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1)$ on the model hexa is pictured in figure 5.2. Write α_c for the action where card c is shown. It is isomorphic to (local interpretation) $\mathsf{hexa}[\![L_{123}(L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1)]\!]$, see chapter 4. We leave detailed computations to the reader but give one example: note that $(rwb, \alpha_r) \sim_1 (rbw, \alpha_r)$ because $\alpha_r \sim_1 \alpha_r$ (the action of showing red is 'indistinguishable from itself' for all agents) and $rwb \sim_1 rbw$ (1 holds red in both rwb and rbw). See also the introductory part to this chapter.

Example 39
The product interpretation of the action type $L_1?w_2 \cup L_1?w_2$ on hexa is the model consisting of *four* worlds $(rwb, !L_1?w_2 \cup L_1?w_2)$ and $(bwr, !L_1?w_2 \cup L_1?w_2)$,

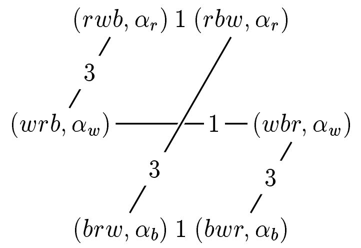


Figure 5.2: Product interpretation in hexa of 1 showing his card to 2

$(rwb, L_1?w_2\cup\ !L_1?w_2)$ and $(bwr, L_1?w_2\cup\ !L_1?w_2)$, with only reflexive access for player 1. However, the local interpretation of that action type consists of *two* worlds only. These worlds are the two states of the $\emptyset$-model consisting of the deals rwb and bwr: According to the definition of local interpretation, the action $!L_1?w_2 \cup L_1?w_2$ is identical to the action $L_1?w_2\cup\ !L_1?w_2$, because they define the same relation between knowledge states: for all M, $[M[\![!L_1?w_2 \cup L_1?w_2]\!]] = [M[\![L_1?w_2\cup\ !L_1?w_2]\!]] = [M[\![L_1?w_2]\!]] = \{M[\![L_1?w_2]\!]\}$.

Example 40
The product interpretation of the action type $L_1?r_1 \cup L_2?w_2$ on hexa is the (not $S5$!) model consisting of four worlds $(rwb, !L_1?r_1 \cup L_2?w_2)$, $(rbw, !L_1?r_1 \cup L_2?w_2)$, $(rwb, L_1?r_1\cup\ !L_2?w_2)$, and $(bwr, L_1?r_1\cup\ !L_2?w_2)$, with $(rwb, !L_1?r_1 \cup L_2?w_2) \sim_1 (rbw, !L_1?r_1 \cup L_2?w_2)$ and $(rwb, L_1?r_1\cup\ !L_2?w_2) \sim_2 (bwr, L_1?r_1\cup\ !L_2?w_2)$. Note that the knowledge action type frame for $L_1?r_1 \cup L_2?w_1$ is the frame consisting of the two actions $!L_1?r_1 \cup L_2?w_1$, accessible from itself for 1, and $L_1?r_1\cup\ !L_2?w_1$, accessible from itself for 2. Indeed, the local interpretation of $L_1?r_1 \cup L_2?w_2$ on hexa consists of a 1-model and a 2-model, both consisting of two worlds. These are the two components of the product interpretation.

Before we present proposition 15 on the relation between local and product interpretation, we need to prove a simple property relating local interpretation to preconditions of actions:

Proposition 14 (Preconditions of executable actions)
A knowledge action is executable in a state, if and only if its precondition is satisfied in that state. Let (M, w) be an $S5$ state, let $!_I\tau \in \mathsf{KA}$. Then:

$$(\exists (M', w') : (M, w)[\![!_I\tau]\!](M', w')) \Leftrightarrow M, w \models pre(!_I\tau)$$

Proof By induction on τ, using the (inductive) definitions of pre and of local interpretation $[\![\cdot]\!]$. For an example, two cases.

Case $?\varphi$: We have that $(M, w)[\![!_{()}?\varphi]\!](M', w') \Leftrightarrow M[\![?\varphi]\!]M'$ and $w \mapsto_{!_{()}?\varphi} w' \Leftrightarrow w = w'$ and $M, w \models \varphi$. Also: $M, w \models \varphi \Leftrightarrow M, w \models pre(!_{()}?\varphi)$. Therefore, given that we have such a M' and w', it follows that $M, w \models pre(!_{()}?\varphi)$; and given that $M, w \models pre(!_{()}?\varphi)$, choose $w' = w$ and $M' = M[\![?\varphi]\!]$.

Case τ ; τ': We do just '$\Rightarrow$' ('$\Leftarrow$' is somewhat similar):
Let M', w' be such that $(M,w)[\![!_{(I,J)}(\tau \ ; \ \tau')]\!](M',w')$. There must be M'', w'' such that $(M,w)[\![!_I\tau]\!](M'',w'')$ and $(M'',w'')[\![!_J\tau']\!](M',w')$. From $(M,w)[\![!_I\tau]\!](M'',w'')$ follows, using induction, that $M,w \models pre(!_I\tau)$. As the interpretation of actions is functional, from $(M'',w'')[\![!_J\tau']\!](M',w')$ follows that we also have that for arbitrary (M^*,w^*), if $(M,w)[\![!_I\tau]\!](M^*,w^*)$ then $(M^*,w^*)[\![!_J\tau']\!](M',w')$. From $(M^*,w^*)[\![!_J\tau']\!](M',w')$ follows, using induction, that $(M^*,w^*) \models pre(!_J\tau')$. Therefore $(M,w) \models [!_I\tau]pre(!_J\tau')$. We continue by:

$M,w \models pre(!_I\tau)$ and $M,w \models [!_I\tau]pre(!_J\tau')$
$\Leftrightarrow$
$M,w \models pre(!_I\tau) \wedge [!_I\tau]pre(!_J\tau')$
$\Leftrightarrow$
$M,w \models pre(!_{(I,J)}(\tau \ ; \ \tau'))$ ■

Notation Write $w[\![!_I\tau]\!]$ for the unique w' such that $(M,w)[\![!_I\tau]\!](M',w')$, i.e. such that $w \mapsto_{!_I\tau} w'$. (Instead of $w[\![!_I\tau]\!]$ we could also have written $\mapsto_{!_\tau}(w)$.)

Proposition 15
Let M be an $S5$ model and $\tau \in \mathsf{KT}$ be a knowledge action type executable in M, then:

$$\bigoplus[M[\![\tau]\!]] \underline{\leftrightarrow} M \otimes [\![\tau]\!]^{\otimes}$$

Proof The following relation $\mathfrak{R}$ defines a bisimulation between $\bigoplus[M[\![\tau]\!]]$ and $M \otimes [\![\tau]\!]^{\otimes}$. For all $w \in M$, for all $I \in bu(\tau)$ such that $!_I\tau$ is executable in (M,w):

$$\mathfrak{R}(w[\![!_I\tau]\!], (w, !_I\tau))$$

Note that proposition 14 guarantees that $w[\![!_I\tau]\!]$ is a meaningful expression in the context of the definition: if $(w, !_I\tau)$ is in the domain of $M \otimes [\![\tau]\!]^{\otimes}$, it holds that $(M,w) \models pre(!_I\tau)$ and therefore $!_I\tau$ is executable in (M,w), so that we can use the notation $w[\![!_I\tau]\!]$ for its image under that execution. The relation $\mathfrak{R}$ induces a surjection but does not induce a bijection from $M \otimes [\![\tau]\!]^{\otimes}$ to $\bigoplus[M[\![\tau]\!]]$, because different actions $w[\![!_I\tau]\!]$ and $w[\![!_J\tau]\!]$ may define the same world in $[M[\![\tau]\!]]$, see example 39. To establish that the relation is a bisimulation, note that $V_{w[\![!_I\tau]\!]} = V_w = V_{(w,!_I\tau)}$ and that both 'back' and 'forth' immediately follow from the following observation:

$w[\![!_I\tau]\!] \sim_a w'[\![!_J\tau]\!]$
$\Leftrightarrow$ by lemma λ below, that is proven by induction on τ
$w \sim_a w'$ and $!_I\tau \sim_a !_J\tau$
$\Leftrightarrow$ by definition 44
$(w, !_I\tau) \sim_a (w', !_J\tau)$

We now prove the following lemma (λ):

Lemma λ for proposition 15
Let M be an $S5$ model, τ be a knowledge type, $w, w' \in M$, and $I, J \in bu(\tau)$. Then $w[\![!_I\tau]\!] \sim_a w'[\![!_J\tau]\!] \Leftrightarrow w \sim_a w'$ and $!_I\tau \sim_a !_J\tau$.

Proof: By induction on τ.

Case $?\varphi$:

$w[\![!_{()}?\varphi]\!] \sim_a w'[\![!_{()}?\varphi]\!]$
$\Leftrightarrow$ no access is defined on the $\emptyset$-model $M[\![?\varphi]\!]$
false
$\Leftrightarrow$
$w \sim_a w'$ and **false**
$\Leftrightarrow$ $!_{()}?\varphi \sim_a !_{()}?\varphi$ never holds, see definition 40
$w \sim_a w'$ and $!_{()}?\varphi \sim_a !_{()}?\varphi$

Case $L_B\tau$: $w[\![!_{(I)}L_B\tau]\!] \sim_a w'[\![!_{(J)}L_B\tau]\!] \Leftrightarrow w \sim_a w'$ and $!_{(I)}L_B\tau \sim_a !_{(J)}L_B\tau$

'$\Rightarrow$'.
$w[\![!_{(I)}L_B\tau]\!] \sim_a w'[\![!_{(J)}L_B\tau]\!]$
$\Leftrightarrow$ worlds from $M[\![L_B\tau]\!]$ are states (b)
$(M'', w[\![!_I\tau]\!]) \sim_a (M^*, w'[\![!_J\tau]\!])$

We distinghuish case $M'' = M^*$ from case $M'' \neq M^*$. First, $M'' = M^*$:

$(M'', w[\![!_I\tau]\!]) \sim_a (M'', w'[\![!_J\tau]\!])$
$\Leftrightarrow$ definition of local interpretation
$w[\![!_I\tau]\!]) \sim_a w'[\![!_J\tau]\!])$
$\Leftrightarrow$ induction
$w \sim_a w'$ and $!_I\tau \sim_a !_J\tau$
$\Rightarrow$ definition 40
$w \sim_a w'$ and $!_{(I)}L_B\tau \sim_a !_{(J)}L_B\tau$

Then, $M'' \neq M^*$:

$(M'', w[\![!_I\tau]\!]) \sim_a (M^*, w'[\![!_J\tau]\!])$
$\Leftrightarrow$ definition of local interpretation
$w \sim_a w'$ and $a \notin gr(M'') \cup gr(M^*)$
$\Leftrightarrow$
$w \sim_a w'$ and $a \notin ala(!_I\tau) \cup ala(!_J\tau)$
$\Leftrightarrow$ $a \in B$, see definition 40

$w \sim_a w'$ and $!_{(I)}L_B\tau \sim_a !_{(J)}L_B\tau$

'$\Leftarrow$'.

Similar to '$\Rightarrow$'. However, we now have to proceed from cases '$!_I\tau \sim_a !_J\tau$' and '$a \notin ala(!_I\tau) \cup ala(!_J\tau)$'. The second implies that *different* models in $[M[\![\tau]\!]]$ are produced by local interpretation of $!_I\tau$ and $!_J\tau$ in M, respectively; the first implies that they produce the *same* model.

Case $\tau\ ;\ \tau'$:

$w[\![!_{(I,J)}(\tau\ ;\ \tau')]\!] \sim_a w'[\![!_{(K,L)}(\tau\ ;\ \tau')]\!]$
$\Leftrightarrow$ definition of local interpretation
$w[\![!_I\tau]\!][\![!_J\tau']\!] \sim_a w'[\![!_K\tau]\!][\![!_L\tau']\!]$
$\Leftrightarrow$ induction
$w[\![!_I\tau]\!] \sim_a w'[\![!_K\tau]\!]$ and $!_J\tau' \sim_a !_L\tau'$
$\Leftrightarrow$ induction
$w \sim_a w'$ and $!_I\tau \sim_a !_K\tau$ and $!_J\tau' \sim_a !_L\tau'$
$\Leftrightarrow$ definition 40
$w \sim_a w'$ and $!_{(I,J)}(\tau\ ;\ \tau') \sim_a !_{(K,L)}(\tau\ ;\ \tau')$

Case $\tau \cup \tau'$: Suppose $w[\![!_{i(I,J)}(\tau \cup \tau')]\!] \sim_a w'[\![!_{i(K,L)}(\tau \cup \tau')]\!]$. Suppose $i = 0$ ($j = 0$ proceeds similarly). In step (a) of the proof we use that, according to the definition of local interpretation, $w[\![!_{0(I,J)}(\tau \cup \tau')]\!] = w[\![!_I\tau]\!]$.

$w[\![!_{0(I,J)}(\tau \cup \tau')]\!] \sim_a w'[\![!_{0(K,L)}(\tau \cup \tau')]\!]$
$\Leftrightarrow$ (a)
$w[\![!_I\tau]\!] \sim_a w'[\![!_K\tau]\!]$
$\Leftrightarrow$ induction
$w \sim_a w'$ and $!_I\tau \sim_a !_K\tau$
$\Leftrightarrow$ definition 40
$w \sim_a w'$ and $!_{0(I,J)}(\tau \cup \tau') \sim_a !_{0(K,L)}(\tau \cup \tau')$

This concludes the proof of lemma λ. The proof of this lemma concludes the proof of proposition 15. ■

The relation between local and product interpretation has now been sufficiently investigated to allow the description of game actions as knowledge actions. Apart from that, it also seems to be worthy of further investigation.

5.2 Game actions and knowledge actions

Let $\mu = \langle q, Q, r, R, \mathsf{pub}\rangle$ be a game action for a state (M, v). Let $!_I\tau$ be a knowledge action. A world in the game action frame μ^- is a *set of worlds from M* that is a possible answer to the question of the game action, see chapter 2. A world in the knowledge action frame $[\![!_I\tau]\!]^\otimes$ is an *action* $!_J\tau$ of type τ, see definition 42 on page 74. The knowledge action $!_I\tau$ describes the game action μ, if the knowledge action frame and the game action frame are 'the same', i.e. isomorphic, and if in that way actions are matched to answers such that the precondition of an action is satisfied by exactly those worlds in M that the answer consists of.

Definition 46 (Description of a game action by a knowledge action)
Knowledge action $\alpha \in \mathsf{KT}$ *describes* game action μ for state (M, v) if $\mu^- \cong [\![\alpha]\!]^\otimes$ and if, when $\mathcal{I}$ is the isomorphism, $\mathcal{I}(R, \alpha)$ and for all possible answers R' in μ, and for all actions α' of the same type as α:

$$\mathcal{I}(R', \alpha') \Leftrightarrow \forall w \in R' : M, w \models pre(\alpha')$$

Example 41 (Show again)
The knowledge action $\alpha_r = L_{123}(!L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1)$ describes the game action $\mathsf{show}^{2,-}_{1,r}$. E.g., we have that $\mathcal{I}(\{rwb, rbw\}, \alpha_r)$, and $pre(\alpha_r) = r_1$, and indeed $\mathsf{hexa}, rwb \models r_1$.

The constraints on a game action $\mu = \langle q, Q, r, R, \mathsf{pub}\rangle$, for an $\mathbf{A}$ state (M, v), with $|\mathbf{A}| = n$ (see chapter 2), translate into the following constraints on a knowledge action α that describes it:

- access for respondent r on frame $[\![\alpha]\!]^\otimes$ is the identity
- all tests $?\psi$ occurring in the description in $\mathcal{L}_n^\Box$ of a game action are equivalent in M to a disjunction of what the respondent knows:

$$\exists\varphi_1, ..., \varphi_j \in \mathcal{L}_n^\Box : M \models \psi \leftrightarrow \bigvee_{i=1}^{j} K_r\varphi_i$$

- $[\![\alpha]\!]^\otimes$ is an $S5_n$ frame (i.e. an $\mathbf{A}$ frame)

The second constraint is equivalent to saying that $\forall R' \in Q : \exists\varphi_1, ..., \varphi_j \in \mathcal{L}_n^\Box : \forall w \in R' : M, w \models \bigvee_{i=1}^{j} K_r\varphi_i$ (every possible answer R' is characterized in M by a precondition ψ that occurs as a test $?\psi$ in the description). We will show below that all game actions can be described by knowledge actions. However, not all knowledge actions that are executable in a knowledge game state describe game actions:

Example 42
All of the following knowledge actions are executable in state (hexa, rwb), but none of them describe game actions:

- $L_{123}?r_1 \cup !L_{12}?r_1$
  This is not a game action, because the frame for this knowledge action is not an $S5_3$ frame. Point $L_{123}?r_1 \cup !L_{12}?r_1$ of this frame is not reflexive for agent 3.

- $L_{12}?r_1$
  This is not a game action, because the group of this knowledge action is strictly smaller than the group of hexa.

- $L_{123}?(K_1 r_1 \vee K_2 w_2)$
  This is not a game action, because the test is not on the knowledge of a player (more precise: because the interpretation of the test formula is not a union of $\sim_a$ equivalence classes in hexa for some player a).

- $L_{123}(!L_{23}?K_2 w_2 \cup L_{23}?K_3 b_3)$
  This is not a game action, because not all tests are on knowledge of the same player.

- $L_{123}(!L_{12}(!L_1?K_2 w_2 \cup L_1?K_2 b_3) \cup L_{12}(L_2?K_2 w_2 \cup L_2?K_2 b_3))$
  This is not a game action, because there is apparently no respondent, that should be contained in every group that learns.

The game action sorts show, noshow, win and nowin that we encounter in knowledge games, are described by the following knowledge actions. We just give the descriptions, and just for win and show, and leave the comparison with the definitions in chapter 2 to the reader. Proposition win_q describes that q can win (e.g. that he knows the deal of cards):

$$\begin{array}{rcl} \mathsf{win}^q & : & !L_{\mathbf{A}}?\mathsf{win}_q \cup L_{\mathbf{A}}?\neg\mathsf{win}_q \\ \mathsf{show}^{q,\{c^1,\ldots,c^t\}}_{r,c^j} & : & !L_{\mathbf{A}}(!L_{qr}?c^j_r \cup \bigcup_{i\neq j=1}^t L_{qr}?c^i_r) \cup L_{\mathbf{A}}?\bigwedge_{i=1}^t \neg c^i_r \end{array}$$

We generally restrict the description to the publicly accessible alternatives. Thus a show action is described by $L_{\mathbf{A}}(!L_{qr}?c^j_r \cup \bigcup_{i\neq j=1}^t L_{qr}?c^i_r)$ and a win action by $L_{\mathbf{A}}?\mathsf{win}_q$. See also chapter 6.

Procedure 'describe' Given a knowledge action and a game action, we can test whether the knowledge action describes the game action. We can go one step further: given a game action, we can construct a knowledge action that describes it. From the possible answers, we can compute preconditions. Those will be the test formulas in the knowledge actions. From the publicity, we can compute the learning operators that must occur in the knowledge action.

Preconditions Let $s = (M, v)$ be a knowledge game state. As knowledge game states are finite, every world w from s has a (unique) state description $\Delta_{(M,w)}$ in $\mathcal{L}_n^C$, see chapter 3, or [BM96, vB98] for further references. Let W' be a subset of M (of $W = \mathcal{D}(M)$), then $\Delta_{W'} := \bigvee_{w \in W'} \Delta_{(M,w)}$ is the description of W'. We apply this to game actions: to every alternative $R' \in Q$ of the question corresponds a formula $\Delta_{R'}$ that characterizes it: in other words such that $w \in R' \Leftrightarrow M, w \models \Delta_{R'}$. The $\Delta_{R'}$ will be the test formulas in the knowledge actions we are constructing.

From the publicity of the game action, i.e. from the access pub on the game action frame, we compute a hierarchy of learning. This hierarchy determines the learning operators that occur in the knowledge action describing the game action. We give a general definition for frames with equivalence relations:

Definition 47 (Hierarchy of learning)
Let $F = \langle W, (\sim_a)_{a \in \mathbf{A}} \rangle$ be an $S5$ frame. The usual partial order $\leq$ on binary relations (see also appendix A) induces partial orders $\leq_F$ on $\mathcal{P}(\mathbf{A})$ and $\preceq_F$ on a subset of $\mathcal{P}(\mathbf{A})$:

$$
\begin{array}{lrcl}
\forall A, B \subset \mathbf{A}: & A \leq_F B & \Leftrightarrow & \sim_A \leq \sim_B \\
 & A =_F B & \Leftrightarrow & A \leq_F B \text{ and } B \leq_F A \\
 & [A]_{=_F} \preceq_F [B]_{=_F} & \Leftrightarrow & A \leq_F B \\
 & & & \\
 & \mathcal{P}(\mathbf{A})_F & = & \{A \subseteq \mathbf{A} \mid \forall B \subseteq \mathbf{A} : B \in [A]_{=_F} \Rightarrow B \subseteq A\} \\
 & & & \\
\forall A, B \in \mathcal{P}(\mathbf{A})_F: & A \preceq_F B & \Leftrightarrow & [A]_{=_F} \preceq_F [B]_{=_F} \\
 & A \prec_F B & \Leftrightarrow & A \preceq_F B \text{ and } B \neq A \\
 & A \prec_F^1 B & \Leftrightarrow & A \prec_F B \text{ and } \neg\exists C : A \prec_F C \text{ and } C \prec_F B
\end{array}
$$

Relation $=_F$ is an equivalence relation that induces a partition on $\mathcal{P}(\mathbf{A})$; the equivalence class containing A is $[A]_{=_F}$. Every $=_F$ equivalence class has a top element. We let every $=_F$ equivalence class be represented by its top element. We then 'downgrade' relation $\leq_F$ to the partial order $\preceq_F$ on the set of 'representatives' of $=_F$ equivalence classes, and make it a strict order $\prec_F$. Note that $A \prec_F B \Rightarrow A \subset B$.[2] The bottom of this order $\prec_F$ is $\emptyset$. The top of this order is $\mathbf{A}$. We then derive a functional 'successor relation' $\prec_F^1$.

The order $\prec_F$ precisely reflects what different groups learn in a knowledge action type with F as frame: for any meaningful[3] subprogram $L_A\tau$ in that action type, A is a 'threshold of learning': for any subprogram $L_B\tau'$ of that type that contains $L_A\tau$, $A \prec_F^1 B$. For any subprogram $L_B\tau'$ of that type that $L_A\tau$ contains, $B \prec_F^1 A$. In other words: $A \prec_F B$ means 'A learns more than B', and $A \prec_F^1 B$ means 'A are the first to learn more than B'.

[2] We have that $\sim_A \leq \sim_B \Leftrightarrow (\bigcup_{a \in A} \sim_a)^* \subseteq (\bigcup_{a \in B} \sim_a)^*$ and that B is the largest group defining $\sim_B$.

[3] We may define 'meaningful' as follows: if $L_A\tau$ is part of τ', then $[\![\tau'[L_A\tau := \tau]]\!] \neq [\![\tau']\!]$

For game actions μ, with F the frame underlying pointed frame μ^-, we may write $\prec_\mu$ for $\prec_F$. The broadcast unit Br of the game action is the smallest nonempty subset of $\mathbf{A}$ in the $\prec_\mu$ order: $\emptyset \prec^1_\mu Br$.

The procedure $\mathsf{describe}$ maps game actions to knowledge actions, and uses the order $\prec_\mu$:

Definition 48 (Procedure $\mathsf{describe}$)
Let $\mu = \langle q, Q, r, R, \mathsf{pub}\rangle$ be a game action. Then:

$$\forall A, A' \preceq_\mu \mathbf{A} : \forall Q' \subseteq Q : \forall R' \in Q' :$$

$$\begin{array}{rcl}
\mathsf{describe}(A, Q', \mu) & = & !L_A(\bigcup_{A' \prec^1_\mu A} \mathsf{describe}(A', Q' \lceil [R]_{\mathsf{pub}_A}, \mu) \cup \\
& & \bigcup^{R' \neq R}_{[R']_{\mathsf{pub}_A} \subseteq Q'} L_A(\bigcup_{A' \prec^1_\mu A} \mathsf{describe}(A', Q' \lceil [R']_{\mathsf{pub}_A}, \mu) \\
\mathsf{describe}(\emptyset, Q', \mu) & = & ?\bigvee_{R' \in Q'} K_r \Delta_{R'} \\
\mathsf{describe}(\mu) & = & \mathsf{describe}(\mathbf{A}, Q, \mu)
\end{array}$$

In the definition, by $\bigcup^{R' \neq R}_{[R']_{\mathsf{pub}_A} \subseteq Q'}$ we mean nondeterministic choice between subprograms for all pub_A equivalence classes that are different from each other, different from $[R]_{\mathsf{pub}_A}$, and contained in Q'. If empty, that part of the description is deleted. Concerning $?\bigvee_{R' \in Q'} K_r \Delta_{R'}$, we remind the reader that every possible answer corresponds to a union of equivalence classes of the respondent r to the question. As we are in $S5$, we may as well write $\Delta_{R'}$ instead of $K_r \Delta_{R'}$. Further, note that we do *not* need sequential execution at all to describe a game action.[4]

Fact 6
Let μ be a game action for state (M, w). Then $\mathsf{describe}(\mu) \in \mathsf{KA}$.

Obvious.

Example 43 (Computing a knowledge action for 1 showing red to 2)
We apply the procedure to the game action $\mathsf{show}^{2,-}_{1,r}$. For this show action, the learning hierarchy is $\emptyset \prec^1 \{1,2\} \prec^1 \{1,2,3\}$:

$\mathsf{describe}(\mathsf{show}^{2,-}_{1,r})$
$=$
$\mathsf{describe}(\{1,2,3\}, \{\{rwb, rbw\}, \{wrb, wbr\}, \{brw, bwr\}\}, \mathsf{show}^{2,-}_{1,r})$
$=$
$L_{123}(\mathsf{describe}(\{1,2\}, \{\{rwb, rbw\}, \{wrb, wbr\}, \{brw, bwr\}\}, \mathsf{show}^{2,-}_{1,r}))$
$=$ the 3 possible answers are the show_{12} equiv. classes
$L_{123}(!L_{12}(\mathsf{describe}(\emptyset, \{\{rwb, rbw\}\}, \mathsf{show}^{2,-}_{1,r})) \cup$
$L_{12}(\mathsf{describe}(\emptyset, \{\{wrb, wbr\}\}, \mathsf{show}^{2,-}_{1,r})) \cup L_{12}(\mathsf{describe}(\emptyset, \{brw, bwr\}\}, \mathsf{show}^{2,-}_{1,r})))$

[4]But as we do not have an algebra of knowledge actions, we cannot generalize this remark.

$=$
$L_{123}(!L_{12}?\Delta_{\{rwb,rbw\}} \cup L_{12}?\Delta_{\{wrb,wbr\}} \cup L_{12}?\Delta_{\{brw,bwr\}})$

We have that $\Delta_{\{rwb,rbw\}}$ is equivalent to $(\delta_{rwb} \vee \delta_{rbw}) \wedge C_{123}\mathsf{33}$, and because $\mathsf{hexa} \models ((\delta_{rwb} \vee \delta_{rbw}) \wedge C_{123}\mathsf{33}) \leftrightarrow r_1$. Similarly for the descriptions of the other possible answers. Therefore knowledge action $\mathsf{describe}(\mathsf{show}^{2,-}_{1,r})$ has the same interpretation as the familiar action $L_{123}(!L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1)$ on hexa, as is to be expected.

We conjecture the following proposition, and suggest a proof by induction on $\prec_\mu$.

Proposition 16
Let μ be a game action. Then $\mathsf{describe}(\mu)$ describes game action μ.

5.3 Complexity

We can measure the size of an $S5$ state by the number of worlds of its underlying model. Preferably, the model is minimal in the sense that it is not bisimilar to a smaller one, but we do not demand that. By executing a knowledge action α in a state s, we get a new state $s[\![\alpha]\!]$. What is the size of $s[\![\alpha]\!]$? Typical for knowledge actions is that the size of $s[\![\alpha]\!]$ can be larger than the size of s, namely when agents have real choice in that action. An example in (hexa, rwb) is the action where player 1 whispers in the ear of player 2 that he does not have the white card ($\mathsf{whisper}$). The resulting model consists of 12 worlds, see chapter 4.

In *general*, the size of $s[\![\alpha]\!]$ is exponential in the length of the action α, as measured by its operator depth.[5] This suggests intractible computations. We do not want to give an in-depth treatment of these computations here. But in *fact*, instead of exponentially growing, states may shrink, even when choice is involved.

We were motivated to look into the complexity of action execution, because we didn't understand why it was possible *at all* to play Cluedo. As this game consists of a long sequence of card showing actions, shouldn't we expect the state space to blow up? So why is it fun to play the game? Apart from other reasons, this is also because Cluedo game states can actually get smaller, even though we expect them to get larger because the players can choose. Why? In this section we *only* answer that question. We continue with a definition.

Definition 49 (Size of models and states)
Let (M, w) be an $S5$ state, where $M = \langle W, (\sim_a)_{a\in\mathbf{A}}, V\rangle$. Then:

size of model M	$\lvert M\rvert$	$=$	$\lvert\mathcal{D}(M)\rvert$
size of state (M, w)	$\lvert(M, w)\rvert$	$=$	$\lvert M\rvert$
relevant size of state (M, w)	$\lvert\lvert(M, w)\rvert\rvert$	$=$	$\lvert[w]_{\mathbf{A}}\rvert$

[5] We use that $\frac{|s[\![\alpha;\alpha']\!]|}{|s|} = \frac{|s[\![\alpha]\!]|}{|s|} \cdot \frac{|s[\![\alpha;\alpha']\!]|}{|s[\![\alpha]\!]|} = \frac{|s[\![\alpha]\!]|}{|s|} \cdot \frac{|(s[\![\alpha]\!])[\![\alpha']\!]|}{|s[\![\alpha]\!]|}$

The relevant size of a state is the number of worlds that are relevant ($\sim_{\mathbf{A}}$-accessible) to the players: the number of states that a player has to consider in his reasoning.

We now compute the effect of game actions in initial knowledge games states. By definition, initial knowledge game states are minimal (in the bisimulation contraction sense). All actions where the publicitiy is the identity on the question, result in a reduction of the size of the state they are executed in. They are described by knowledge actions of the form $!L_{\mathbf{A}}?\varphi \cup \tau$. Because $||s[\![!L_{\mathbf{A}}?\varphi \cup \tau]\!]|| = ||s[\![L_{\mathbf{A}}?\varphi]\!]||$, these actions all correspond to public announcements. The public announcement of φ, restricts a state to the worlds that satisfy φ. Therefore, game actions win and nowin reduce the size of a state, as they are public announcements. For knowledge games, the only candidates for combinatorial explosion are the action sorts show and noshow. Because only the *relevant* size of a state is of interest, the sort noshow also results in a reduction: no agent in **A** has alternatives to this action of type $!L_{\mathbf{A}}?\varphi \cup \tau$.

We now compute the size of the initial state of a knowledge game, and the change in size as a result of executing a show action.

Size of the initial state of a knowledge game Let $d \in \mathbf{A}^{\mathbf{C}}$. Let $|\mathbf{A}| = n$ and $|\mathbf{C}| = m$. Write $\sharp i$ for $|d^{-1}(i)|$. Except for boundary cases, the set $D_{\sharp d}$ where all players hold the same number of cards as in d is the set of relevant card deals. It is the domain of the initial state si_d of a knowledge game for d (see chapter 2). The total number of relevant card deals can be computed as follows: First player 1 has to draw $\sharp 1$ cards from the stack of m cards. There are $\binom{m}{\sharp 1}$ ways to do that. Then player 2 has to draw $\sharp 2$ cards from the stack of $m - \sharp 1$ cards. There are $\binom{m-\sharp 1}{\sharp 2}$ ways to do that. And so on. So we have that:

$$|si_d| = |I_d| = |D_d| = \prod_{i=1}^{n} \binom{m - \sum_{j<i} \sharp j}{\sharp i} = \frac{m!}{\prod_{i=1}^{n}(\sharp i)!}$$

Show The game action of player r showing a card c^i (one of t cards) to player q is described by the knowledge action $L_{\mathbf{A}}(!L_{qr}?c_r^i \cup \bigcup_{j\neq i=1}^{t} L_{qr}?c_r^j)$. The size of $si_d[\![L_{\mathbf{A}}(!L_{qr}?c_r^i \cup \bigcup_{j\neq i=1}^{t} L_{qr}?c_r^j)]\!]$ is computed as follows:

$|si_d[\![L_{\mathbf{A}}(!L_{qr}?c_r^i \cup \bigcup_{j\neq i=1}^{t} L_{qr}?c_r^j)]\!]|$
$=$
$|I_d[\![L_{\mathbf{A}}(\bigcup_{j=1}^{t} L_{qr}?c_r^j)]\!]|$
$=$
$\sum_{j=1}^{t} |I_d[\![L_{qr}?c_r^j]\!]|$
$=$

$\sum_{j=1}^{t} |I_d[\![?c_r^j]\!]|$
$=$
$\sum_{j=1}^{t} |\{w \in I_d \mid I_d, w \models c_r^j\}|$
$=$
$t \cdot |I_d| \cdot \frac{\sharp r}{m}$
$=$
$|si_d| \cdot \frac{t \cdot \sharp r}{m}$

Therefore, if $\frac{t \cdot \sharp r}{m} < 1$, then $|si_d[\![\mathsf{show}]\!]| < |si_d|$: the size of the next game state is smaller than the size of the initial game state. In Cluedo every player holds three cards and a request is for one out of three cards (namely, for a combination of a murder, a room and a weapon card). So if the first action in a Cluedo game is a show action, the size of the next game state is only $\frac{9}{21}$th of the size of the initial game state. We do not know whether $|s[\![\mathsf{show}]\!]| \leq |s|$ for every Cluedo game state s, but we conjecture that it keeps shrinking. We already mentioned that noshow, win and nowin actions also result in a reduction of the size of the game state. This explains why it is feasible to play Cluedo: there are less possibilities all the time![6] This concludes our short excursion into the complexity of executing knowledge actions.

5.4 Conclusions

We introduced the notion of product interpretation. We have proven where product interpretation and local interpretation are the same. By means of product interpretation we defined when a knowledge action describes a game action. We suggested a procedure to compute a knowledge action from a given game action. The result of an initial show action in Cluedo results in a decrease of the size of the game state.

[6] We do not know the reduction due to a nowin action. We also computed the size reduction of the initial game state due to noshow: noshow is decribed by $L_{\mathbf{A}}?\bigwedge_{j=1}^{t} \neg c_r^j$. We now have that $si_d[\![L_{\mathbf{A}}?\bigwedge_{j=1}^{t} \neg c_r^j]\!] = |si_d[\![L_{\mathbf{A}}?\bigwedge_{j=1}^{t} \neg c_r^j]\!]| = |I_d[\![L_{\mathbf{A}}?\bigwedge_{j=1}^{t} \neg c_r^j]\!]| = |I_d[\![?\bigwedge_{j=1}^{t} \neg c_r^j]\!]|$. The size of $I_d[\![?\bigwedge_{j=1}^{t} \neg c_r^j]\!]$ is the number of worlds satisfying that test. The probability that a player does not have any of the cards $c^1, ..., c^t$ is the probability that that player's first card is not any of those t, i.e. $\frac{m-t}{m}$, multiplied with the probability that his second card isn't, i.e. $\frac{m-1-t}{m-1}$, and so on until $\frac{m-(\sharp r-1)-t}{m-(\sharp r-1)}$. Therefore the size reduction is

$$\frac{\binom{m-r}{t}}{\binom{m}{t}}$$

Chapter 6

Examples

In chapters 2 and 4 we gave some examples of knowledge game states. We now give many other examples. In section 6.1 we describe various actions in a language for only two agents and one atom. The context is better known under the name 'nightclub or lecture'. In section 6.2 we give an overview of all game actions and all game states in the hexa game, and we discuss other knowledge actions in that domain. We pay some attention to the topic of unsuccessful updates, because of its relevance to the analysis of games. In section 6.3 we visualize the effect of a player choosing between different cards to show. In section 6.4 we present the knowledge actions that describe game actions in Cluedo. In section 6.5 we discuss the concept of 'suspicion'. In section 6.6 we describe the exchange of secrets over a network, better known as 'spreading gossip'.

6.1 Nightclub or lecture

Anne and Bert are sitting at a table, having coffee. A messenger comes in and delivers an urgent message in an envelope, to Anne. The letter contains either an invitation for a night out in Amsterdam, or an obligation to give a lecture instead. Anne and Bert commonly know that these are the only alternatives. This situation can be modelled as follows: There is one atom p, describing 'the letter contains an invitation for a night out in Amsterdam', so that $\neg p$ stands for the lecture obligation. There are two agents 1 (Anne) and 2 (Bert). Letter is the model $\langle \{w, w'\}, \{\sim_1, \sim_2\}, V \rangle$ with both $\sim_1$ and $\sim_2$ the universal relation on $\{w, w'\}$, and with $V_w(p) = 1, V_{w'}(p) = 0$. Below we list some types of action that are executable in letter, one can imagine actions of those types being executed in states of letter. Figure 6.1 pictures the models resulting from their execution. In the figure we name the worlds w and w' by their atomic descriptions p and $\neg p$, respectively.

- Anne is invited for a night out in Amsterdam and reads the letter *aloud*: $L_{12}?p$

- Bert is seeing that Anne reads the letter (and this is publicly known): $L_{12}(L_1?p \cup L_1?\neg p)$

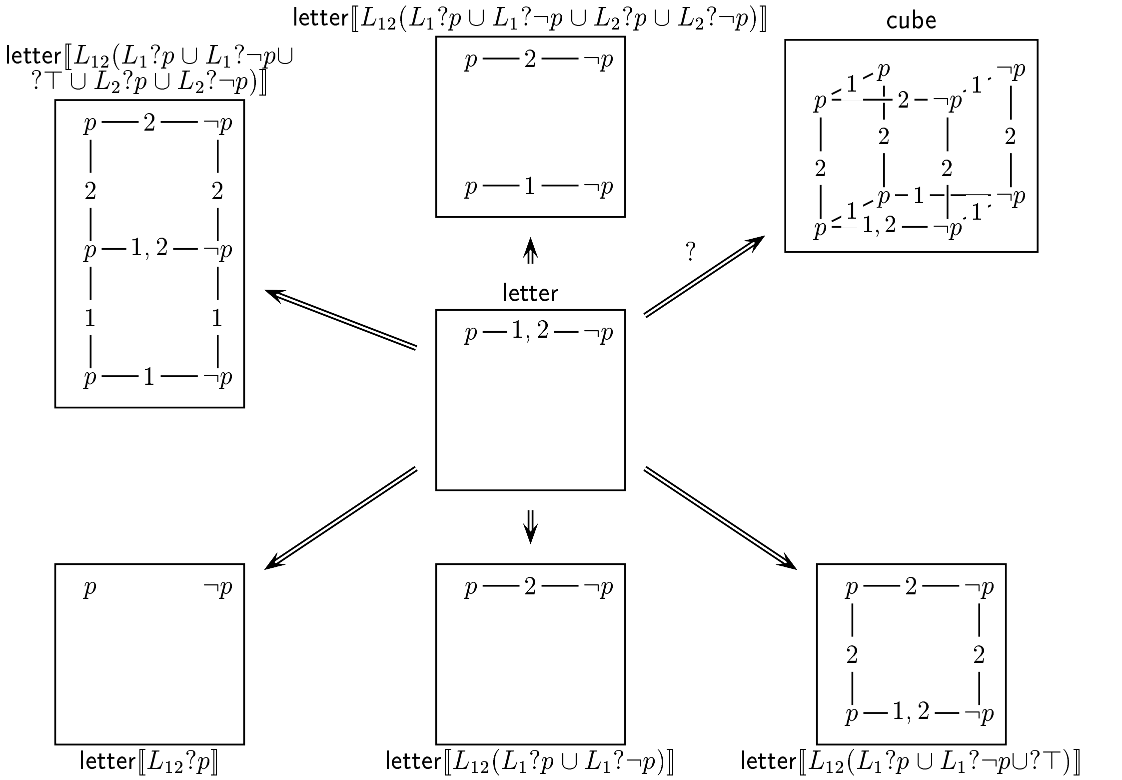


Figure 6.1: Some models for two agents and one atom. Assume transitivity of access.

- Bert suspects that Anne has read the letter (and this is publicly known): $L_{12}(L_1?p \cup L_1?\neg p\cup\ ?\top)$
- An outsider tells Anne and Bert that one of them has read the letter: $L_{12}(L_1?p \cup L_1?\neg p \cup L_2?p \cup L_2?\neg p)$
- An outsider tells Anne and Bert that one of them may have read the letter: $L_{12}(L_1?p \cup L_1?\neg p\cup\ ?\top \cup L_2?p \cup L_2?\neg p)$
- An outsider tells the agents that some of them may have read the letter. (By 'some' we mean 0, 1 or 2.) The resulting model is cube.

Two players each holding one card The model letter can also be seen as the pre-initial state of a knowledge game for two players each holding one card. Suppose the cards are white and black. There are two possible deals, $b|w$ and $w|b$.[1] The two deals $b|w, w|b$ correspond to the two worlds w, w', or, more figuratively, the invitation for a night out in Amsterdam corresponds to the black card, and the lecture obligation to the white card. There are four atomic propositions: b_1, w_1, b_2, w_2, where b_1 stands for '1 holds black', etc., and we have that $b_1 \wedge w_2$ corresponds to p. Agent 1 reading the letter corresponds to agent 1 turning her

[1] $b|w$ is the deal $d \in \{1,2\}^{\{b,w\}}$ with $d(b) = 1$ and $d(w) = 1$. See chapter 2.

card. Agent 1 reading the letter aloud corresponds to both players turning cards. $\mathsf{Letter}[\![L_{12}?p]\!]$, bottom left in figure 6.1, corresponds to the initial model for this knowledge game.

How ignorant can one be? In letter ignorance is somehow public. Although both players can imagine all deals of cards (i) (of that size, i.e. where they each hold one card), both players know also that this holds as well for the other player (ii). From a game point of view, a player is worse off when (i) holds but (ii) not. In $\mathsf{letter}[\![L_{12}(L_1?p \cup L_1?\neg p \cup\ ?\top \cup L_2?p \cup L_2?\neg p)]\!]$, top left in figure 6.1, they can be worse off in that way. In either of the two middle worlds in that model, both players do not know whether p, but they also do not know whether the other player knows p: they even can imagine the other player to know either p or $\neg p$ or nothing: their uncertainty about the deals of cards (i.e. valuations) that are still considered by the other player, is maximal.

In cube, this idea is carried to its full extent: in any world of cube it holds, that whatever an agent knows of the atomic description of the world, he can imagine the other agent to know *anything* more or less of it. We think this kind of model merits further attention: we conjecture that any model constructed by executing a KT action in letter is a restriction of cube, in other words: all knowledge actions can also be interpreted as *public announcements* in cube. We have not further investigated that. The model cube itself is *not* the result of executing a KT action type in letter, for that, we would need an operation $\cap$ of simultaneous execution, see chapter 4.[2].

6.2 Everything on three players and three cards

The hexa game has been a motivating example for every part of this research. The knowledge state $(\mathsf{hexa}, rwb) = (\langle\{rwb, rbw, wbr, wrb, brw, bwr\}, \{\sim_1, \sim_2, \sim_3\}, V\rangle, rwb)$ represents the knowledge of three players each holding one card, for actual deal of cards rwb. For all deals d, e in the domain of hexa: $d \sim_a e \Leftrightarrow d^{-1}(a) = e^{-1}(a)$, and $V_d(c_a) = 1 \Leftrightarrow d(c) = a$. See the middle picture in figure 6.2. The designated state rwb is indicated by a sans serif typeface. We give an overview of game actions and corresponding knowledge actions for hexa games.

In chapter 2 we described four different sorts of legal game actions in knowledge games: show, noshow, win and nowin. As in hexa a player can ask for either 1, 2 or 3 cards, the following seven sorts of action can be executed in hexa games. They are accompanied by the knowledge actions that describe them, their meaning will be clear. As in hexa winning is knowing the deal of cards, $\mathsf{win}_a = \bigvee_{d \in \mathcal{D}(\mathsf{hexa})} K_a \delta_d$.

[2]Its description would then be: $L_{12}(L_1?p \cup L_1?\neg p \cup ?\top \cup L_2?p \cup L_2?\neg p \cup (L_1?p \cap L_2?p) \cup (L_1?\neg p \cap L_2?\neg p))$

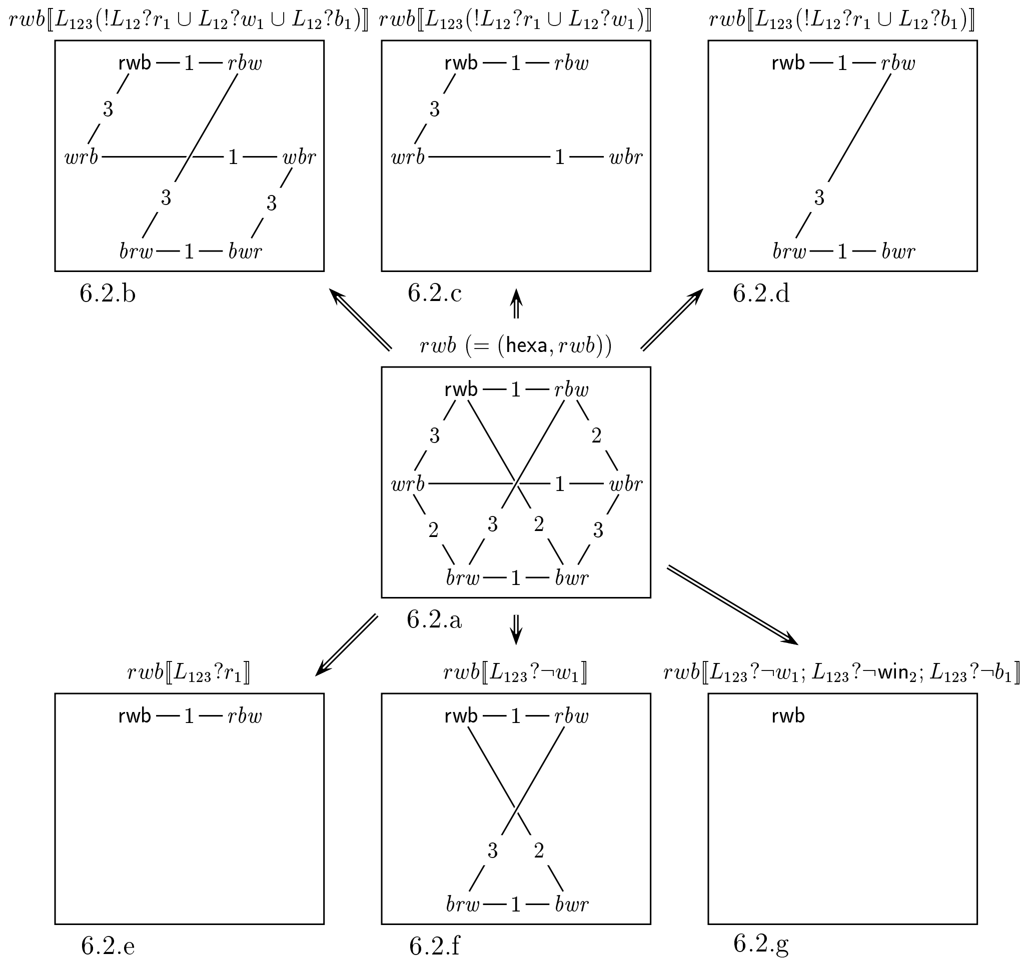


Figure 6.2: Overview of all non-agent-similar game states for hexa games

$\mathsf{show}_{b,c}^{a,-}$	$L_{123}(!L_{ab}?c_b \cup L_{ab}?c'_b \cup L_{ab}?c''_b)$
$\mathsf{show}_{b,c}^{a,\{c,c'\}}$	$L_{123}(!L_{ab}?c_b \cup L_{ab}?c'_b)$
$\mathsf{show}_{b,c}^{a,c}$	$L_{123}?c_b$
$\mathsf{noshow}_{b}^{a,\{c,c'\}}$	$L_{123}?(\neg c_b \wedge \neg c'_b)$
$\mathsf{noshow}_{b}^{a,c}$	$L_{123}?\neg c_b$
win^{a}	$L_{123}?\mathsf{win}_a$
nowin^{a}	$L_{123}?\neg\mathsf{win}_a$

Agent-similarity More interesting for a modal logician than this overview of game actions, is an overview of $S5$ states that can be produced by action sequences. Such an overview can be rather concise, if we only look at frame properties of the models. An example: model rwb—2—bwr can be transformed into model rwb—1—rbw by switching the roles of players 1 and 2, and by mapping worlds properly; i.e.: by the permutation (213) of the sequence $(1, 2, 3)$ of players, and a bijection $\mathcal{I}$ on the set of deals $\{rwb, bwr\}$ of that model: $\mathcal{I}(rwb) = rwb$ and $\mathcal{I}(bwr) = rbw$. The bijection $\mathcal{I}$ is somewhat like a frame isomorphism (see appendix A). Multiagent models / states are 'agent-similar' if they are the same with respect to such a permutation and bijection:

Definition 50 (Agent-similar)
Let $\mathbf{A}$ be a set of agents, let $\mathbf{C}$ be a set of cards. Let $\mathbf{P} = \mathbf{C} \times \mathbf{A}$ be the set of atoms. Let $M = \langle W, (\sim_a)_{a \in \mathbf{A}}, V \rangle$ and $M' = \langle W', (\sim'_a)_{a \in \mathbf{A}}, V' \rangle$ be two $S5$ models for $\mathbf{A}$ and $\mathbf{P}$. Then M and M' are agent-similar, notation $M \equiv_{\mathbf{A}} M'$, iff there is a permutation f of the set of agents $\mathbf{A}$ and a bijective relation $\mathcal{I}$ between M and M' such that:

$$\forall w, w'' \in W : w \sim_a w'' \Leftrightarrow \mathcal{I}(w) \sim_{f(a)} \mathcal{I}(w'')$$

Two $S5$ states (M, w) and (M', w') are agent-similar, iff M and M' are agent-similar and w' is the image of w given the permutation establishing the similarity.

There are only seven not agent-similar states and six not agent-similar models for hexa. Figure 6.2 pictures those states. Note that the *models* underlying 6.2.c and 6.2.d are agent-similar, although these *states* aren't, because different points have been chosen in the model. It will be clear what questions by 2 that are answered by 1 are described by the knowledge actions in the figure, except for 6.2.g (see below).

The state in figure 6.2.f, that is reached by game action $\mathsf{noshow}_1^{2,w}$ and is described by the knowledge action $L_{123}?\neg w_1$, is of further interest. This is because of the continuation of this game: Player 2 cannot win in this state, as he cannot distinguish actual deal rwb from deal bwr. By, therefore, passing his move to the next player, player 2 implicity announces that he cannot win (yet). The result of executing this action delivers the state $(\mathsf{hexa}, rwb)[\![L_{123}?\neg w_1]\!][\![L_{123}?\neg \mathsf{win}_2]\!]$. The resulting state rwb—2—bwr is $\equiv_{123}$-similar to figure 6.2.e.

If now player 1 or player 3 were to move, that player simply announces his knowledge and wins, without the need for any further requests to the other players. The announcement is described by the knowledge action $L_{123}?\mathsf{win}_1$ or $L_{123}?\mathsf{win}_3$, respectively. In neither case does the execution of the action change the state of the game: after the announcement, 2 *still* cannot distinguish between rwb and bwr.

Now suppose, instead of 1 and 3 winning, that 2 is allowed a second chance. He asks 1 for the blue card; such that 1 again responds by saying 'no'. This interaction is described by the knowledge action $L_{123}?\neg b_1$. The resulting game state is pictured in 6.2.g.

Figure 6.2.g models the singleton state rwb where there is public knowledge of the deal of cards. This state can be reached from (hexa, rwb) by the game action sequence $(\mathsf{noshow}_1^{2,w}, \mathsf{nowin}^2, \mathsf{noshow}_1^{2,b})$, see the previous paragraph. It can only be reached by such a clearly suboptimal action sequence. A direct way to produce

it is by executing knowledge action $L_{123}?\delta_{rwb}$, i.e. by a public announcement that the deal of cards is *rwb*. This, however, is not a *game* action.

6.2.1 Other actions for three players and three cards

We continue with some other examples of knowledge states and knowledge actions for three players each holding a card. The examples in the list below are pictured in figure 6.3. In the figure, assume access to be transitive. The names of the worlds are the deals that atomically characterize them, so that some worlds have the same name. They can still be distinguished from each other because they have different access to other worlds.

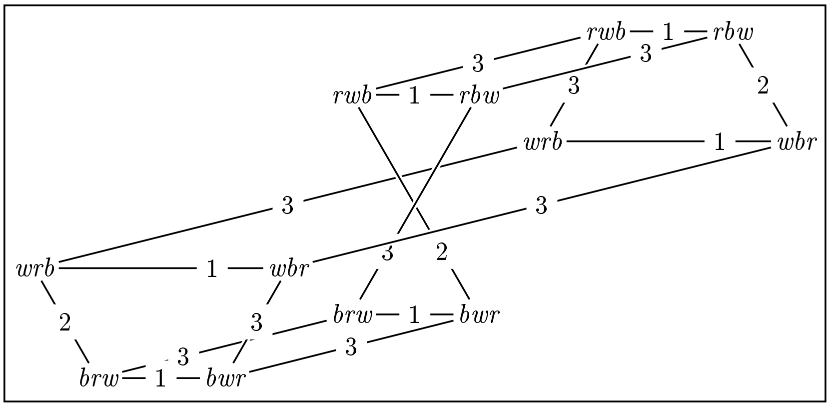
1. In the initial model hexa, player 1 whispers in 2's ear a card that he (1) doesn't have: $L_{123}(L_{12}?\neg r_1 \cup L_{12}?\neg w_1 \cup L_{12}?\neg b_1)$
2. Players $1, 2, 3$ play hexa knowledge games. There is a fourth player present, who doesn't hold a card or otherwise interact with the game. He gets bored and leaves. When he returns, player 2 tells him that he has asked player 1 for one of two cards and that player 1 has responded to his request: $L_{1234}(L_{123}(L_{12}?w_1 \cup L_{12}?b_1) \cup L_{123}(L_{12}?r_1 \cup L_{12}?b_1) \cup L_{123}(L_{12}?r_1 \cup L_{12}?w_1) \cup L_{123}?r_1 \cup L_{123}?w_1 \cup L_{123}?b_1)$
3. The 'pre-initial' model where the three cards have been dealt, but nobody has looked at his card.
4. In the pre-initial model, 1 looks at his card: $L_{123}(L_1?r_1 \cup L_1?w_1 \cup L_1?b_1)$
5. In the pre-initial model, all players look at their cards: look_{123}

The 'whispering' action executed in figure 6.3.1 has been presented in detail in chapter 4. Note that the model of figure 6.3.2 consists of 18 nonsimilar worlds and that, whatever the actual state of the game, player 4 can imagine *nine* different actions to have taken place: Player 2 can have asked three different questions, namely the three combinations of two cards out of three. To each question player 1 can have responded in three different ways: showing one card, showing the other card, and saying that he doesn't have either of the two cards. The pre-initial model in figure 6.3.3 has been described in chapter 3. In general, the knowledge action type $\mathsf{look}_{\mathbf{A}}$, as look_{123} in figure 6.3.4, is defined as follows:

Suppose the actual deal of cards is $d \in \mathbf{A}^{\mathbf{C}}$. Let $n = |\mathbf{A}|$. The action 'player a looks at his cards' corresponds to the action 'player a learns his atomic state'. This is expressed by the action type $\mathsf{look}(a)$. We define:

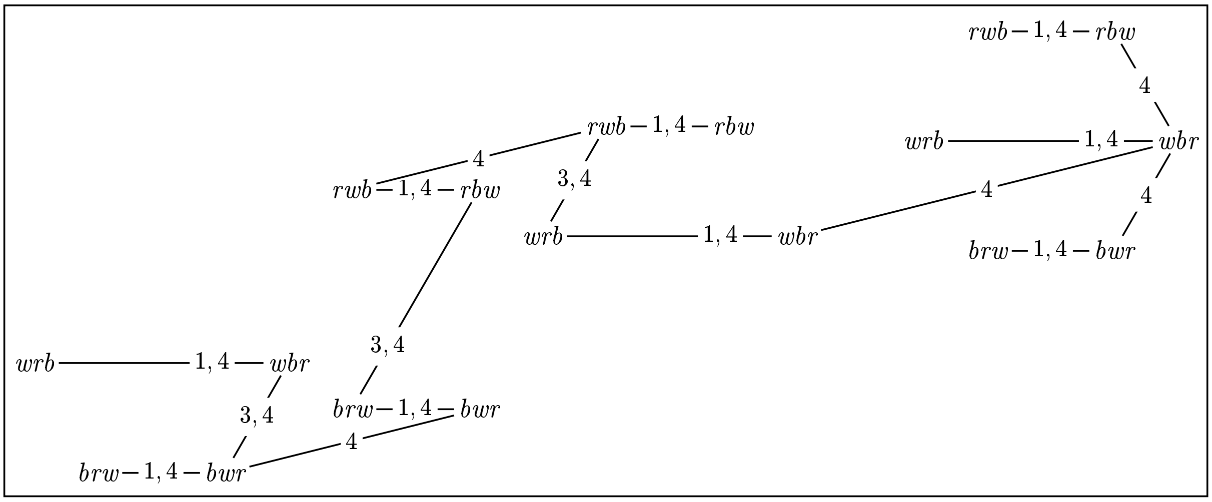
$$\begin{array}{rcl} \mathsf{look}(a) & := & L_{\mathbf{A}}(\bigcup_{d' \in D_d} L_a?\delta^a_{d'}) \\ \mathsf{look}((a_1, ..., a_n)) & := & \mathsf{look}(a_1) \ ; \ ... \ ; \ \mathsf{look}(a_n) \\ \mathsf{look}_{\mathbf{A}} & := & \bigcup_{a_1 \neq ... \neq a_n \in \mathbf{A}} \mathsf{look}((a_1, ..., a_n)) \end{array}$$

$\mathsf{hexa}[\![L_{123}(L_{12}?\neg r_1 \cup L_{12}?\neg w_1 \cup L_{12}?\neg b_1)]\!]$



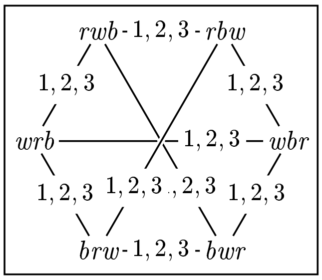
6.3.1

$\mathsf{hexa}[\![L_{1234}(L_{123}(L_{12}?w_1 \cup L_{12}?b_1) \cup L_{123}(L_{12}?r_1 \cup L_{12}?b_1) \cup L_{123}(L_{12}?r_1 \cup L_{12}?w_1) \cup L_{123}?r_1 \cup L_{123}?w_1 \cup L_{123}?b_1)]\!]$



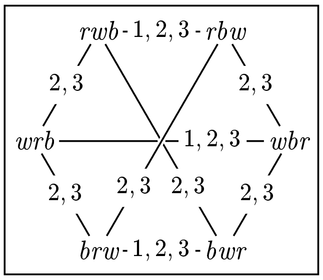
6.3.2

hexa0



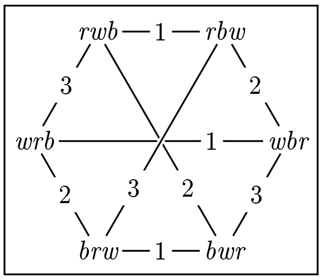
6.3.3

$\mathsf{hexa0}[\![L_{123}(L_1?r_1 \cup L_1?w_1 \cup L_1?b_1)]\!]$



6.3.4

$\mathsf{hexa0}[\![\mathsf{look}_{123}]\!] = \mathsf{hexa}$



6.3.5

Figure 6.3: Knowledge states for three agents and three cards

It can be proven (omitted) that the interpretation of action type $\mathsf{look}_\mathbf{A}$ is functional on the model $preI_d$. So we can write $preI_d[\![\mathsf{look}_\mathbf{A}]\!]$. Also, it can be shown that $preI_d[\![\mathsf{look}_\mathbf{A}]\!] = I_d$. In chapter 3, this provided us with an indirect proof of the uniqueness of an initial model I_d, such as hexa.

6.2.2 Unsuccessful updates

Consider again the two plays of the hexa game ($\mathsf{noshow}_1^{2,w}$, nowin^2, win^1) and ($\mathsf{noshow}_1^{2,w}$, nowin^2, win^3) (see figure 6.2.f). The point here is, that players 1 or 3 *could win* after player 2 told them he *could not win*. We are touching on a topic described in the literature ([Ger99]) as the *unsuccessful update*. An unsuccessful update is a KA knowledge action $L_A?\varphi$, i.e. an announcement to group A, for which there is a knowledge state (M, w) such that $(M, w)[\![L_A?\varphi]\!] \not\models \varphi$. Formula φ is called the unsuccessful update formula.[3] We give two examples, and a conjecture of game theoretical interest:

Example 44 (Unsuccessful update)
A simple example: in the initial state of the game for three players and three cards, with actual deal rwb, player 1 says to other players: "You don't know that I have the red card.". We have to be more precise: player 1 is implicating "I have the red card and both of you don't know that." The action executed is $L_{123}?K_1(r_1 \wedge \neg K_2 r_1 \wedge \neg K_3 r_1)$. We have that:

$$\begin{array}{rcl} (\mathsf{hexa}, rbw) & \models & K_1(r_1 \wedge \neg K_2 r_1 \wedge \neg K_3 r_1) \\ (\mathsf{hexa}, rwb)[\![L_{123}?K_1(r_1 \wedge \neg K_2 r_1 \wedge \neg K_3 r_1)]\!] & \models & C_{123}?r_1 \\ (\mathsf{hexa}, rwb)[\![L_{123}?K_1(r_1 \wedge \neg K_2 r_1 \wedge \neg K_3 r_1)]\!] & \not\models & K_1(r_1 \wedge \neg K_2 r_1 \wedge \neg K_3 r_1) \end{array}$$

The resulting state rwb—1—rbw is that of figure 6.2.e, i.e. $(\mathsf{hexa}, rwb)[\![L_{123}?r_1]\!]$.

Example 45 (Unsuccessful update)
In state $(\mathsf{hexa}, rwb)[\![L_{123}?\neg w_1]\!]$, pictured by figure 6.2.f, player 2, instead of passing his move to the next player, mumbles, "Gosh, somebody can already win now". Player 2 seems to be implicating the following: *Another* player can win. Let us, for the sake of the example, be even more strict (though not conversationally implicated) and assume that he is actually saying: '*One* other player can win'. The proposition describing it is

$$K_2\mathsf{winother} := K_2((\mathsf{win}_1 \wedge \neg\mathsf{win}_3) \vee (\mathsf{win}_3 \wedge \neg\mathsf{win}_1))$$

When executing $L_{123}?K_2\mathsf{winother}$, only the worlds survive where 2 knows that either 1 or 3 can win; in worlds rwb and bwr 2 knows that 3 can win; 1 cannot win in any of the four worlds. Therefore, the state resulting from executing

[3] Actually, the form in [Ger99] generalizes to $M[\![L_A?\varphi]\!] \not\models C_A\varphi$.

$L_{123}?K_2\mathsf{winother}$ in $(\mathsf{hexa}, rwb)[\![L_{123}?\neg w_1]\!]$ is the same as that resulting from executing $L_{123}?\neg\mathsf{win}_2$ (see example figure 6.2.f):

$$\mathsf{rwb}\text{—}2\text{—}bwr$$

In the resulting state, *both* 1 and 3 know the deal of cards, and 2 knows that:

$$(\mathsf{hexa}, rwb)[\![L_{123}?\neg w_1]\!][\![L_{123}?K_2\mathsf{winother}]\!] \models K_2(\mathsf{win}_1 \wedge \mathsf{win}_3)$$

and therefore

$$(\mathsf{hexa}, rwb)[\![L_{123}?\neg w_1]\!][\![L_{123}?K_2\mathsf{winother}]\!] \not\models K_2\mathsf{winother}$$

If player 1 were now to move, he could win!

Is "nobody can win" an unsuccessful update? In example 45 player 1 can *only* win because player 2 announced that either 1 or 3 can win. Or, put differently, because 2 announces that he cannot win, 1 can win. Now in this example, 3 could have won *anyway*, even before 2's announcement. Is there a knowledge game state where no player can win, but after announcing that fact some player can win? In other words, is there a game state (M, w) for a set $\mathbf{A}$ of agents such that:

$$\exists a \in \mathbf{A} : (M, w)[\![L_{\mathbf{A}}?(\bigwedge_{a\in\mathbf{A}} \neg\mathsf{win}_a)]\!] \models \mathsf{win}_a$$

Now abbreviate $\mathsf{nonewin} := \bigwedge_{a\in\mathbf{A}} \neg\mathsf{win}_a$. We then have:

$$(M, w)[\![L_{\mathbf{A}}?\mathsf{nonewin}]\!] \models \neg\mathsf{nonewin}$$

We do not know the answer to this question. An answer is of practical importance for, e.g., building Cluedo-playing programs, because every show or noshow action is followed by a nowin action. Does it pay to execute nowin, or is this a waste of time and computation? Clearly, an answer is also of game theoretical interest.

6.3 Choosing between cards

Every show action in a hexa game is a forced show action: either you *have* one of the requested cards, and you show it, or you don't, and you can't show any of them. Forced show actions result in a restriction of the original model: less worlds, or less access among worlds. The game for 3 players and 4 cards is the most simple knowledge game where some player can choose which card to show. Assume that player 3 holds 2 cards (i.e. the size of the deal is $1|1|2$). The cards are called north, east, south, and west; or n, e, s, w. The actual deal of cards is: 1 holds north, 2 holds east, and 3 holds south and west: $n|e|sw$ or just $nesw$.

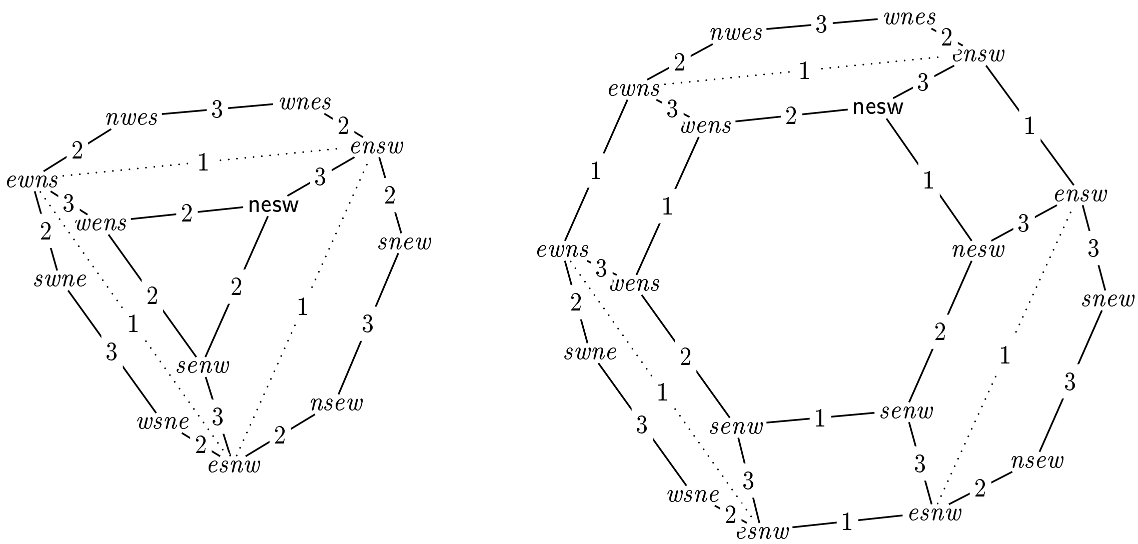


Figure 6.4: On the left, a truncated tetrahedron representing the initial knowledge state where 1 holds north, 2 holds south, and 3 holds east and west. Access for player 1 is only shown in a typical case. On the right, a truncated octahedron representing the state of the game after 3 has shown 2 his south card, given a request for one of his cards. Again, not all access for player 1 is shown.

The initial state $(\mathsf{tetra}, nesw)$ of this game is pictured on the left in figure 6.4. Its shape is that of a truncated tetrahedron, one of the semiregular polyhedra (Archimedean solids). Imagine the corners of the tetrahedron to represent the card that player 2 holds. Truncating a corner creates a triangle. The corners of that triangle represent the (three) different states where 2 holds a certain card. Its edges therefore represent the accessibility relation for player 2. The facing triangle in the figure represents the equivalence class where 2 holds east. At a 'deeper' level, the dotted-line triangle represents an equivalence class for player 1, namely where he holds east. The edges shared by different hexagons represent 3's access.

In the initial state of the game, player 2 asks player 3 to show him one of his cards, and 3 responds by showing the south card. This is the game action

$$\mathsf{show}^{2,-}_{3,s}$$

and the knowledge action that describes it is

$$L_{123}(L_{23}?n_3 \cup L_{23}?e_3 \cup \, !L_{23}?s_3 \cup L_{23}?w_3)$$

The result of executing $\mathsf{show}^{2,-}_{3,s}$ is visualized on the right in figure 6.4. It is a truncated octahedron (another semiregular polyhedron). The edge of tetra that is representing 3's equivalence class where he holds south and west, *nesw*—3—*ensw*, is blown up to a square, so to speak. This happens to all six edges (all six equivalence classes of player 3). In that process, the four triangles of the tetrahedron are simultaneously 'blown up' to four hexagons.

The designated world nesw in the figure on the right corresponds to the world where player 3 has chosen to show the south card. It is different from the other world labeled *nesw*, which corresponds to player 3 having chosen west. Note that in the designated world nesw indeed player 2 can distinguish worlds where 3 holds south from worlds where this is not the case, such as the right *senw* in the lower edge of the facing hexagon of the picture. In the non-designated world *nesw* instead, player 2 cannot make that distinction. These worlds nesw and *nesw* are still connected by a 1-link, in other words: player 1 cannot distinguish the world where 3 has shown south from the world where 3 has shown west instead.

6.4 Cluedo

Cluedo is a knowledge game for six players and a table, where every player holds three cards, and where a question is for one out of three cards. We abstract from the fact that there are different types of cards, which restricts the number of card requests that can be made and the number of card combinations that can be on the table. See chapters 1 and 2. Just as in other knowledge games, there are only four different sorts of action, show, noshow, win and nowin. We give them in more detail and include their descriptions:

- player a asks player b for one of three cards c, c', c'' and b responds by showing (only to a) card c:

$$\mathsf{show}_{b,c}^{a,\{c,c',c''\}}$$
$$L_{123456}(!L_{ab}?c_b \cup L_{ab}?c'_b \cup L_{ab}?c''_b)$$

- player a asks player b for one of three cards c, c', c'' and b responds by saying that he doesn't have any of them:

$$\mathsf{noshow}_{b}^{a,\{c,c',c''\}}$$
$$L_{123456}?(\neg c_b \wedge \neg c'_b \wedge \neg c''_b)$$

- player a announces that he has won the game (that he knows the deal of cards):

$$\mathsf{win}^{a}$$
$$L_{123456}?\mathsf{win}_a$$

- player a passes his move to the next player (and thus implicitly announces that he cannot win the game yet):

$$\mathsf{nowin}^a$$
$$L_{123456}?\neg\mathsf{win}_a$$

In the initial state of Cluedo a show action reduces the complexity of the game state, as measured in the number of nonsimilar worlds, see section 5. In the case of Cluedo, winning the game is knowing the cards on the table. You can win, if in all worlds that are accessible to you, the table 'holds' the same three cards, or in other words: if your information state is contained in an equivalence class of $\sim_0$. It is not known if a nowin action ever changes the information state in nontrivial cases, in other words: if it can be an unsuccessful update, see section 6.2.2. In trivial cases, such as asking for the three cards that you hold yourself, it is informative, see chapter 1.

6.5 Suspicion

Let M be a knowledge model, let τ be an action type executable in that model. Suppose $gr(M) = \mathbf{A}$. A common occurrence is that agents outside $gr(\tau)$ *suspect* that τ has been executed. We can describe this suspicion by the knowledge action type:

$$L_{\mathbf{A}}(\tau \cup\ ?\top)$$

If actually nothing happened, the action that has taken place is:

$$L_{\mathbf{A}}(\tau \cup\ !?\top)$$

If, on the other hand, an action $!_I\tau$ of type τ has been executed, the action that has taken place is:

$$L_{\mathbf{A}}(!_I\tau \cup\ ?\top)$$

For suspicion, see also [Bal99].

Example 46 (Has player 1 cheated?)
Consider the model hexa0, or $preI_{rwb}$: three players have each been dealt one card but haven't turned their cards. The action type where the other players *suspect* player 1 to have looked at his card, is described by:

$$L_{123}(L_1?r_1 \cup L_1?w_1 \cup L_1?b_1 \cup\ ?\top)$$

The resulting model is pictured in figure 6.5. Schematic $\sim_2$ and $\sim_3$ access between the two hexagons is to be interpreted as follows: worlds with the same name are $\sim_2$- and $\sim_3$-related. (As a result all worlds are $\sim_2$- and $\sim_3$-related.) Note that the frame for this action type consists of four alternative actions as worlds, with $\sim_2$ and $\sim_3$ the universal relation and $\sim_1$ the identity on this domain.

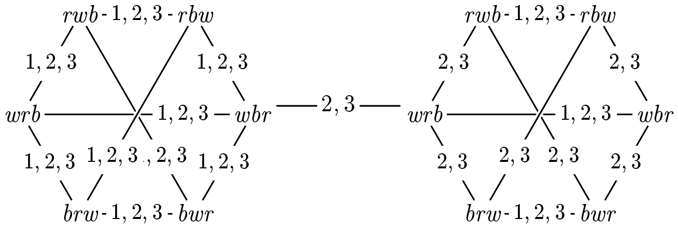


Figure 6.5: hexa0$[\![L_{123}(L_1?r_1 \cup L_1?w_1 \cup L_1?b_1 \cup\ ?\top)]\!]$

6.6 Spreading gossip

One of the multiple-choice questions in the 1999 edition of the nationwide Dutch Science Quiz was the following:

"Six friends each know a secret. They call each other. In each call they exchange all the secrets that they currently know of. How many calls are needed to spread all the news?"[4]

The answer options were: 7, 8 and 9. The correct answer is: 8. In the aftermath of the quiz, the more general question what the minimum would be for any number of callers each having one secret, created a bit of a stir in the media, see the website of the Netherlands Organization for Scientific Research, `http://www.nwo.nl/nwo/quiz/roddelen`. It includes a procedure for communicating n secrets (for $n \geq 4$) in $2n - 4$ calls[5] and a proof ([Hur00]) that $2n - 4$ is minimal.

Optimal call sequence There are various way to communicate all secrets in 8 calls to all persons 1,2,..., 6, most of which start with '1 and 2 call each other', '3 and 4 call each other' en '5 en 6 call each other'. Here, 'calling each other' means 'learning each other's secrets'. Table 6.1 is an example. Call the sequence: six.
Prolonging the pleasures of gossip Instead of the *minimum* number, one might wonder what the *maximum* number of calls is where each time something new is learnt. For n secrets this is $\binom{n}{2}$. For $n = 6$ the maximum can be reached by the following sequence of calls: 1 learns from $2, 3, 4, 5, 6$; 2 learns from $3, 4, 5, 6$; etc. Somewhat surprisingly, the maximum number of *informative* calls happens

[4] In Dutch: *"Zes vriendinnen hebben ieder 'n roddel. Ze bellen elkaar. In elk gesprek wisselen ze alle roddels uit die ze op dat moment kennen. Hoeveel gesprekken zijn er minimaal nodig om iedereen op de hoogte te brengen van alle zes de roddels?"*

[5] Procedure suggested by Gerard Renardel, personal communication. Let ij mean 'i calls j and they pass each other their secrets'. Proof: $n = 4$: 12,34,13,24; $n = 4 + k$: first make k calls from person 1 to the persons over 4: $15, 16, ..., 1(4 + k)$. Then let 1 to 4 make calls as in the case of $n = 4$: $12, 34, 13, 24$. Now repeat the first part of the procedure: $15, 16, ..., 1(4 + k)$.

	call	1	2	3	4	5	6
0		a	b	c	d	e	f
1	12	ab	ab	c	d	e	f
2	34	ab	ab	cd	cd	e	f
3	56	ab	ab	cd	cd	ef	ef
4	13	$abcd$	ab	$abcd$	cd	ef	ef
5	45	$abcd$	ab	$abcd$	$cdef$	$cdef$	ef
6	16	$abcdef$	ab	$abcd$	$cdef$	$cdef$	$abcdef$
7	24	$abcdef$	$abcdef$	$abcd$	$abcdef$	$cdef$	$abcdef$
8	35	$abcdef$	$abcdef$	$abcdef$	$abcdef$	$abcdef$	$abcdef$

Table 6.1: Six, an optimal sequence for communicating six secrets

to be the maximum number of *different* calls between 2 from n persons. Not *any* order of different calls postpones the moment of full information, and not every optimal sequence of calls consists of all different ones.[6]

Multiagent systems Thus far we have described the communication from the viewpoint of an observer that is registering all calls. The telephone company, so to speak. If we describe it from the viewpoint of the callers themselves, i.e. as a multi-agent system, it becomes more complex. What exactly is communicated here? Are the secrets generally or publicly known after the communications? This depends on some further assumptions about the communications protocol. If we *only* assume all communication to be faultless and all secrets to be exchanged in a call, but make *no* further assumptions, after an optimal call sequence the secrets are generally known, i.e. known to all. However, persons do not know that they know all secrets, nor do they know that the other persons know all secrets, they do not even know that there are six persons as a matter of fact. If we assume that it is public knowledge that there are six persons, that each person has one secret, and that all secrets are different, then after the optimal call sequence everybody knows all secrets. Unfortunately, you don't know that the sequence has been executed. You might be one of the first to know all secrets, e.g. person 1 after the sixth call in six. You might also be one of the last, as 3 and 5 after the eighth and last call in six. We need some further assumptions: We will assume that everybody knows which calls have been made to whom, although the secrets that have been exchanged in that call are unknown.

Getting to know each other This calls for a somewhat differently tuned

[6] In the optimal procedure mentioned in footnote 5, person 1 communicates *twice* with all persons over 4. In six, see table 6.1, all calls are different.

example, where the assumptions are obvious. Assume that the six persons are seated around a table, and that they each have a card with the secret written on it. A 'call' now corresponds to two persons exchanging their cards, and both writing their secrets on the other's card, unless the secrets were already on it. The other players obviously see cards being exchanged, but are not allowed to see what is written on them.

Assume that the six secrets a, b, c, d, e, f in the example above are actually the six persons' *names* Anne, Bill, Cath, Dave, Erne and Faye. It seems that we have described a protocol to get to know each other. Now do you know who Anne is, after an optimal eight card exchanges? In the example above, apart from Anne herself, only Bill knows that. The others only know that somebody is called Anne. Apparently this is not the way to make yourself known.

What the six secrets are, is not only generally but also *publicly* known after six. Who a secret originates with, is not even *generally* known after six! Depending on the protocol, there is at most one player who knows the sources of all secrets. This is never the case in an optimal sequence.

Describing the action of calling Communicating secrets can be described as a KT *knowledge action type* in the language $\mathcal{L}_6^{\Box}$ (see chapter 4). The six persons are called $1, 2, 3, 4, 5, 6$. The six secrets are the values of six propositions $p_1, ..., p_6$.[7] The action "i and j learn each others' secrets" is a knowledge action type that is the sequence of six programs of type "everybody learns that a and b learn whether a knows the k-th secret, and everybody learns that a and b learn whether b knows the k-th secret":

$$\begin{array}{lll} \mathsf{call}_{ij} & := & ;_{k=1}^{6}(\quad L_{123456}(L_{ij}?K_ip_k \cup L_{ij}?K_i\neg p_k \cup L_{ij}?\neg(K_ip_k \cup K_i\neg p_k)) \\ & & \qquad ; \\ & & \qquad L_{123456}(L_{ij}?K_jp_k \cup L_{ij}?K_j\neg p_k \cup L_{ij}?\neg(K_jp_k \cup K_j\neg p_k)) \\ & &) \end{array}$$

It will be clear that the knowledge action type describing six, is:

$$\mathsf{call}_{12}\ ; \mathsf{call}_{34}\ ; \mathsf{call}_{56}\ ; \mathsf{call}_{13}\ ; \mathsf{call}_{45}\ ; \mathsf{call}_{16}\ ; \mathsf{call}_{24}\ ; \mathsf{call}_{35}$$

Example 47

The specific action where 5 learns that 2 knows that the value of the secret p_4 is 0, is the KA *knowledge action*:

$$L_{123456}(L_{25}?K_2p_4 \cup\ !L_{25}?K_2\neg p_4 \cup L_{25}?\neg(K_2p_4 \cup K_2\neg p_4))$$

It is a subprogram of an action of type call_{25}. The exclamation mark points at the choice by 2 and 5: what 2 and 5 have really learnt but what is unknown to $1, 3, 4$ and 6, who only know the three alternatives for 2 and 5.

[7]It is less convenient to call the secrets $a, b, ...$, as the agents would have to extend their language all the time, while uncovering them.

6.7 Conclusions

We have presented various uses of the dynamic epistemic language $\mathcal{L}_n^{\Box}$. We introduced the concept of agent-similarity of multiagent models. We paid some attention to the topic of unsuccessful updates. It is unclear whether not being able to win can be an unsuccessful update. Further research should investigate that.

Chapter 7

Update, suspicion, and hypercubes

Our research is on topics as diverse as the logic of questions and answers, the description of finite models, the area of dynamic epistemics, and the modelling of interpreted systems. In previous chapters we already paid attention to the work of other researchers in these areas. In this chapter we discuss the work of some researchers in more detail. A concise, recent, presentation of the logic of questions and answers is [Gro99]. This was already discussed in chapter 2. The description of game states was presented in chapter 3 and is a clear application of the results presented in [vB98, BM96]. In chapter 3 we discussed the relation in detail. In the present chapter we mainly discuss the area of dynamic epistemics. It has seen many publications over the last few years, e.g., [FHMV95, Vil99, vLvdHM95, Koo99, Ren99, vB00a, vB00b, vdM97]. The area initially came to the full attention of the research community by the treatment of public announcements in the famous 'Muddy Children Problem', see [FHMV95].

The work of three researchers in particular is much related to ours. An integrated approach including announcements to subgroups, has been put forward in [GG97]. Gerbrandy's thesis, [Ger99], presents this dynamic epistemics DEL in more generality. We present an embedding of a subclass of our programs into Gerbrandy's DEL programs and suggest a correspondence. Gerbrandy's approach is based on non-well-founded set theory, a non-standard semantics. More recently Alexandru Baltag developed a logic of epistemic actions, based on a standard semantics; [BMS99] treats announcements, and [Bal99] gives an entire framework for epistemic dynamics.[1] We present an embedding of a subclass of our programs into Baltag's class of epistemic action expressions and suggest a correspondence. Our research should probably be seen as a special case of the more general framework as presented by Gerbrandy and by Baltag. Part of its interest lies in the detailed description of new sorts of epistemic action, namely actions in games, and the detailed description of new sorts of information state, namely knowledge games states. That part of our research relates to the modelling with $S5$ models of interpreted systems called hypercubes, as in [Lom99, LvdMR00]. Card game states and hypercubes are much alike.

[1] I discuss the versions of these manuscripts that were available at the time of my research. These are the 1999 versions.

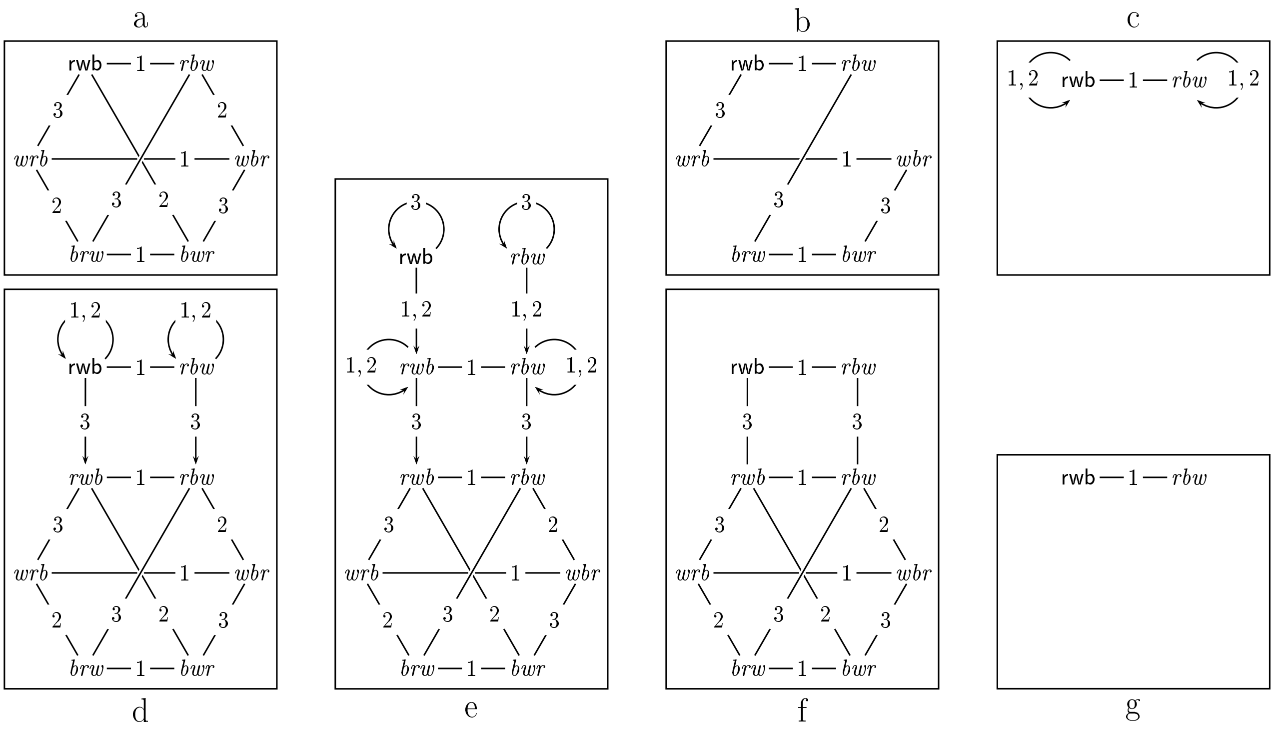


Figure 7.1: Different ways of 1 showing 2 the red card

In section 7.1 we discuss Jelle Gerbrandy's 'Dynamic Epistemic Logic' DEL. In section 7.2 we discuss Alexandru Baltag's 'Logic of Epistemic Actions'. In section 7.3 we discuss the work of Alessio Lomuscio on interpreted systems. We introduce the work of both Gerbrandy and Baltag by means of an introductory example:

The action that player 1 *only* shows player 2 his red card is described by the action $L_{123}(!L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1)$. In figure 7.1.a and 7.1.b the game state (hexa, rwb) and the game state resulting from executing this action are once more shown. The action is publicly interpreted. The action that player 1 shows player 2 his red card *without any assumption on what player 3 learns from that communication* is described by $L_{12}?r_1$. This action is not publicly interpretable: the resulting model has no access defined for player 3. The resulting state is shown in figure 7.1.c; unlike our general convention for $S5$ models, we have in this case explicitly drawn reflexive access, i.e. for 1 and 2. This is to distinguish that figure from figure 7.1.g, where, as usual, reflexive access is implicit, in that case for 1, 2 and 3. Figure 7.1.g is the state resulting from 1 *telling* 2 that he has red, so that this becomes public knowledge. This game action is described by $L_{123}?r_1$.

What state results when 1 shows red to 2, but instead of not assuming anything for 3, we *assume that player 3 learns nothing at all from the communication* *between 1 and 2*? Figure 7.1.d pictures the state resulting from that action. That state is no longer $S5$, because in the point rwb player 3 still 'knows' that 2 can imagine that 1 does not hold red, although this is not actually the case.[2] This example is typical for both [Ger99] and for [BMS99]. In Gerbrandy's DEL, outsiders to a group learning something are always supposed to learn nothing. In our approach we don't make any assumptions about outsiders. In terms of [BMS99], the action resulting in figure 7.1.d is a truthful announcement. Truthful announcements have the *form* 'group A learn proposition φ', just as here.

There are still other ways of 1 showing red to 2. We can make other assumptions about what 3 learns, apart from learning nothing. What if 3 learns more than nothing, e.g. what if he is seeing that 1 shows 2 the red card, although 1 and 2 do not notice that? Now figure 7.1.e results: in the actual world 3 knows that 1 and 2 have common knowledge of 1's red card. We can go on in this manner. The current visualization of our models appears a bit awkward for that. DEL, based on a non-well-founded set semantics, has a much more fitting representation in infinite tree visualizations of models. See the next section.

Now assume that 3 only *suspects* that 1 has shown red to 2, and that this is publicly known. In KA this is described by $L_{123}(!L_{12}?r_1 \cup\ ?\top)$. Figure 7.1.f results. Although in the actual world 1 and 2 share knowledge about 1's red card, 3 cannot distinguish this world from the world where this is not the case. We have treated the topic of suspicion in chapter 6. A more general foundation is given in [Bal99].[3]

Finally assume that 3 knows that 1 has shown red to 2, and that this is publicly known. Again, figure 7.1.g results. This is described by KA action $L_{123}L_{12}?r_1$, which is equal to $L_{123}?r_1$.

Altogether, we have now seen six of infinitely many ways of 1 showing red to 2!

7.1 Gerbrandy: Dynamic Epistemic Logic

We have always assumed that agents do no lie about facts and about their knowledge. If that is the case, everything that is learnt is actually the case, or put more generally: a program that is learnt can be actually executed. More general than learning L is updating U, that does not require such truthful behaviour. The action

$$L_{12}?r_1$$

[2]The only world accessible to 3 from point rwb is the world rwb right below it. From that world, world bwr is accessible to 2. In bwr 1 doesn't hold red.

[3]The actions resulting in figures 7.1.b and 7.1.g can also be modelled in [Bal99].

where 1 and 2 learn that 1 holds red, can (almost) be described in more detail as 'first test on 1 holding red, and only then update with that information':

$$?r_1 \ ; \ U_{12}?r_1$$

For the moment, ignore what happens to player 3. The order of the test and the update is essential: imagine 1 telling 2 "I have red and you don't know that". After updating with that information, it is no longer true that 2 doesn't know that, so the test will always fail after the update, even when it succeeds before the update. This update operator U is the core of Gerbrandy's language DEL. On (hexa, rwb), the interpretation of $L_{12}?r_1$ does not correspond to that of $?r_1 \ ; \ U_{12}?r_1$: the last assumes that 3 learns nothing at all, so that figure 7.1.d results, whereas the first doesn't assume anything about 3.

Similarly to what we did with $L_{12}?r_1$, the action type

$$L_{123}(L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1)$$

can be described by:

$$\begin{gathered}(?r_1 \ ; \ U_{12}?r_1) \cup (?w_1 \ ; \ U_{12}?w_1) \cup (?b_1 \ ; \ U_{12}?b_1) \ ; \\ U_{123}((?r_1 \ ; \ U_{12}?r_1) \cup (?w_1 \ ; \ U_{12}?w_1) \cup (?b_1 \ ; \ U_{12}?b_1))\end{gathered}$$

The difference between a knowledge action and a knowledge action type, was that some agents, in this case 1 and 2, know which choice is made from the alternatives, whereas other agents, in this case 3, don't know that. We therefore remove these choices from the subprogram 'outside the scope of the update operator U_{123}'. This may help to explain that the action

$$L_{123}(!L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1)$$

is described in update semantics as ($\pi_{\mathsf{red}} =$):

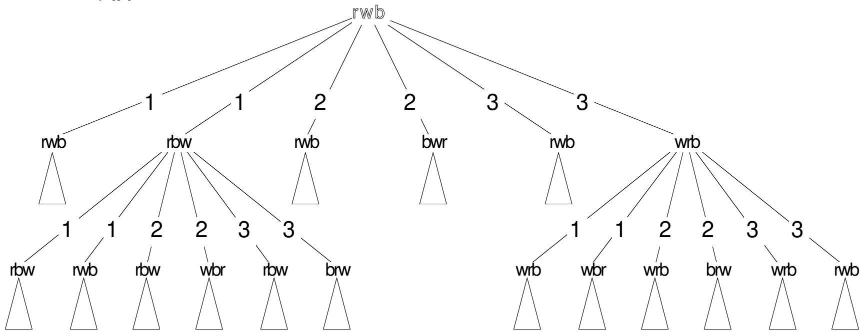
$$(?r_1 \ ; \ U_{12}?r_1) \ ; \ U_{123}((?r_1 \ ; \ U_{12}?r_1) \cup (?w_1 \ ; \ U_{12}?w_1) \cup (?b_1 \ ; \ U_{12}?b_1))$$

In this case, the interpretations on (hexa, rwb) of both programs correspond!

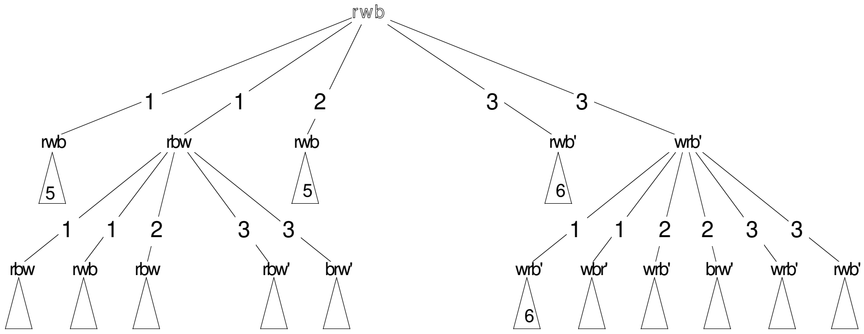
States as infinite trees The logic DEL does not have a standard semantics but a semantics based on non-well-founded set theory. Semantic objects now have an infinitary character. To a state corresponds a semantic object that is called a possibility. A possibility can be visualized as an infinite labeled tree, where the root of the tree is the point of the state to which it corresponds.

The stages of interpretation of π_{red} are visualized in figure 7.2. In all pictures in figure 7.2, nodes have been named by the deals that atomically characterize them. The triangles denote infinite subtrees. Two of the subtrees at depth 1 of the tree have been written out for depth 2. In all pictures, nodes characterized

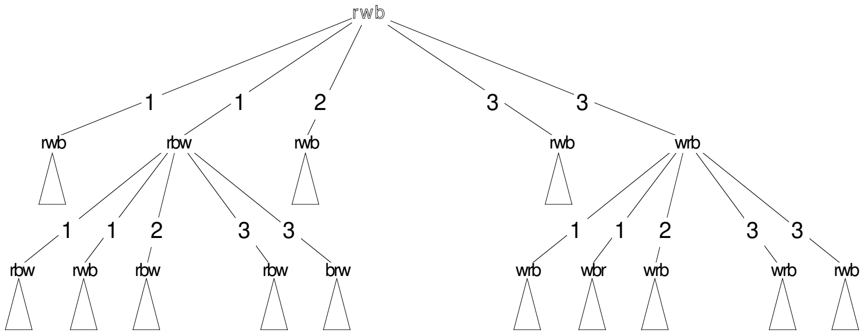
7.2.a: v_{rwb}



7.2.b: $v_{rwb}[\![U_{12}?r_1]\!]$



7.2.c: $v_{rwb}[\![(?r_1\ ;\ U_{12}?r_1)\ ;\ (U_{123}((?r_1\ ;\ U_{12}?r_1) \cup (?w_1\ ;\ U_{12}?w_1) \cup (?b_1\ ;\ U_{12}?b_1))]\!]$



7.2.d: $v_{rwb}[\![U_{123}((?r_1\ ;\ U_{12}?r_1) \cup (?w_1\ ;\ U_{12}?w_1) \cup (?b_1\ ;\ U_{12}?b_1))]\!]$

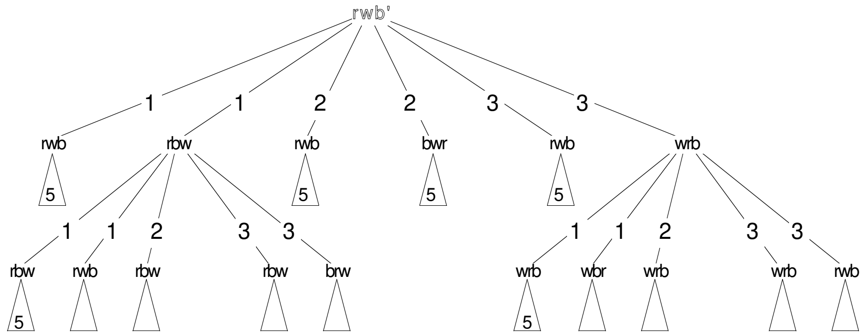


Figure 7.2: Infinite tree visualizations of possibilities

by all six different deals have now been reached. Nodes with the same name in the same picture have identical subtrees, so they are bisimilar.

Interpreting a DEL program can be visualized as pruning such an infinite tree. To compute a test in the tree of a possibility, check if the test formula holds at the root; if so, the same tree results, if not, nothing results. For an update $U_A\pi$ in a tree picturing a possibility w, look at all nodes v (i.e. possibilities) accessible from the root w to agents from group A. If π cannot be executed there (e.g. if the test formula doesn't hold, when it is a test program), the subtree generated by v is pruned from the tree. Otherwise, $U_A\pi$ is computed in v. We can also say that in v we have computed π ; $U_A\pi$. Sequential execution and choice have the obvious interpretation. As in our framework, we write $v[\![\pi]\!]w$ if w results from interpreting π in v, and $v[\![\pi]\!]$ if the interpretation is functional.

The state (hexa, rwb) is pictured in figure 7.2.a. The possibility corresponding to state (hexa, rwb) is called v_{rwb}. To interpret π_{red} on v_{rwb}, first we interpret $?r_1$; $U_{12}?r_1$. The test $?r_1$ succeeds, and (only) all 2-accessible possibilities where r_1 does not hold, are pruned. The result is figure 7.2.b. It is bisimilar to 7.1.d. Similarly, we interpret the public update part $U_{123}(...)$ of the program. The result is figure 7.2.c. That figure is bisimilar to figure 7.1.b. Just interpreting that public update is not enough: then figure 7.2.d would result.

7.1.1 Dynamic epistemic semantics

In this section we present a short overview of the semantics of Dynamic Epistemic Logic DEL.

Definition 51 (Possibility, information state)
Given are a set of agents $\mathbf{A}$ and a set of atoms $\mathbf{P}$. A *possibility* w is a function that assigns to each atom $p \in \mathbf{P}$ a value $w(p) \in \{0, 1\}$ and to each agent $a \in \mathbf{A}$ an information state $w(a)$. An *information state* is a set of possibilities. ([Ger99], 12)

Do not confuse the information state of an agent with a pointed Kripke model that we also call a state (although Gerbrandy does not call that a state, understandably).

There is a precise correspondence between Kripke models and possibilities:

Definition 52 (Decoration, solution, picture)
A *decoration* of a state (M, w) is a function δ that assigns to each world v of (M, w) a possibility δ_v that has the same valuation of atoms and such that the information state $\delta_v(a)$ of each agent a corresponds to the set of worlds accessible to a in (M, w): i.e. $\delta_v(a) = \{\delta_{v'} \mid v \to_a v'\}$, where $\to_a$ denotes the accessibility relation for a. Possibility δ_w is called the *solution* of the state (M, w), notation $\mathsf{sol}(M, w)$. State (M, w) is called a *picture* of possibility δ_w. ([Ger99], 38)

So to a state (M, w) corresponds the possibility δ_w. Generally, instead of δ_w one simply writes w as well, unless confusion would otherwise result. Calling a state a picture of the corresponding possibility, makes it convenient to refer to the visualization of that state as the same thing: namely a picture, whether this is an infinite tree or a finite structure. The information state $w(a)$ of an agent a in a possibility w corresponds to the equivalence class $[w]_{\sim_a}$ for an agent a in a state (M, w).

To define possibilities, one can use *equations*. We then write $v(p) = 1$ if the value of p in v is 1, $v(p) = 0$ if the value of p in v is 0, and $v(a) = \sigma$, if σ is the information state of a in v. Formally, the v in the equations is actually an *indeterminate*, and *the possibility* v is the solution for v given this system of equations. Generally, we can simply identify them. ([Ger99], 14)

Definition 53 (Language of DEL**)**
Given a set of agents $\mathbf{A}$ and a set of atoms $\mathbf{P}$, the set of sentences of DEL is the smallest set closed under $p \mid \varphi \wedge \psi \mid \neg\varphi \mid K_a\varphi \mid [\pi]\varphi$ $(a \in \mathbf{A}, p \in \mathbf{P}, \varphi, \psi \in \mathsf{DEL})$, where $[\pi]$ is a dynamic modal operator for DEL program π. The set of programs of is the smallest set closed under $?\varphi \mid U_A\pi \mid \pi\ ;\ \pi' \mid \pi \cup \pi'$ $(\varphi \in \mathsf{DEL}, A \subseteq \mathbf{A}, \pi$ and π' programs). ([Ger99], 89, 90, 92)

Actually, Gerbrandy distinguishes single agent updates U_a from group updates U_A. As $U_a = U_a$, the distinction is not essential. The language does not have common knowledge operators. This is because, although the semantics of such an extended language is clear, Gerbrandy has only found an axiomatization and corresponding completeness proof of the restricted language.

Definition 54 (Interpretation of formulas)
By inductive cases:

$$\begin{array}{rcl}
w \models p & \text{iff} & w(p) = 1 \\
w \models \varphi \wedge \psi & \text{iff} & w \models \varphi \text{ and } w \models \psi \\
w \models \neg\varphi & \text{iff} & w \not\models \varphi \\
w \models K_a\varphi & \text{iff} & \forall v \in w(a) : v \models \varphi \\
w \models [\pi]\varphi & \text{iff} & \forall v : w[\![\pi]\!]v \Rightarrow v \models \varphi \qquad ([\text{Ger99}], 38, 91)
\end{array}$$

In the following definition, $w[A]v$ means that w differs from v at most in the information states of the agents in group A. Further, we use the abbreviation:

$$u[\![\pi]\!][\![\pi']\!]v \Leftrightarrow \exists w : u[\![\pi]\!]w \text{ and } w[\![\pi']\!]v$$

Definition 55 (Interpretation of programs)
By inductive cases:

$$\begin{array}{rll} w[\![?\varphi]\!]v & \text{iff} & w \models \varphi \text{ and } w = v \\ w[\![\pi \; ; \; \pi']\!]v & \text{iff} & \text{there is a } u \text{ such that } w[\![\pi]\!]u[\![\pi']\!]v \\ w[\![U_A\pi]\!]v & \text{iff} & w[A]v \text{ and } \forall a \in A : v(a) = \{v' \mid \exists w' \in w(a) : w'[\![\pi]\!][\![U_A\pi]\!]v'\} \\ w[\![\pi \cup \pi']\!]v & \text{iff} & \text{either } w[\![\pi]\!]v \text{ or } w[\![\pi']\!]v \qquad \text{([Ger99], 90, 92)} \end{array}$$

Write $v[\![\pi]\!]$ for the unique w, if there is one, such that $v[\![\pi]\!]w$. As updates are total and functional we can use that notation for possibilities resulting from updates.

Gerbrandy gives a sound and complete axiomatization of the set of all theorems (set of DEL sentences true in all possibilities), see [Ger99], 93–102. This concludes our overview of DEL semantics.

We now present some examples:

Example 48 (System of equations)
The possibility v_{rwb} is the solution of state (hexa, rwb). In figure 7.2.a we pictured v_{rwb} as an infinite tree, and in figure 7.1.a as a hexagon. All six possibilities corresponding to states of hexa are defined by the following system of equations. (As already mentioned, the equations are actually in indeterminates, and possibilities are the solutions for the equations, but we will simply identify them.) We have omitted the six times nine equations for the boolean values of atoms, such as $v_{rwb}(r_1) = 1, v_{rwb}(r_2) = 0$, etc.

$$\begin{array}{lll} v_{rwb}(1) = \{v_{rwb}, v_{rbw}\} & v_{brw}(1) = \{v_{brw}, v_{bwr}\} & v_{wrb}(1) = \{v_{wrb}, v_{wbr}\} \\ v_{rwb}(2) = \{v_{rwb}, v_{bwr}\} & v_{brw}(2) = \{v_{brw}, v_{wrb}\} & v_{wrb}(2) = \{v_{wrb}, v_{brw}\} \\ v_{rwb}(3) = \{v_{rwb}, v_{wrb}\} & v_{brw}(3) = \{v_{brw}, v_{rbw}\} & v_{wrb}(3) = \{v_{wrb}, v_{rwb}\} \\ v_{rbw}(1) = \{v_{rwb}, v_{rbw}\} & v_{bwr}(1) = \{v_{bwr}, v_{brw}\} & v_{wbr}(1) = \{v_{wbr}, v_{wrb}\} \\ v_{rbw}(2) = \{v_{rbw}, v_{wbr}\} & v_{bwr}(2) = \{v_{bwr}, v_{rwb}\} & v_{wbr}(2) = \{v_{wbr}, v_{rbw}\} \\ v_{rbw}(3) = \{v_{rbw}, v_{brw}\} & v_{bwr}(3) = \{v_{bwr}, v_{wbr}\} & v_{wbr}(3) = \{v_{wbr}, v_{bwr}\} \end{array}$$

The possibility resulting from execution of action π_{red} in v_{rwb} is like the one above, but with the information state of player 2 in any possibility v consisting of just v. It is pictured by both figure 7.2.c and figure 7.1.b.

Example 49 (Example computation)
We illustrate the execution of action π_{red} in v_{rwb} by showing part of the computation for interpreting $?r_1 \; ; \; U_{12}?r_1$ in that possibility:

$$v_{rwb}[\![?r_1 \; ; \; U_{12}?r_1]\!]v \Leftrightarrow \exists x : v_{rwb}[\![?r_1]\!]x \text{ and } x[\![U_{12}?r_1]\!]v$$

As $v_{rwb} \models r_1$ we have that $x = v_{rwb}$. We continue:

$v_{rwb}[\![U_{12}?r_1]\!]v$
$\Leftrightarrow$
$v_{rwb}[\{1,2\}]v$ and $\forall a \in \{1,2\} : v(a) = \{v' \mid \exists v'' \in v_{rwb}(a) : v''[\![?r_1]\!][\![U_{12}?r_1]\!]v'\}$

We can write $v = v_{rwb}[\![U_{12}?r_1]\!]$, because updates are functional and total. By definition, $v_{rwb}[\![U_{12}?r_1]\!]$ differs from v_{rwb} only in the information state of 1 and of 2. Therefore, the information state of player 3 in $v_{rwb}[\![U_{12}?r_1]\!]$ is the same as in v_{rwb}:

$$v_{rwb}[\![U_{12}?r_1]\!](3) = \{v_{rwb}, v_{wrb}\}$$

We continue with the information state of 2: $v_{rwb}[\![U_{12}?r_1]\!](2) = \{v' \mid \exists v'' \in v_{rwb}(2) : v''[\![?r_1]\!][\![U_{12}?r_1]\!]v'\}$. We have that $v_{rwb}(2) = \{v_{rwb}, v_{bwr}\}$. Test $?r_1$ succeeds on v_{rwb}, thus $v_{rwb}[\![?r_1]\!] = v_{rwb}$, and the resulting $v_{rwb}[\![U_{12}?r_1]\!]$ is the one we are computing right now. However, in possibility v_{bwr} in the information state of player 2 the test $?r_1$ *fails*, as 1 holds blue there. So:

$$v_{rwb}[\![U_{12}?r_1]\!](2) = \{v_{rwb}[\![U_{12}?r_1]\!]\}$$

In case of the information state of 1, the test $?r_1$ can be successfully executed in both possibilities in $v_{rwb}(1)$, i.e. in v_{rwb} and v_{rbw}. Therefore, we can compute that

$$v_{rwb}[\![U_{12}?r_1]\!](1) = \{v_{rwb}[\![U_{12}?r_1]\!], v_{rbw}[\![U_{12}?r_1]\!]\}$$

We now continue by computing the information states of agents in $v_{rbw}[\![U_{12}?r_1]\!]$, etc.

7.1.2 Relation between KA ∪ KT and DEL programs

The interpretation of the KA action $L_{12}?r_1$ in $S5$ state (hexa, rwb) is *different* from the interpretation of the DEL program $?r_1 \ ; \ U_{12}?r_1$ in the solution v_{rwb} of that state. However, the interpretation of the KA action $L_{123}(!L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1)$ in that state is *the same* as (corresponds to) the interpretation of the DEL program π_{red} in that possibility. If we delete access for agent 3 from (hexa, rwb), and information states for agent 3 from v_{rwb}, the interpretations of $L_{12}?r_1$ and $?r_1 \ ; \ U_{12}?r_1$ are 'the same' again. The emerging pattern is, that *publicly interpretable* $\mathsf{KT} \cup \mathsf{KA}$ programs correspond to their DEL counterparts.

We define an embedding $\triangle : \mathcal{L}_n^\Box \to \mathsf{DEL}$. The only crucial case is $(L_B\pi)^\triangle$. We conjecture that the interpretation of a publicly interpretable $\mathsf{KT} \cup \mathsf{KA}$ program π corresponds to that of the DEL program $\pi^\triangle$.

Definition 56 (Embedding $\mathcal{L}_n^\Box \to \mathsf{DEL}$)
The operation $\triangle$ translates $\mathsf{KT} \cup \mathsf{KA}$ programs (without common knowledge operators in tests, as these operators do not occur in DEL) into DEL programs. By

inductive cases on formulas, knowledge action types, and actions:

$$\begin{array}{rcl}
p^{\triangle} & := & p \\
(\varphi \wedge \psi)^{\triangle} & := & \varphi^{\triangle} \wedge \psi^{\triangle} \\
(\neg\varphi)^{\triangle} & := & \neg\varphi^{\triangle} \\
(K_a\varphi)^{\triangle} & := & K_a\varphi^{\triangle} \\
([\pi]\varphi)^{\triangle} & := & [\pi^{\triangle}]\varphi^{\triangle} \\
\\
(?\varphi)^{\triangle} & := & ?\varphi^{\triangle} \\
(L_B\pi)^{\triangle} & := & \pi^{\triangle} \,; U_B\pi^{\triangle} \\
(\pi \,; \pi')^{\triangle} & := & \pi^{\triangle} \,; \pi'^{\triangle} \\
(\pi \cup \pi')^{\triangle} & := & \pi^{\triangle} \cup \pi'^{\triangle} \\
\\
(!_{()}?\varphi)^{\triangle} & := & (?\varphi)^{\triangle} \\
(!_{(I)}L_B\pi)^{\triangle} & := & (!_I\pi)^{\triangle} \,; U_B\pi^{\triangle} \\
(!_{(I,J)}(\pi \,; \pi'))^{\triangle} & := & (!_I\pi)^{\triangle} \,; (!_J\pi')^{\triangle} \\
(!_{0(I,J)}(\pi \cup \pi'))^{\triangle} & := & (!_I\pi)^{\triangle} \\
(!_{1(I,J)}(\pi \cup \pi'))^{\triangle} & := & (!_J\pi')^{\triangle}
\end{array}$$

Example 50
We compute that $(L_{123}(!L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1))^{\triangle} = \pi_{\mathsf{red}}$:

$(L_{123}(!L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1))^{\triangle}$
$=$ see chapter 4
$(!_{(0(0(((),(())),(()))))}L_{123}((L_{12}?r_1 \cup L_{12}?w_1) \cup L_{12}?b_1))^{\triangle}$
$=$
$(!_{0(0(((),(())),(()))}((L_{12}?r_1 \cup L_{12}?w_1) \cup L_{12}?b_1))^{\triangle}$;
$U_{123}((L_{12}?r_1 \cup L_{12}?w_1) \cup L_{12}?b_1)^{\triangle}$

We compute the left part, the computation of the right part is similar:

$(!_{0(0(((),(())),(()))}((L_{12}?r_1 \cup L_{12}?w_1) \cup L_{12}?b_1))^{\triangle}$
$= (!_{0(((),(()))}(L_{12}?r_1 \cup L_{12}?w_1))^{\triangle}$
$= (!_{(())}L_{12}?r_1)^{\triangle}$
$= (!_{()}?r_1)^{\triangle} \;;\; U_{12}(?r_1)^{\triangle}$
$= (?r_1)^{\triangle} \;;\; U_{12}(?r_1)^{\triangle}$
$= ?r_1^{\triangle} \;;\; U_{12}?r_1^{\triangle}$
$= ?r_1 \;;\; U_{12}?r_1$

Clearly the $\triangle$ translation of a $\mathsf{KT} \cup \mathsf{KA}$ program π is a DEL program. We suggest that the interpretations of π and $\pi^{\triangle}$ correspond on *restrictions* of possibilities, a concept that does not occur in [Ger99], but naturally comes up from the viewpoint of local interpretation. Informally define the *restriction of a possibility to a subgroup of agents* as follows: let v be a possibility for group A of agents, let $B \subseteq A$, and let E be the system of equations such that v is a solution for indeterminate x. Define $E\restriction B$ as the system of equations where all equations for information states of agents not in B have been deleted from E. Then $v\restriction B$, the restriction of v to B, is a solution for x in $E\restriction B$. In a picture for the possibility $v\restriction B$ all branches labelled by agents not in B have been pruned.

For restrictions of possibilities, the following appears to hold: Let π be a DEL program such that all occurrences of modal or program operators are in A. Let x, x' be possibilities. Then (not just implication, but equivalence):

$$x[\![\pi]\!]x' \Leftrightarrow x\restriction A[\![\pi]\!]x'\restriction A$$

We conjecture that the embedding $\triangle$ is adequate in the following sense: Let (M, w) be an $S5$ state. Let $\pi \in \mathsf{KP} \cup \mathsf{KA}$ (without common knowledge operators in tests). Then for all (M', w'):

$$(M, w)[\![\pi]\!](M', w') \Leftrightarrow \exists v, \exists B : \mathsf{sol}(M, w)[\![\pi^{\triangle}]\!]v \text{ and } v\restriction B = \mathsf{sol}(M', w')$$

From the conjecture then immediately follows where $\mathsf{KP} \cup \mathsf{KA}$ programs 'are the same' as their $\triangle$ counterparts: Let (M, w) be an $S5$ state. Let $\pi \in \mathsf{KP} \cup \mathsf{KA}$ be publicly interpretable on (M, w). Then for all (M', w'):

$$(M, w)[\![\pi]\!](M', w') \Leftrightarrow \mathsf{sol}(M, w)[\![\pi^{\triangle}]\!]\mathsf{sol}(M', w')$$

The class of fully introspective and reflexive possibilities $\mathcal{P} \cap \mathcal{N} \cap \mathcal{T}$ (see [Ger99], 43) corresponds to the class of multiagent $S5$ states. Let $\mathsf{knowledge}$ be the class of DEL programs such that $\mathcal{P} \cap \mathcal{N} \cap \mathcal{T}$ is closed under their application. Write $(\mathsf{KT} \cup \mathsf{KA})^{\triangle}$ for the class of DEL programs that are translations of knowledge actions or types. From the conjectured adequacy of the embedding follows that $(\mathsf{KT} \cup \mathsf{KA})^{\triangle} \subseteq \mathsf{knowledge}$. Jelle Gerbrandy has been looking in vain for a description of $\mathsf{knowledge}$ (personal communication). We hope to have contributed to this quest.

One may also wonder whether $(\mathsf{KT} \cup \mathsf{KA})^{\triangle} = \mathsf{knowledge}$. This, however, is not the case. A simple example: showing your card is a KT type, showing the red card is a KA action. Both have been translated into DEL programs. However, 'showing the red card or showing the blue card' is described by a DEL program that is in $\mathsf{knowledge}$ but that does *not* correspond (see chapter 4) to a KT or KA program: KA actions are not closed under 'choice', but DEL programs are! These matters clearly require further investigation.

This ends the discussion of Gerbrandy's Dynamic Epistemic Logic DEL.

7.2 Baltag: Logic of Epistemic Actions

We now present Alexandru Baltag's 'Logic of Epistemic Actions' [Bal99] and its precursor 'The Logic of Public Announcements and Common Knowledge' [BMS99] that was written in collaboration with Moss and Solecki.

7.2.1 Public and truthful announcements

Gerbrandy's DEL programs describe many types of agent interaction. However, they are interpreted in a non-standard semantics. In [BMS99] a restricted class of these programs, namely public announcements, are interpreted by means of a standard Kripke semantics. Public announcements are updates on tests φ for a subgroup B of the 'public' $\mathbf{A}$ (in DEL, they have the form $[U_B?\varphi]$). Note that our use of the word 'public' is different from that in [BMS99], as is clear from the previous sentence. We use 'public' for 'the entire group of agents'. Therefore we call those updates just '*announcements*'. Modal operators for announcements are written as $[\varphi]_B$. In [BMS99], the following procedure is presented to interpret the announcement of φ to a subgroup B in a model M: Take two copies of the domain of the original model M 'and redefine access on the result'. In the resulting model $M \oplus^{\varphi,B} M$ then holds $M \oplus^{\varphi,B} M \models \psi \Leftrightarrow M \models [\varphi]_B\psi$.

Definition 57 (Interpretation of announcements)
Let $M = \langle W, (\rightarrow_a)_{a\in\mathbf{A}}, V\rangle$. The model $M \oplus^{\varphi,\mathcal{B}} M$ is defined as follows. The worlds of $M \oplus^{\varphi,B,t} M$ are the elements of the direct sum $M + M$. The left injection of M in $M + M$ is called new, the right injection old. Access for an agent $a \in \mathbf{A}$ is defined as follows (models are left implicit):

$$\begin{array}{rll}
\mathsf{new}(w) \rightarrow_a \mathsf{new}(w') & \text{if} & w \rightarrow_a w', a \in B \text{ and } w' \models \varphi \\
\mathsf{new}(w) \rightarrow_a \mathsf{old}(w') & \text{if} & w \rightarrow_a w' \text{ and } a \notin B \\
\mathsf{old}(w) \rightarrow_a \mathsf{new}(w') & & \text{never} \\
\mathsf{old}(w) \rightarrow_a \mathsf{old}(w') & \text{if} & w \rightarrow_a w'
\end{array}$$

The valuation of atoms on $\mathsf{old}(w)$ and on $\mathsf{new}(w)$ is as on w.

Truthful announcement operators $[\varphi]^t_B$ are also defined. Their interpretation is like that of announcement operators $[\varphi]_B$, except that the new copy of M only consists of worlds where φ holds. The resulting model is written $M \oplus^{\varphi,B,t} M$. We have that $[\varphi]^t_B\psi \leftrightarrow \varphi \wedge [\varphi]_B\psi$. The truthful announcement of φ to B corresponds to the DEL program $?\varphi\ ;\ U_B?\varphi$. If B is the public $\mathbf{A}$, it also corresponds to $L_{\mathbf{A}}?\varphi$, otherwise it does not correspond to a KA action.

Example 51
The action type of 1 showing 2 his red card in the model hexa is the same as the truthful announcement of r_1 to the group $\{1,2\}$. The state in figure 7.1.d on page 104 pictures the result of this announcement in actual state (hexa, rwb). It is therefore also a picture of $(\mathsf{hexa} \oplus^{r_1,\{1,2\},t} \mathsf{hexa}, rwb)$. We have to rename the worlds: the top left world should be named $\mathsf{new}(rwb)$, the top right world $\mathsf{new}(rbw)$; the worlds w in the hexagon should be renamed $\mathsf{old}(w)$.

Before we continue to present [Bal99], a useful analogy. The construction in [BMS99] can be seen as 'defining a successor function' for interpreting updates in standard Kripke semantics: adding one copy of a model to a given model. In [vD99] we defined 'addition' for interpreting updates in standard semantics: adding as many copies as you want, although restricted to game actions. In chapter 4 we defined addition in general. In [Bal99] one finds 'multiplication', and also in chapter 5.

7.2.2 The logic of epistemic actions

The impressive 'A Logic of Epistemic Actions', [Bal99], has not yet reached its final form, and is continually undergoing revisions to further generalizations. The discussion will therefore be restricted. We start with an example. In [Bal99] one can e.g. model *suspicions*.

Suspicion The state $(\mathsf{hexa} \oplus^{r_1,\{1,2\},t} \mathsf{hexa}, rwb)$ is no longer $S5$, as player 3 does not believe the actual state $\mathsf{new}(rwb)$ of the world to be possible (in the figure: rwb in sansserif). This is because not only 3 does not *know* that 1 and 2 have learnt something, but 3 also does not even *suspect* it. With the epistemic action language in [Bal99] one can model both absence and presence of such 'suspicions'. The result of the announcement of r_1 to 1 and 2, where 3 suspects this (and this is publicly known), is the state pictured in figure 7.1.f on page 104. In [Bal99] suspicion is a primitive operation that defines access between actions.[4] We continue with a restricted presentation of the crucial definitions.

Act(∅): a language for epistemic actions Baltag defines a language $Act(\emptyset)$ for action expressions. Assume a multiagent modal language with common knowledge operators and with dynamic operators $[\alpha]$, for, as they are called, closed action expressions $\alpha \in Act(\emptyset)$. Open action expressions in $Act(\vec{x})$ are constructed by one of the following operations, for the sake of readability we use the same notation for operators as in KT:

$$?\varphi \mid x^a \mid \alpha(\vec{x})^a \mid \alpha(\vec{x})^a \cap \beta(\vec{x})^a \mid \alpha(\vec{x})^a \ ; \ \beta(\vec{x})^a \mid \mu y \ . \ \alpha(y, \vec{x})^a$$

Action expression α^a means that agent a suspects action α being executed. It does not mean that α is executed. The operator μ in an action $\mu y \ . \ \alpha(y, \vec{x})^a$ is a fixed-point operator. It expresses self-reference. A closed action expression is one where all variables $\vec{x}$ are bound by μ operators.[5]

Example 52
That 1 shows red to 2, given that these are all agents, is described by the action expression $\mu x \ . \ (?r_1 \cap x^1 \cap x^2)$. It describes the action x such that the test r_1 is

[4]In $\mathcal{L}_n^{\Box}$, the (commonly known) suspicion by all outsiders of an announcement of φ to B is described by the KA action $L_{\mathbf{A}}(?\top \cup\, !L_B?\varphi)$, see chapter 6. The suspicion by one outsider a of that announcement is described by $L_{B\cup\{a\}}(?\top \cup\, !L_B?\varphi)$.

[5]The current version of this action language also contains operators for factual change and an IFTHENELSE operator, personal communication.

executed, and such that 'this' ('x': self-reference) is simultaneously suspected by both 1 and 2.

Example 53
That 1 shows red to 2 without 3 suspecting it, is described by $\mu x\ .\ (?r_1 \cap x^1 \cap x^2 \cap (\mu y\ .\ (\rho \cap y^1 \cap y^2 \cap y^3))^3)$. Here, ρ is the trivial action where nothing happens, just as for $?\top \in \mathsf{KT}$.[6] Player 3 only 'suspects' (considers) that it is publicly known that nothing happens: $\mu y\ .\ (\rho \cap y^1 \cap y^2 \cap y^3)$.

That 1 shows red to 2 *with* 3 suspecting it, and the standard example that 1 shows red to 2 without 3 seeing which card, have a more complex description in $Act(\emptyset)$. We come back to it later.

Act: models for epistemic actions From closed action expressions one can construct actions. First we define the preconditions $PRE(\alpha)$ of an action expression and for each agent $a \in \mathbf{A}$ an accessibility relation between action expressions. Preconditions are defined by induction on the structure of action expressions: $PRE(?\varphi) = \{\varphi\}$, $PRE(\alpha^a) = \emptyset$, $PRE(\alpha \cap \beta) = PRE(\alpha) \cup PRE(\beta)$, $PRE(\alpha\ ;\ \beta) = PRE(\alpha) \cup [\alpha] PRE(\beta)$, $PRE(\mu x\ .\ \alpha(x)) = PRE(\alpha(\mu x\ .\ \alpha(x)))$. Accessibility is also defined by induction on the structure of action expressions: $\alpha^a \to_a \alpha$, $(\alpha \to_a \gamma$ or $\beta \to_a \gamma) \Rightarrow \alpha \cap \beta \to_a \gamma$, $(\alpha \to_a \beta$ and $\alpha' \to_a \beta') \Rightarrow \alpha;\alpha' \to_a \beta;\beta'$, $\alpha(\mu x.\alpha(x)) \to_a \beta \Rightarrow \mu x.\alpha(x) \to_a \beta$. The set of alternatives to a given action expression α is the set of $\to_{\mathbf{A}}$-accessible action expressions: $ALT(\alpha) = \{\beta \mid \alpha \to_{\mathbf{A}} \beta\}$. Together this defines a pointed structure $||\alpha||$ that is called the 'real' action:

$$||\alpha|| = (\langle ALT(\alpha), (\to_a)_{a \in \mathbf{A}}, PRE(ALT(\alpha))\rangle, \alpha)$$

This can then be be 'identified' with a semantic object $(\langle K, (\to_a^K)_{a \in \mathbf{A}}, PRE^K\rangle, k_0)$ that is defined on a domain of abstract 'tokens', instead of action alternatives. The identification is by means of the access and the preconditions, that are the same on both. The class of epistemic actions is called *Act*. We can now define the interpretation of an epistemic action in a state, in Baltag's terms: *epistemic update*.

Definition 58 (Interpretation of epistemic actions)
Given a multiagent state $s = (\langle W, (\to_a^W)_{a \in \mathbf{A}}, V^W\rangle, w_0)$ and an epistemic action $\alpha = (\langle K, (\to_a^K)_{a \in \mathbf{A}}, PRE^K\rangle, k_0)$, *the update* $s \cdot \alpha$ *of* s *by* α is defined as follows: $s \cdot \alpha = (\langle W \cdot K, (\to_a)_{a \in \mathbf{A}}, V\rangle, (w_0, k_0))$, where $W \cdot K = \{(w, k) \mid w \in W$ and $k \in K$ and $w \models PRE^K(k)\}$, where $(w, k) \to_a (w', k') \Leftrightarrow w \to_a^W w'$ and $k \to_a^K k'$, and where $V_{(w,k)} = V_w$.

[6] Actually Baltag writes τ for the trivial action, but we prefer to avoid confusion with the knowledge type variable τ.

Example 54
By applying definition 58 we can compute the epistemic actions for the action expressions from examples 52 and 53. Action expression μx . $(?r_1 \cap x^1 \cap x^2)$ describes the epistemic action consisting of a single node / token k with $k \to_1 k$ and $k \to_2 k$ and with $PRE(k) = \{r_1\}$. The update of (hexa, rwb) by this action delivers figure 7.1.c on page 104; except that for world rwb in that figure, we have to read (rwb, k) and for world rbw, (rbw, k). Action expression μx . $(?r_1 \cap x^1 \cap x^2 \cap (\mu y\ .\ (\rho \cap y^1 \cap y^2 \cap y^3))^3)$ describes the epistemic action consisting of two nodes / tokens k, k' with $k \to_1 k$, $k \to_2 k$, $k \to_3 k'$, $k' \to_1 k'$, $k' \to_2 k'$, $k' \to_3 k'$, and with $PRE(k) = \{r_1\}$ and $PRE(k') = \{\top\}$ (or, $\emptyset$). The update of (hexa, rwb) by this action delivers figure 7.1.d (just as in example 51): note that $PRE(k)$ is satisfied in only two worlds but $PRE(k')$ is satisfied in all worlds. Again, we have to do some renaming of worlds: all the 'old' worlds (deals) d are renamed (d, k'), the new worlds are renamed (rwb, k) and (rbw, k), again.

7.2.3 Relation between KA ∪ KT and Act(∅)

Baltag's notions of *epistemic action* and *epistemic update* are very similar to the notions of *knowledge action frame* and *product interpretation*, respectively, in chapter 5. Our framework can probably be seen as an application of [Bal99]. The $\mathsf{KA} \cup \mathsf{KT}$ programs are more restricted, because they only describe actions that preserve the $S5$ character of states. However, they may be nondeterministic. The relation should be properly investigated. In this section we suggest a translation $\triangledown$ of KA actions into $Act(\emptyset)$ action expressions. This may help to make the comparison.

Definition 59 (Embedding KA → Act(∅))
The operation $\triangledown$ translates KA actions into $Act(\emptyset)$ action expressions. The definition is by induction on the structure of formulas and of actions. In the clause for $L_B\tau$, $C_{I,J} := \{a \in B \mid\ !_I\tau \sim_a\ !_J\tau\}$.

$$\begin{array}{rcl}
p^\triangledown & = & p \\
(\neg\varphi)^\triangledown & = & \neg\varphi^\triangledown \\
(\varphi \wedge \psi)^\triangledown & = & \varphi^\triangledown \wedge \psi^\triangledown \\
(K_a\varphi)^\triangledown & = & K_a\varphi^\triangledown \\
(C_B\varphi)^\triangledown & = & C_B\varphi^\triangledown \\
([\alpha]\varphi)^\triangledown & = & [\alpha^\triangledown]\varphi^\triangledown \\
([\tau]\varphi)^\triangledown & = & \bigwedge_{I \in bu(\tau)}[(!_I\tau)^\triangledown]\varphi^\triangledown
\end{array}$$

$$
\begin{array}{rcl}
(!_{()}\varphi)^\triangledown & = & ?\varphi^\triangledown \\
(!_{(I,J)}(\tau \ ; \ \tau'))^\triangledown & = & (!_I\tau)^\triangledown \ ; \ (!_J\tau')^\triangledown \\
(!_{0(I,J)}(\tau \cup \tau'))^\triangledown & = & (!_I\tau)^\triangledown \\
(!_{1(I,J)}(\tau \cup \tau'))^\triangledown & = & (!_J\tau')^\triangledown \\
(!_{(I)}L_B\tau)^\triangledown & = & \mu x \ . \ (\ \ (!_I\tau)^\triangledown\cap \\
& & \quad \bigcap_{b\in B} x^b\cap \\
& & \quad \bigcap_{J\neq I\in bu(\tau)} \bigcap_{a\in C_{I,J}} (\mu y \ . \ (\ \ (!_J\tau)^\triangledown\cap \\
& & \quad\quad \bigcap_{b'\in B} y^{b'}\cap \\
& & \quad\quad \bigcap_{a'\in C_{I,J}} x^{a'} \\
& & \quad\quad))^a \\
& & \quad)
\end{array}
$$

The translation $\triangledown$ embeds the class of knowledge actions into that of closed action expressions. Knowledge types cannot be translated into action expressions, as they may be nondeterministic. Because knowledge actions can contain tests on formulas containing dynamic modal operators for knowledge types, we have to get rid of those in the translation: this is done in the clause for $([\tau]\varphi)^\triangledown$. Crucial in the translation is the role of local choice operator '!'. Note that the clause $(!_{0(I,J)}(\tau \cup \tau'))^\triangledown = (!_I\tau)^\triangledown$ in the translation, corresponds to the definition $pre(!_{0(I,J)}(\tau \cup \tau')) = pre(!_I\tau)$ of the precondition of an action, in chapter 5.

Action $!_{(I)}L_B\tau$ can be paraphrased as the action where B learn τ with local choice according to (I). Its translation $(!_{(I)}L_B\tau)^\triangledown$ can be paraphrased as follows. It is the action expression where $(!_I\tau)^\triangledown$ is actually executed, and all in B suspect that, and all alternatives $(!_J\tau)^\triangledown$ to $(!_I\tau)^\triangledown$ are suspected by those in B who cannot distinguish between these actions (therefore $C_{I,J} = \{a \in B \mid \, !_I\tau \sim_a \, !_J\tau\}$). The number of epistemic alternatives to an action $(!_I\tau)^\triangledown$ is exactly the number of (other) bundles in $bu(\tau)$.

With this translation we can finally present the action expressions for '1 shows red to 2, with 3 seeing that a card is being shown', and for '1 shows red to 2, with 3 (publicly) suspecting that'.

Example 55

The description of the KA action $L_{123}(!L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1)$ as an $Act(\emptyset)$ action expression is computed by the translation $\triangledown$ as follows. We refrain from details. The tests are in bold, to improve readability:

$$
\begin{array}{l}
(L_{123}(!L_{12}?\mathbf{r_1} \cup L_{12}?\mathbf{w_1} \cup L_{12}?\mathbf{b_1}))^\triangledown \\
= \\
\mu x \ . \ (\quad \mu y \ . \ (?\mathbf{r_1} \cap y^1 \cap y^2) \cap \\
(x^1 \cap x^2 \cap x^3) \cap \\
(\mu z \ . \ (\mu u \ . \ (?\mathbf{w_1} \cap u^1 \cap u^2) \cap (z^1 \cap z^2 \cap z^3) \cap x^3)^3 \cap \\
(\mu w \ . \ (\mu v \ . \ (?\mathbf{b_1} \cap v^1 \cap v^2) \cap (w^1 \cap w^2 \cap w^3) \cap x^3)^3)
\end{array}
$$

The epistemic action that is described by this action expression consists of three alternatives, with access as for the well-known game action $\mathsf{show}^{2,-}_{1,r}$: universal for 3, identity for 1 and for 2.

Example 56
The description of the KA action $L_{123}(?\top \cup\, !L_{12}?r_1)$ as an $Act(\emptyset)$ action expression is $\mu x\ .\ (?r_1 \cap x^1 \cap x^2 \cap x^3 \cap (\mu y\ .\ (\rho \cap y^1 \cap y^2 \cap y^3 \cap x^3))^3)$. Interpretation of the corresponding epistemic action results in figure 7.1.f.

We conjecture the following relation between KA and Act: Let '=' be identity for programs as defined in chapter 4: two programs are the same if their local interpretation defines the same relation on the class of (models and / or) states (modulo bisimilarity). Let $\alpha = !_I\tau \in \mathsf{KA}$ such that for all M where τ is executable, all models in $[M[\![\tau]\!]]$ are of the same group. Then:

$$\alpha = \alpha^{\triangledown}$$

Using the notion of action bisimilarity proposed in [Bal99], and the notion of product interpretation from chapter 5, we may also say that:

$$[\![\alpha]\!]^{\otimes} \underline{\leftrightarrow} ||\alpha^{\triangledown}||$$

Also, the translations $\triangle$ and $\triangledown$ may help to clarify the relation between [Ger99] and [Bal99]. This concludes our discussion of research on dynamic epistemics.

7.3 Lomuscio: Hypercubes

In his recent PhD thesis [Lom99] and in various other publications ([LR98a, LR98b, LvdMR00]) Alessio Lomuscio studies interpreted systems called hypercubes. Hypercubes consist of global states for all combinations of local states of all agents (more precise: the full cartesian product of local states). Hypercubes for n agents can be modelled as $S5_n$ models. Lomuscio characterizes their frames and proves a correspondence result. We will not present his results, but only argue that knowledge game states may be seen as global states of an interpreted system, and that further study of the characterization of game states may profit from Lomuscio's results.

Given are a set $\mathbf{A}$ of n agents and a set $\mathbf{C}$ of m cards, and a deal of cards $d \in \mathbf{A}^{\mathbf{C}}$. We can *also* think of deal d as a *global state*

$$d = \langle l^1_d, ..., l^n_d; l^e_d \rangle$$

of an interpreted system, where l^a_d is the *local state* of agent $a \in \mathbf{A}$ and where l^e_d is the state of the environment, which is meaningless in our context. The

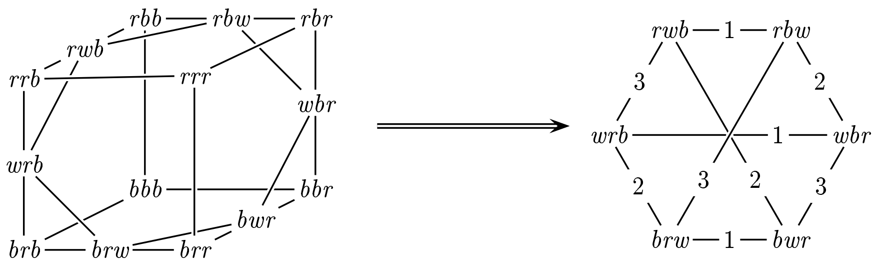


Figure 7.3: On the left a $S5_3$ model for a 3-hypercube. On the right hexa.

interpreted system corresponds to the model I_d for initial knowledge game state $s_d = (I_d, d)$. It consists of all deals of cards of size $\sharp d$ (i.e., deals where all players hold the same number of cards as in d, see chapter 2), and where, as usual for interpreted systems, global states / deals d and d' are indistinguishable from each other for agent a if they agree on his local state: $d \sim_a d' \Leftrightarrow l_d^a = l_{d'}^a$. The local state l_d^a of agent a can be identified with the set $d^{-1}(a)$ of $|d^{-1}(a)| = \sharp a$ cards that a holds. Indeed, deals cannot be distinghuished from each other by a player if they agree on his own cards.

First assume that all players hold one card, i.e. $m = n$. Then every agent has n local states : we may think of a card c held by a player as his local state. The $S5_n$ model M that represents that all players can have all local states without restriction, corresponds to an n-hypercube in terms of [Lom99]. But of course, there are restrictions: all local states must have a different 'value', so to speak: a card can be dealt to one player *only*. The initial model I_d is described by, figuratively speaking, the intersection of an $(n-1)$-dimensional 'hyperplane' with the model M for the hypercube. This hyperplane is defined by the equation that the local states of the n agents 'add up to the set of cards'. The hyperplane represents the model (function space $\mathbf{A}^{\mathbf{C}}$) of all deals of n cards over n players.

If player holds more than one card, we may think of a card c held by a player as the *value* c of a local state *variable*. Every agent a has $\sharp a$ state variables. If a player holds 2 cards, imagine him to have $\binom{m}{\sharp a}$ local states, etc.

In figure 7.3 we illustrate the suggested relation between hypercubes and models for game states. The cube on the left is the $S5_3$ model for the 3-hypercube for 3 agents each having three possible local states: r,w and b. It consists of 27 worlds. These are not all drawn in the figure. E.g. world bbr (or $b|b|r$) corresponds to the global state where agent 1 and 2 have local state b and agent 3 has local state r. The model hexa fits into the cube. It is the intersection of the cube with the 'plane' consisting of the deals in $\{1,2,3\}^{\{r,w,b\}}$: the set of all (incidentally, also 27) deals of three different cards over three players. An example: the deal $rb|w|\varepsilon$, where 1 holds red and blue and 2 holds white and 3 nothing, is in the plane, but neither in the cube, nor in the hexagon.

A comparison of hypercubes and game states may help to describe frame properties of game states. In chapter 2 we have shown that in an initial knowledge game state all worlds are $(\sim_{\cup \mathbf{A}})^3$-related. From one of the frame properties for hypercubes (directedness) follows that all worlds in hypercubes are $(\sim_{\cup \mathbf{A}})^2$-related.

In a more recent publication, [LvdMR00], apart from hypercubes also the more general concept of *full system* is defined. In a full system, local states of agents *still* do not depend upon each other, as they do in models for knowledge game states, but there *is* interaction with the state of the environment. Indeed, in an example of a full system that is not a hypercube, the environment is modelled as a set of stacks of cards, from which a known number of cards is drawn for each agent player! Maybe an even wider class of interpreted systems can be defined, that includes both full systems and systems for knowledge games.

Chapter 8

Conclusions

In the board game Cluedo the players have to determine what the murder cards on the table are, by reasoning about their own cards, and about what they get to know of other players' cards in game actions. Typical for Cluedo is that a finite number of cards is dealt over a finite number of players, that the players can only see their own cards, and that cards do not change hands. We call these games knowledge games. A knowledge game that exemplifies most of the general features that we want to study, is the game for three players each holding one card. We describe game states, game actions, and their consequences.

A *deal of cards* is a function from cards to players. Two deals are indistinguishable for a player if they agree on his cards. The set of *relevant deals* is the set of deals that are publicly indistinguishable from the actual deal of cards. These are the deals that players still have to consider while reasoning about card possession. In the *initial state of the game*, the cards have been dealt but the players have not acquired knowledge about other players' cards. The initial state of the game is represented by a pointed multiagent $S5$ model on the set of deals where all players hold the same number of cards as in the actual deal. The point of the model is the actual deal. Accessibility is indistinguishability of deals. Propositional atoms describe that a player holds a card. Other knowledge game states are also $S5$, but agents may know more.

The properties of the agents in the initial state of a knowledge game are described by the theory $\mathsf{kgames} = \{\mathsf{deals}, \mathsf{seedontknow}\}$. Deals is the disjunction of atomic state descriptions $\delta_{d'}$ of relevant deals d'. It expresses that exactly one of the relevant card deals must be actually the case. Seedontknow expresses that a player considers deals, if and only if they correspond to what he knows about his own cards. Various agent properties follow from kgames, such as that every player knows his own cards and that every player holds a fixed number of cards, and different ways to describe ignorance. By a bisimulation proof we show that kgames (uniquely) describes the model I_d for the initial state of a knowledge game. We also characterize the knowledge game state where the cards have been dealt but where the players have not yet turned their cards. A theory prekgames describes the model $preI_d$ for that state. State descriptions that are computed with standard methods for finite multiagent $S5$ models, are equivalent to our results.

Game actions in knowledge games consist of card requests and responses to those requests, plus some other moves, such as announcing, or guessing, that you have won. A typical game action is that of a player showing a card (only) to another player. The remaining players see that a card is being shown, but cannot see which card. Questions and answers are combined in one format for game actions: a quintuple consisting of the requesting player, the question, the answering player, the answer, and the '*publicity*': what other players get to know about the answer. The question is a set of possible answers. The actual answer is one of those. Each possible answer is a set of worlds of the current game state. Not any set of worlds: an answer must be the union of classes of the partition of that state for the responding player. The question must cover the state: each world must be contained in a possible answer. All desirable constraints on 'publicity', such as that the answering player controls the response, can be realized by formalizing it as a function from players to partitions of the question. The game action format also describes other conceivable game actions, sometimes by the trick of having a player respond to his own question. A game action is executable in a game state if the answer contains the point of that state. A game action minus the roles of the requesting and responding player corresponds to a pointed $S5$ frame on the set of possible answers: a *game action frame*. The next game state is a restriction of the direct product of the current game state and a game action frame.

This game action format is purely semantic: game actions are defined for a given game state. We introduce a general action language to describe game actions, and a corresponding notion of interpretation that we call local interpretation. The language $\mathcal{L}_n^{\Box}$ for dynamic epistemic logic contains dynamic modal operators for, what we call, *knowledge action types* and *knowledge actions*. The basic programming constructs in the action language are test, sequence, choice, learning and local choice. The first four define the class of knowledge action types KT. From an action type we construct a knowledge action of that type by the operation of *local choice*. KA is the class of knowledge actions. In the knowledge game for three players and three cards where player 1 holds red, player 2 holds white, and player 3 holds blue, the game action of 1 showing his red card to 2 is described by the knowledge action $L_{123}(!L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1)$. This stands for: 1 and 2 learn that 1 holds red, and 1, 2 and 3 learn that either 1 and 2 learn that 1 holds red, or that 1 and 2 learn that 1 holds white, or that 1 and 2 learn that 1 holds blue.

The *local interpretation* of a knowledge action type is a relation between multiagent $S5$ models and their worlds. The 'learning' operator is interpreted as follows: given a set of models that is the result of executing knowledge action type τ in a given model M, the result of executing 'group A of agents learn τ' is the direct sum of that set of models plus access added for the agents 'only in A':

an agent cannot distinguish worlds in two different models in the direct sum if he does not occur in those models and if he could not distinguish their origins in M under the interpretation of τ. The local interpretation of a knowledge action in a state (M, w) is derived from the interpretation of that action's type τ by choosing, determined by local choice, *one* model image in the interpretation of τ in M and *one* world image for w in that model image. It can be seen as a constraint on the state transformations induced by τ. Bisimilarity of models and states is preserved after execution of knowledge types and actions.

Game actions are described by knowledge actions. To establish that correspondence, we define another notion of interpretation, called *product interpretation*. The *knowledge action type frame* of a knowledge action type is defined on the set of actions of that type. Access on this frame is determined by the structure of actions. The frame of a knowledge action is a pointed knowledge action type frame. The state transformation induced by that frame is the product interpretation of a knowledge action in an $S5$ state. The notions of local interpretation and product interpretation are the same, up to bisimilarity of states. They are not the same if the local interpretation of a knowledge action type in a model consists of models for different groups. A knowledge action *describes* a game action for a given game state, if there is an isomorphism between the game action frame for that game action and the knowledge action frame for that knowledge action. The isomorphism relates actions of the same type to possible answers to the question of the game action, such that the precondition of an action is satisfied in the worlds that the answer consists of. We suggest a procedure describe that constructs from a game action a knowledge action that describes it.

Apart from game actions we can describe many other communicative acts in $\mathcal{L}_n^{\Box}$. We give examples. We describe suspicion, and we describe sequences of calls over a network. We compare our research to other recent work in the area of dynamic epistemics and multiagent systems.

We have shown that all game states and all game actions in knowledge games can be formally described in a logical language. Given this formal description of games, we can start to think about optimal strategies for winning them. There are some formidable obstacles to overcome here. It is unclear what the individual preferences of a player are among the different questions he can ask. This requires a comparison of the partition refinements, i.e. the new game states, created by the possible answers to those questions. It also requires a (recursive) analysis of the questions that the next player can ask given those refinements. Even when we have individual preferences for all players, it is unclear what the mixed strategy equilibria are for such an imperfect information game. Finally someone may be able to answer the question, what is the value of Cluedo?

Appendix A

Epistemic logic, models

Language of epistemic logic

Definition 60 (Language of multiagent epistemic logic)
Let $\mathbf{P}$ be a set of atomic propositions, let $\mathbf{A}$ be a set of n agents. The set $\mathcal{L}_n$ of multiagent epistemic formulas (for $\mathbf{A}$ and $\mathbf{P}$) is the smallest set closed under:

$$\begin{array}{rll} p \in \mathcal{L}_n & \text{if} & p \in \mathbf{P} \\ \neg\varphi \in \mathcal{L}_n & \text{if} & \varphi \in \mathcal{L}_n \\ \varphi \wedge \psi \in \mathcal{L}_n & \text{if} & \varphi, \psi \in \mathcal{L}_n \\ K_a\varphi \in \mathcal{L}_n & \text{if} & a \in \mathbf{A} \text{ and } \varphi \in \mathcal{L}_n \end{array}$$

Definition 61 (Multiagent epistemic logic with common knowledge)
Let $\mathbf{P}$ be a set of atomic propositions, let $\mathbf{A}$ be a set of n agents. The set $\mathcal{L}_n^C$ of multiagent epistemic formulas (for $\mathbf{A}$ and $\mathbf{P}$) is the smallest set closed under:

$$\begin{array}{rll} p \in \mathcal{L}_n^C & \text{if} & p \in \mathbf{P} \\ \neg\varphi \in \mathcal{L}_n^C & \text{if} & \varphi \in \mathcal{L}_n^C \\ \varphi \wedge \psi \in \mathcal{L}_n^C & \text{if} & \varphi, \psi \in \mathcal{L}_n^C \\ K_a\varphi \in \mathcal{L}_n^C & \text{if} & a \in \mathbf{A} \text{ and } \varphi \in \mathcal{L}_n^C \\ C_A\varphi \in \mathcal{L}_n^C & \text{if} & A \subseteq \mathbf{A} \text{ and } \varphi \in \mathcal{L}_n^C \end{array}$$

Formula $K_a\varphi$ means 'a knows φ'. Another epistemic interpretation is as 'a believes φ', but we never intend it to mean that. Formula $C_A\varphi$ means 'A commonly know φ', or (or 'group A commonly knows φ' or 'φ is commonly known by A'). If $A = \mathbf{A}$ we also say that 'φ is publicly known', or that 'φ is public knowledge'. Uppercase boldface $\mathbf{A}$ always denotes the public.

We introduce the usual abbreviations (let $p \in \mathbf{P}, a \in \mathbf{A}, A \subseteq \mathbf{A}$): $\varphi \vee \psi := \neg(\neg\varphi \wedge \neg\psi)$, $\varphi \rightarrow \psi := \neg\varphi \vee \psi$, $\varphi \leftrightarrow \psi := (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$, $\top := p \vee \neg p$, $\bot := p \wedge \neg p$, $M_a\varphi := \neg K_a \neg\varphi$, $E_A\varphi := \bigwedge_{a \in A} K_a\varphi$. Formula $M_a\varphi$ means 'a can imagine that φ'. Formula $E_A\varphi$ means 'A generally know φ', (or 'group A generally knows φ' or 'φ is generally known by A'). Another abbreviation:

Definition 62 (Exclusive disjunction)
Exclusive disjunction $\triangledown$ is defined as an n-ary operation, for each $n \geq 2$:

$$\triangledown_{i=1}^{n}\varphi_i := (\varphi_1 \wedge \neg\varphi_2 ... \wedge \neg\varphi_n) \vee (\neg\varphi_1 \wedge \varphi_2 ... \wedge \neg\varphi_n) \vee ... \vee (\neg\varphi_1 \wedge \neg\varphi_2 ... \wedge \varphi_n)$$

Instead of $\triangledown_{i=1}^{n}\varphi_i$ we also write $\varphi_1 \triangledown ... \triangledown \varphi_n$.

Semantics of epistemic logic

We only consider models where the accessibility relations R_a are equivalence relations, written as $\sim_a$.

Definition 63 ($\mathbf{S5_n}$ frame/model/state)
Given are a set $\mathbf{A}$ of (n) agents and a set $\mathbf{P}$ of atoms. An $S5_n$ *model* is a triple

$$\langle W, (\sim_a)_{a\in\mathbf{A}}, V\rangle$$

where W is the (nonempty) domain, for each $a \in \mathbf{A}$, $\sim_a \subseteq W \times W$ is the accessibility relation for agent a, which is an equivalence relation, and the valuation V is a function from worlds to propositional valuations (functions from the set of atoms to values 0 and 1): $V : W \to \mathbf{P} \to \{0,1\}$.[1] The pair $\langle W, (\sim_a)_{a\in\mathbf{A}}\rangle$ is called an $S5_n$ *frame*. An $S5_n$ *state* is a pair (M, w) where M is an $S5_n$ model and $w \in M$.

Instead of $S5_n$ model (frame, state) we also say *multiagent* $S5$ *model* (frame, state). In state (M, w), world w is the *point* or *designated world* of the model. If M is a multiagent epistemic model, and $s = (M, w)$ is a multiagent epistemic state, we also say that *s is a state for M*, or that *M is the model underlying s*, or that *M is the model for s*. Also, if $F = \langle W, (\sim_a)_{a\in\mathbf{A}}\rangle$ is an $S5_n$ frame and $w \in W$, then (F, w) is a pointed frame with point (or designated world) w. If M is a $S5_n$ model (frame, state) for group $\mathbf{A}$, we also say that M is an $\mathbf{A}$ model (frame, state) or that the *group* of M is $\mathbf{A}$.

Let $a \in \mathbf{A}, B \subseteq \mathbf{A}$. We introduce the abbreviations:

$$\begin{array}{rcl} \sim_B & := & (\bigcup_{a\in B} \sim_a)^* \\ \sim_{\cup B} & := & \bigcup_{a\in B} \sim_a \end{array}$$

The first is also an equivalence relation, the second not. The notations are nonstandard, but they are very convenient and we will use them often: Relation $\sim_{\cup B}$ is access for modal operator E_B. Relation $\sim_B$ is access for modal operator C_B. If an equivalence relation $\sim$ is the identity on the domain, we write '='. If it is the universal relation on the domain, we write U, or (given domain W) $W \times W$.

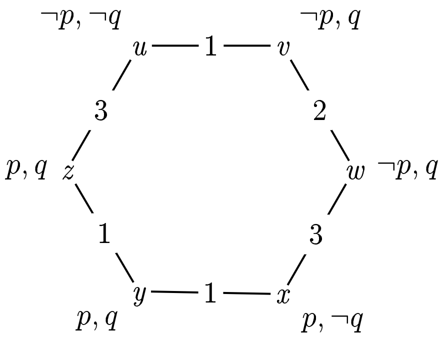
Definition 64 (Information state)
Let $(\langle W, (\sim_a)_{a\in\mathbf{A}}, V\rangle, w)$ be an $S5_n$ state. The information state of agent $a \in \mathbf{A}$ in this state is the set of worlds that are a-accessible from w: $[w]_{\sim_a}$. We also say that $[w]_{\sim_a}$ is the equivalence class of $\sim_a$ (actually) inhabited by a.

[1] Alternatively, we may define valuations as functions from atoms to subsets of the domain: $V : \mathbf{P} \to \mathcal{P}(W)$. We then have, for an atom $p \in \mathbf{P}$, that $V_p \subseteq W$. Such a subset V_p stands for the set of worlds where the atom p holds.

Definition 65 ($\mathbf{S5_n^C}$ ($\mathbf{S5_n^{EC}}$) model)
An $S5_n^C$ model ($S5_n^{EC}$ model) is a multiagent $S5_n$ model with accessibility relations $\sim_B$ ($\sim_{\cup B}$ and $\sim_B$) added for all subgroups $B \subseteq \mathbf{A}$.

Every $S5_n$ model can also be seen as an $S5_n^C$ model and also as an $S5_n^{EC}$ model. The difference is of interest for the completeness theorems for the proof systems of these logics. Purely seen as models, we can identify the notions. We will therefore call also $S5_n^C$ and $S5_n^{EC}$ models, $S5_n$ models.

Example 57 (Conventions in pictures of $S5$ models)
Consider the following $S5_3$ model:



As in this figure, we never label the arcs with accessibility relation $\sim_a$ but just with the agent a for which it is access. As in this picture, reflexivity and transitivity are always assumed. Thus we have, e.g., that $u \sim_1 u, u \sim_2 u, x \sim_1 z$.

Definition 66 (Semantics of epistemic logic)
Let $M = \langle W, (\sim_a)_{a \in \mathbf{A}}, V \rangle$ be an $S5_n$ ($S5_n^{EC}$) model. Let w be a world in M. Let $p \in \mathbf{P}$, $a \in \mathbf{A}$, $B \subseteq \mathbf{A}$.

$$
\begin{array}{lll}
M, w \models p & \text{iff} & V_w(p) = 1 \\
M, w \models \neg\varphi & \text{iff} & M, w \not\models \varphi \\
M, w \models \varphi \wedge \psi & \text{iff} & M, w \models \varphi \text{ and } M, w \models \psi \\
M, w \models K_a\varphi & \text{iff} & \forall w' : w' \sim_a w \Rightarrow M, w' \models \varphi \\
M, w \models C_B\varphi & \text{iff} & \forall w' : w' \sim_B w \Rightarrow M, w' \models \varphi
\end{array}
$$

Proof theory of epistemic logic

Definition 67 (Proof system $\mathbf{S5_n}$)
$\mathbf{S5_n}$ is the following proof system:

$\mathbf{S5_n} \vdash \varphi$	if φ is a tautology
$\mathbf{S5_n} \vdash K_a(\varphi \to \psi) \to (K_a\varphi \to K_a\psi)$	distribution
$\mathbf{S5_n} \vdash \varphi \Rightarrow \mathbf{S5_n} \vdash K_a\varphi$	necessitation
$(\mathbf{S5_n} \vdash \varphi$ and $\mathbf{S5_n} \vdash \varphi \to \psi) \Rightarrow \mathbf{S5_n} \vdash \psi$	modus ponens
$\mathbf{S5_n} \vdash K_a\varphi \to \varphi$	veridicality
$\mathbf{S5_n} \vdash K_a\varphi \to K_aK_a\varphi$	positive introspection
$\mathbf{S5_n} \vdash \neg K_a\varphi \to K_a\neg K_a\varphi$	negative introspection

There are no multiagent equivalences. A proof is a finite sequence of steps $1, .., m$. Each proof step is either an axiom, or the result of an application of modus ponens or necessitation. Instead of $\mathbf{S5_n} \vdash \varphi$ we also write $\vdash_{\mathbf{S5_n}} \varphi$ or, if the context is clear, just $\vdash \varphi$. The $\mathbf{S5_n}$ proof system is sound and complete with respect to the class of $S5_n$ models. For details, see [HC84, MvdH95, FHMV95].

Definition 68 (Proof system $\mathbf{S5_n^{EC}}$)
$\mathbf{S5_n^{EC}}$ is the proof system consisting of all $\mathbf{S5_n}$ axioms and derivation rules plus:

$\mathbf{S5_n^{EC}} \vdash C_A(\varphi \to \psi) \to (C_A\varphi \to C_A\psi)$	distribution
$\mathbf{S5_n^{EC}} \vdash \varphi \Rightarrow \mathbf{S5_n^{EC}} \vdash C_A\varphi$	necessitation
$\mathbf{S5_n^{EC}} \vdash C_A\varphi \to \varphi$	veridicality
$\mathbf{S5_n^{EC}} \vdash C_A\varphi \to E_AC_A\varphi$	$C \to EC$
$\mathbf{S5_n^{EC}} \vdash C_A(\varphi \to E_A\varphi) \to (\varphi \to C_A\varphi)$	induction

The induction axiom is easier to recognize in its equivalent form:

$$\mathbf{S5_n^{EC}} \vdash (\varphi \wedge C_A(\varphi \to E_A\varphi)) \to C_A\varphi)$$

Again, instead of $\mathbf{S5_n^{EC}} \vdash \varphi$ we also write $\vdash_{\mathbf{S5_n^{EC}}} \varphi$ or, if the context is clear, just $\vdash \varphi$. The $\mathbf{S5_n^{EC}}$ proof system is sound and complete with respect to the class of $S5_n^{EC}$ models. In the completeness proof, a given consistent formula φ is satisfied by a world in the finite model that is the filtration of the canonical model for $\mathbf{S5_n^{EC}}$ through the set of subformulas (or negations of subformulas) of φ. Again, for details, see [HC84, MvdH95, FHMV95].

By $\vdash \varphi \Rightarrow \vdash \psi$ we mean that there is an axiomatic proof of ψ in $\mathbf{S5_n}$ (or $\mathbf{S5_n^{EC}}$) plus the axiom φ. By $\varphi \vdash \psi$ we mean $\vdash \varphi \to \psi$: there is a proof of ψ from the *assumption*, not axiom, φ. For an example, we have that $p \Rightarrow \Box p$, i.e. $\vdash p \Rightarrow \vdash \Box p$, but that $p \not\vdash \Box p$, because $\not\vdash p \to \Box p$. For proof theory, see [TS96].

Equivalence relations

An equivalence relation $\sim \subseteq W \times W$ is a binary relation that is reflexive, symmetric and transitive. We write $[w]_\sim$ for the equivalence class of $w \in W$: $[w]_\sim = \{w' \mid w \sim w'\}$. It is often more convenient to think of an equivalence relation as a *partition* of the domain into equivalence classes:

Definition 69 (Covering)
Let U be some domain of objects (the 'universe'). Given a set $X \subseteq U$, a *covering* of X is a nonempty set Y of subsets of U such that X is contained in their union: $X \subseteq \bigcup Y$.

If $X = U$ we have that $X = \bigcup Y$, so that all sets in Y are subsets of X: $\forall Z \in Y : Z \subseteq X$. Unless the universe U is explicitly different from X, we assume that $U = X$.

Definition 70 (Partition)
A *partition* of X is a covering Y of X such that the elements of Y are pairwise disjoint: $\forall Z, W \in Y : Z \neq W \Rightarrow Z \cap W = \emptyset$.

A partition Y of X induces an equivalence relation $\sim$ on X as follows:

$$\forall x, x' \in X : (x \sim x' \Leftrightarrow \exists Z \in Y : x, x' \in Z).$$

Relation $\sim$ is indeed transitive, because the sets in Y are pairwise disjoint: If $x \sim x'$ and $x' \sim x''$, there are $Z', Z'' \in Y$ such that $x, x' \in Z'$ and $x', x'' \in Z''$. Because x' is both in Z' and Z'' and all sets in Y are pairwise disjoint, $Z' = Z''$. Therefore $x \sim x''$.

A covering Y of X induces a reflexive and symmetrical relation R on X as follows:

$$\forall x, x' \in X : (R(x, x') \Leftrightarrow \exists Z \in Y : x, x' \in Z)$$

Relation R is not necessarily transitive.

Definition 71 (Coarser, finer)
Let R, S be two binary relations. By 'R is coarser than S' we mean:

$$\forall x, x' : S(x, x') \to R(x, x'),$$

by 'R is finer than S', or 'R is a refinement of S' ($R \leq S$, $R \subseteq S$), the reverse.

Let Y be a partition of X, with induced equivalence relation $\sim$. Let Y' be a covering of X, with induced binary relation R. If R is coarser than $\sim$ ($\sim$ is a refinement of R), then all members of Y' are the union of some members of Y:

$$\forall Z' \in Y' : \exists Z_1, ..., Z_n \in Y : Z' = \bigcup_{i=1}^{n} Z_i.$$

We also say that Y' is coarser than Y.

Example 58
The set $X = \{\{1\},\{2\},\{3,4\}\}$ is a partition of the set $\{1,2,3,4\}$. The set $Y = \{\{1,3,4\},\{2,3,4\}\}$ is a covering of $\{1,2,3,4\}$. Covering Y is not a partition of $\{1,2,3,4\}$, and it is coarser than partition X.

Let R be a infix binary relation. We write $[xR]$ for the set $\{y \mid xRy\}$.

Models

We give versions for multiagent $S5$ models of some relevant model theoretic concepts. For readability, we give $S5_1$ versions, the $S5_n$ versions are obvious generalizations.

Definition 72 (Direct sum)
Let $\mathbf{M} = \{M^1, ..., M^m\}$ be a set of $S5_1$ models for set of atoms $\mathbf{P}$. Use the notation $M^i = \langle W^i, \sim^i, V^i \rangle$ for an arbitrary model. The *direct sum* (or *disjoint union*) of $\mathbf{M}$, notation $\bigoplus \mathbf{M}$ is defined as follows:

$$\bigoplus \mathbf{M} \quad = \quad \langle W, \sim, V \rangle$$

where:

$$\begin{array}{rcl} W & = & \{(w,i) \mid i \in \{1,...,m\} \text{ and } w \in W^i\} \\ (w,i) \sim (w',j) & \Leftrightarrow & i = j \text{ and } w \sim^i w' \\ V_{(w,i)} & = & V^i_w \end{array}$$

Instead of $\bigoplus \mathbf{M}$ we also write $M^1 \oplus ... \oplus M^m$.

Definition 73 (Direct product)
Let $\mathbf{M} = \{M^1, ..., M^m\}$ be a set of $S5_1$ models for set of atoms $\mathbf{P}$. Use the notation $M^i = \langle W^i, \sim^i, V^i \rangle$ for an arbitrary model. The *direct product* of $\mathbf{M}$, notation $\bigotimes \mathbf{M}$, is defined as follows:

$$\bigotimes \mathbf{M} \quad = \quad \langle W, \sim, V \rangle$$

where:

$$\begin{array}{rcl} W & = & W^1 \times ... \times W^m \\ & = & \{(w^1, ..., w^m) \mid w^1 \in W^1, ..., w^m \in W^m\} \\ (w^1, ..., w^m) \sim (v^1, ..., v^m) & \Leftrightarrow & w^1 \sim^1 v^1 \text{ and ... and } w^m \sim^m v^m \\ V_{(w^1,...,w^m)}(p) = 1 & \Leftrightarrow & V^1_{w^1}(p) = 1 \text{ and ... and } V^m_{w^m}(p) = 1 \end{array}$$

Again, instead of $\bigotimes \mathbf{M}$ we also write $M^1 \otimes ... \otimes M^m$. We can also define the direct product of, e.g., a model and a frame (or even of two frames). In that case, we relax the constraint on the valuation and require that, for a pair

(w, w') consisting of a world w from the model and a world w' from the frame: $V_{(w,w')}(p) = V_w(p)$.

Definition 74 (Bisimulation)
Let $M = \langle W, \sim, V\rangle$ and $M' = \langle W', \sim', V'\rangle$ be $S5_1$ models (for a set of atoms $\mathbf{P}$). A bisimulation between M and M' is a (nonempty) binary relation $\mathfrak{R} \subseteq W \times W'$ such that:

$$\begin{array}{lrcl}
\textbf{Atoms} & \forall w \in M, v \in M' : & & \\
 & \mathfrak{R}(w, v) & \Rightarrow & V_w = V_v \\
\textbf{Forth} & \forall w, w' \in M, \forall v \in M' : & & \\
 & \mathfrak{R}(w, v) \text{ and } w \sim w' & \Rightarrow & \exists v' \in M' : v \sim' v' \text{ and } \mathfrak{R}(w', v') \\
\textbf{Back} & \forall w \in M, \forall v, v' \in M' : & & \\
 & \mathfrak{R}(w, v) \text{ and } v \sim' v' & \Rightarrow & \exists w' \in M : w \sim w' \text{ and } \mathfrak{R}(w', v')
\end{array}$$

For 'M is bisimilar to M'' we write $M \underline{\leftrightarrow} M'$, or, if $\mathfrak{R}$ is the bisimulation: $\mathfrak{R} : M \underline{\leftrightarrow} M'$.

Definition 75 (Bisimulation (between states))
Let (M, w) and (M', v) be $S5_1$ states. If $\mathfrak{R} : M \underline{\leftrightarrow} M'$ and $\mathfrak{R}(w, v)$ then we also say that state (M, w) is bisimilar to state (M', v), and write $(M, w) \underline{\leftrightarrow} (M', v)$, or $\mathfrak{R} : (M, w) \underline{\leftrightarrow} (M', v)$.

Definition 76 (Isomorphism)
Let $M = \langle W, \sim, V\rangle$ and $M' = \langle W', \sim', V'\rangle$ be $S5_1$ models (for a set of atoms $\mathbf{P}$). An isomorphism between M and M' is a bijective function $\mathfrak{I} : W \to W'$ such that:

$$\begin{array}{lrcl}
\forall w \in M : & V_w & = & V'_{\mathfrak{I}(w)} \\
\forall w, w' \in M : & w \sim w' & \Leftrightarrow & \mathfrak{I}(w) \sim' \mathfrak{I}(w')
\end{array}$$

For 'M is isomorphic to M'' we write $M \cong M'$, or $\mathfrak{I} : M \cong M'$.

We sometimes prefer relational notation for isomorphisms. We then write $\mathfrak{I}(w, w')$ instead of $\mathfrak{I}(w) = w'$. In that sense an isomorphism is a special sort of bisimulation: a functional bisimulation whose converse is also functional and such that all domain elements of M and M' have, respectively, an image and an origin under that bisimulation (in other words: the bisimulation is a zigzag connection between M and M' that is functional and conversely functional).

For frames we relax the constraints on valuation: two frames are isomorphic if there is a bijection that preserves access *only*.

Bibliography

[Bal99] A. Baltag. A logic of epistemic actions. Manuscript, 1999.

[BdRV00] P. Blackburn, M. de Rijke, and Y. Venema. Modal logic. To appear, 2000.

[Bin92] K. Binmore. *Fun and Games.* D.C. Heath, Lexington MA, 1992.

[BM96] J. Barwise and L.S. Moss. *Vicious Circles.* CSLI Publications, Stanford CA, 1996.

[BMS99] A. Baltag, L.S. Moss, and S. Solecki. The logic of public announcements and common knowledge. To appear in 'Proceedings of TARK 98 conference', 1999.

[FHMV95] R. Fagin, J.Y. Halpern, Y. Moses, and M.Y. Vardi. *Reasoning about Knowledge.* MIT Press, Cambridge MA, 1995.

[Ger99] J.D. Gerbrandy. *Bisimulations on Planet Kripke.* PhD thesis, University of Amsterdam, Amsterdam, 1999. ILLC Dissertation Series DS-1999-01.

[GG97] J.D. Gerbrandy and W. Groeneveld. Reasoning about information change. *Journal of Logic, Language, and Information*, 6:147–169, 1997.

[Gol92] R. Goldblatt. *Logics of time and computation.* CSLI Publications, Stanford CA, 2 edition, 1992. CSLI Lecture Notes No. 7.

[Gro99] J. Groenendijk. The logic of interrogation. In T. Matthews and D.L. Strolovitch, editors, *Proceedings of the Ninth Conference on Semantics and Linguistic Theory.* CLC Publications, 1999.

[Har84] D. Harel. Dynamic logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume II, pages 497–604, Dordrecht, 1984. Kluwer Academic Publishers.

[HC84] G.E. Hughes and M.J. Cresswell. *A companion to modal logic.* Methuen, London, 1984.

[Hur00] C.A.J. Hurkens. Spreading gossip efficiently. *Nieuw Archief voor Wiskunde*, 5/1–2:208–210, 2000.

[Koo99] B.P. Kooi. The monty hall dilemma. Master's thesis, Groningen University, Groningen, the Netherlands, 1999.

[Koo00] B.P. Kooi. Master(s)mind(s). Manuscript, 2000.

[Lom99] A.R. Lomuscio. *Knowledge Sharing among Ideal Agents.* PhD thesis, University of Birmingham, Birmingham, UK, 1999.

[LR98a] A.R. Lomuscio and M. Ryan. Ideal agents sharing (some) knowledge. In Henri Prade, editor, *Proceedings of ECAI 98*, 1998.

[LR98b] A.R. Lomuscio and M. Ryan. On the relation between interpreted systems and kripke models. In C. Zhang M. Pagnucco, W.R. Wobcke, editor, *Proceedings of the AI97 workshop on the theoretical and practical foundations of intelligent agents and agent-oriented systems*, pages 46–59. Springer Verlag, Berlin, 1998. Volume 1441, Lecture Notes in Artificial Intelligence.

[LvdMR00] A.R. Lomuscio, R. van der Meyden, and M. Ryan. Knowledge in multiagent systems: Initial configurations and broadcast. Manuscript, 2000.

[MvdH95] J.-J.Ch. Meyer and W. van der Hoek. *Epistemic Logic for AI and Computer Science.* Cambridge Tracts in Theoretical Computer Science 41. Cambridge University Press, Cambridge MA, 1995.

[OR94] M.J. Osborne and A. Rubinstein. *A Course in Game Theory.* MIT Press, Cambridge MA, 1994.

[Ren99] G.R. Renardel de Lavalette. Memories and knowledge games. In J. Gerbrandy, M. Marx, M. de Rijke, and Y. Venema, editors, *JFAK. Essays Dedicated to Johan van Benthem on the Occasion of his 50th Birthday*, Amsterdam, 1999. Amsterdam University Press.

[Taa99] N.A. Taatgen. *Learning without limits.* PhD thesis, University of Groningen, Groningen, 1999.

[TS96] A.S. Troelstra and H. Schwichtenberg. *Basic Proof Theory.* Cambridge University Press, Cambridge MA, 1996.

[vB98] J.F.A.K. van Benthem. Dynamic odds and ends. Technical report, ILLC, University of Amsterdam, 1998. Report ML-1998-08.

[vB00a] J.F.A.K. van Benthem. Hintikka self-applied. Manuscript, 2000.

[vB00b] J.F.A.K. van Benthem. Information update as relativization. Manuscript, 2000.

[vD99] H.P. van Ditmarsch. The logic of knowledge games: showing a card. In Eric Postma and Marc Gyssens, editors, *Proceedings of the Netherlands/Belgium Conference on Artificial Intelligence (BNAIC 99)*, pages 35–42, Maastricht University, 1999.

[vdM97] R. van der Meyden. Common knowledge and update in finite environments. To appear in 'Information and Computation', 1997.

[Vil99] A. Vilks. A logic for changing beliefs with applications to reasoning about choice and games. In H. de Swart, editor, *Proceedings of 'Logic, Game Theory and Social Choice' '99*, pages 68–76, Tilburg, 1999. Tilburg University Press.

[vLvdHM95] B. van Linder, W. van der Hoek, and J.-J.Ch. Meyer. Actions that make you change your mind. In A. Laux and H. Wansing, editors, *Knowledge and Belief in Philosophy and Artificial Intelligence*, pages 103–146, Berlin, 1995. Akademie Verlag.

Samenvatting

In het bordspel Cluedo houden alle spelers spelkaarten vast en moeten ze door middel van vragen en antwoorden over elkaars kaarten bepalen wat de kaarten op tafel zijn, de zogenaamde moordkaarten. Wie dit het eerst weet, heeft gewonnen. Spelers kunnen alleen hun eigen kaarten zien, en kaarten veranderen niet van eigenaar. Zulke spelen noemen we kennisspelen, in het Engels: knowledge games. Een precieze beschrijving van spelsituaties en overgangen daartussen is complex, omdat hierin niet alleen moet vastliggen welke kaarten de spelers vasthouden, maar ook wat spelers van de kaarten van andere spelers weten, wat spelers weten dat andere spelers van hun kaarten weten, enzovoort, en ook hoe deze kennis verandert.

Een *kaartverdeling* is een functie van kaarten naar spelers. In de begintoestand, of beginsituatie, van het spel zijn de kaarten verdeeld maar weten de spelers nog niets over de kaarten van anderen. Twee kaartverdelingen zijn 'hetzelfde' voor een speler als hij in beide dezelfde kaarten heeft. Dit induceert een equivalentierelatie op de verzameling van kaartverdelingen waarin iedere speler hetzelfde aantal kaarten heeft als in de werkelijke kaartverdeling. De werkelijke verdeling is een speciale wereld in deze verzameling. Het resultaat is een zogenaamd gepunt multiagent model van de beginsituatie. Andere spelsituaties zijn ook zo te modelleren, maar de spelers kunnen nu meer kennis hebben.

De eigenschappen van de spelers in de begintoestand van een kennisspel worden beschreven door een logische theorie kgames. Met een bisimulatiebewijs tonen we aan dat kgames het model I_d voor de begintoestand van een kennisspel uniek beschrijft. Het model $preI_d$ voor een spelsituatie waarin de kaarten al wel verdeeld zijn maar de spelers hun eigen kaarten nog niet hebben ingezien, wordt uniek beschreven door de theorie $\mathsf{prekgames}$. Logische toestandsbeschrijvingen die worden berekend met standaardmethoden voor eindige multiagent modellen, zijn equivalent aan onze resultaten.

Zetten in kennisspelen bestaan uit vragen en antwoorden op die vragen. Een typerende zet is het op verzoek laten zien van een kaart aan een andere speler, waarbij de overige spelers wel zien dat er een kaart wordt getoond, maar niet welke. Er zijn vijf parameters die de zet bepalen: de vrager, de vraag, de antwoorder, het antwoord, en de 'publiciteit': wat andere spelers te weten komen over het antwoord. De vraag is een verzameling mogelijke antwoorden. Het werkelijke antwoord is daar één van. Ieder mogelijk antwoord is een (speciale) verzameling werelden van de huidige spelsituatie. De publiciteit is een functie van spelers naar partities van de vraag. Op grond van een gegeven spelsituatie en een zet die uitvoerbaar is in die situatie is de volgende speltoestand te berekenen.

We introduceren nu een algemene taal voor het beschrijven van zetten, en een corresponderende notie van interpretatie genaamd 'locale interpretatie'. De taal $\mathcal{L}_n^\Box$ voor dynamische epistemische logica bevat dynamische modale operatoren voor zogenaamde *actietypen* (in het Engels: knowledge action types) en *kennisacties* (in het Engels: knowledge actions). De basisprogrammeerconstructies in deze actietaal zijn test, opvolging, keuze, leren en locale keuze. De eerste vier definiëren de klasse van actietypen. Uit zo'n actietype kan een actie van dat type geconstrueerd worden door de operatie 'locale keuze'. In het kennisspel voor drie spelers en drie kaarten waarin speler 1 de rode, speler 2 de witte, en speler 3 de blauwe kaart vasthoudt, wordt de zet waarin 1 zijn rode kaart aan 2 laat zien beschreven door de kennisactie $L_{123}(!L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1)$. Dit staat voor: 1 en 2 leren dat 1 rood heeft, en 1, 2 en 3 leren dat ofwel 1 en 2 leren dat 1 rood heeft, ofwel 1 en 2 leren dat 1 wit heeft, ofwel 1 en 2 leren dat 1 blauw heeft ('L' staat voor leren, '$\cup$' voor keuze, en '!' voor locale keuze).

De *locale interpretatie* van een actietype is een relatie tussen multiagent $S5$ modellen en hun werelden. De locale interpretatie van een kennisactie in een toestand (M, w) wordt afgeleid uit de interpretatie van het type τ van die actie, door één beeldmodel te kiezen in de interpretatie van τ in M en één beeldwereld voor w in dat beeldmodel, op basis van locale keuze. Bisimilariteit van modellen en toestanden blijft behouden onder uitvoering van actietypen en kennisacties.

Om precies te maken wanneer een zet in een spel wordt *beschreven* door een kennisactie, definiëren we nog een andere notie van interpretatie. Deze zogenaamde *productinterpretatie* maakt gebruik van een gepunt frame, het kennisactieframe (in het Engels: *knowledge action frame*). Net als voor zetten, is hier dus sprake van een echt semantisch object, en niet van een relatie tussen modellen, zoals voor locale interpretatie. Hiermee kan het verband worden gepreciseerd.

Pas gegeven deze formele beschrijving van kennisspelen, kunnen we ons gaan afvragen wat optimale strategieën zijn om ze te winnen, en, bijvoorbeeld, hoe groot de kans is om Cluedo te winnen. Voorlopig is het resultaat van dit onderzoek, dat we nu over een algemene taal $\mathcal{L}_n^\Box$ beschikken om de dynamiek van communicatie formeel te modelleren. We gaan uitvoerig in op voorbeelden van deze ruimere toepasbaarheid, zoals het beschrijven van vermoedens (zoals bij valsspelen), en van het verspreiden van informatie over een (telefoon)netwerk.

Titles in the ILLC Dissertation Series:

ILLC DS-1996-01: **Lex Hendriks**
Computations in Propositional Logic

ILLC DS-1996-02: **Angelo Montanari**
Metric and Layered Temporal Logic for Time Granularity

ILLC DS-1996-03: **Martin H. van den Berg**
Some Aspects of the Internal Structure of Discourse: the Dynamics of Nominal Anaphora

ILLC DS-1996-04: **Jeroen Bruggeman**
Formalizing Organizational Ecology

ILLC DS-1997-01: **Ronald Cramer**
Modular Design of Secure yet Practical Cryptographic Protocols

ILLC DS-1997-02: **Nataša Rakić**
Common Sense Time and Special Relativity

ILLC DS-1997-03: **Arthur Nieuwendijk**
On Logic. Inquiries into the Justification of Deduction

ILLC DS-1997-04: **Atocha Aliseda-LLera**
Seeking Explanations: Abduction in Logic, Philosophy of Science and Artificial Intelligence

ILLC DS-1997-05: **Harry Stein**
The Fiber and the Fabric: An Inquiry into Wittgenstein's Views on Rule-Following and Linguistic Normativity

ILLC DS-1997-06: **Leonie Bosveld - de Smet**
On Mass and Plural Quantification. The Case of French 'des'/'du'-NP's.

ILLC DS-1998-01: **Sebastiaan A. Terwijn**
Computability and Measure

ILLC DS-1998-02: **Sjoerd D. Zwart**
Approach to the Truth: Verisimilitude and Truthlikeness

ILLC DS-1998-03: **Peter Grunwald**
The Minimum Description Length Principle and Reasoning under Uncertainty

ILLC DS-1998-04: **Giovanna d'Agostino**
Modal Logic and Non-Well-Founded Set Theory: Translation, Bisimulation, Interpolation

ILLC DS-1998-05: **Mehdi Dastani**
Languages of Perception

ILLC DS-1999-01: **Jelle Gerbrandy**
Bisimulations on Planet Kripke

ILLC DS-1999-02: **Khalil Sima'an**
Learning efficient disambiguation

ILLC DS-1999-03: **Jaap Maat**
Philosophical Languages in the Seventeenth Century: Dalgarno, Wilkins, Leibniz

ILLC DS-1999-04: **Barbara Terhal**
Quantum Algorithms and Quantum Entanglement

ILLC DS-2000-01: **Renata Wasserman**
Resource Bounded Belief Revision

ILLC DS-2000-02: **Jaap Kamps**
A Logical Approach to Computational Theory Building (with applications to sociology)

ILLC DS-2000-03: **Marco Vervoort**
Games, Walks and Grammars: Problems I've Worked On

ILLC DS-2000-04: **Paul van Ulsen**
E.W. Beth als logicus

ILLC DS-2000-05: **Carlos Areces**
Logic Engineering. The Case of Description and Hybrid Logics