Department of Computer Science,
University of Otago

UNIVERSITY of OTAGO

SAPERE AUDE

Te Whare Wānanga o Otāgo

Technical Report OUCS-2001-02

The description of game actions in Cluedo

Author:H.P. van Ditmarsch

Status: Submitted to Game Theory and Applications VIII

Department of Computer Science,
University of Otago, PO Box 56, Dunedin, Otago, New Zealand

http://www.cs.otago.ac.nz/trseries/

The description of game actions in Cluedo

Submission to Game Theory and Applications VIII

Hans P. van Ditmarsch

Computer Science, University of Otago, New Zealand. Email: hans@cs.otago.ac.nz

Abstract

Game actions in the well-known murder game Cluedo involve interactions of different subgroups of players, that result in complex knowledge changes. We introduce a dynamic epistemic language to describe actions in Cluedo in formal detail. This provides us with a precise description of Cluedo strategies. Optimal strategies are not yet known.

1 Overview

We start in section 2 with an introduction to the game of Cluedo. In section 3 we continue in more detail with a simpler example: three players each holding one card. In section 4 we present the syntax and semantics of a dynamic epistemic language to describe game actions, we apply the language to describe the actions in the three cards example, and we report on some theoretical results. In section 5 we (formally) describe the game actions and strategies of Cluedo. Cluedo is an example of a *knowledge game*. In section 6 we relate on (knowledge) game state descriptions and game aspects of knowledge games.

2 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

Figure 1: Green has done it with a candlestick in the ballroom: the cards

different rooms: hall, kitchen, dining room, study, sitting room, patio, ballroom, library, pool room.

The game is played on a game 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. There are six players. There are six guest cards, six weapon cards, and 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.

The three categories 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. A player's move consists of the following:

Throw the dice. Try to *reach a room* by walking your pawn over the game board. The number of steps on the board may not exceed the outcome of the throw of dice. If a room is reached *voice a suspicion* about it, i.e. about a guest, a weapon and that particular room. As a consequence of the suspicion, the pawn with the same colour as that of the suspected player is moved to the suspected room, and that weapon token is placed in that room. *Gather responses* to that suspicion from the other players. The other players respond to the suspicion in clockwise fashion: either a player doesn't have any of the requested cards, he says so, and the next player responds to the suspicion; or a player holds at least one of the requested cards, he shows exactly one of those to the requesting player *only*, and no further responses may be gathered. You may now either *end your move* (who is next in turn is again determined clockwise) or, if you think you know enough, *make an accusation.* An accusation is also the combination of a suspect, a weapon and a room card, but it plays another role in the game than a suspicion does:

Each player can make an accusation only once. It is not voiced but written

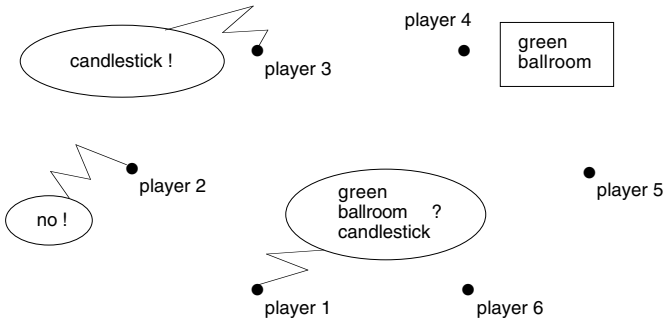

Figure 2: Green has done it with a candlestick in the ballroom: the players

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. If the cards match the accusation, it is successful. The first player who makes a successful accusation, wins the game.

In order to understand why the information changes that result from a move may be complex, we now look in greater detail at an example move.

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. Assume that player 1 starts the game. In his first move, player 1 reaches the ballroom. Player 1 voices the suspicion: 'I think Reverend Green has committed the murder with a candlestick in the ballroom'. Player 2 replies that he does not have any of the requested cards (*nocard*). Player 3 shows the candlestick card to player 1 (*showcard*). Player 1 ends his move (*endmove*).*

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.

How is the knowledge of the players affected by the constituents of this game move? Initially, the players only know their own cards, as this is the first move in the game.

Any suspicion goes: provided one occupies that room, any combination of a room, weapon and guest card can be asked. 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. Although a suspicion of course affects the beliefs and rational behaviour of the players (as we may assume it to be strategically chosen), by itself it does not change their knowledge. The function of a suspicion is to raise an issue, that is resolved by the replies of the other players.

nocard The combination of the issue raised by a suspicion with a negative answer results in an information change. 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 them, but also player 5 knows that player 1 knows that, etc. It has become common knowledge. 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.

showcard The combination of the issue raised by a suspicion with a positive answer also results in an information change. 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 however known by player 4, because 4 holds two of requested cards himself. Curiously enough, neither player 1 nor player 3 know that player 4 knows that. So unlike the case of the nocard action, we now cannot just continue stacking knowledge operators on top of the fact that 3 holds candlestick: it is not common knowledge. Common knowledge among the 6 players is, that player 3 holds at least one of the three requested cards, and that the subgroup consisting of 1 and 3 commonly know which card of those three. From this everybody can deduce that, e.g., 3 does not hold the cards white, scarlett, and conservatory.

endmove Player 1 ends his move. That ending your move is an *epistemic* action is rather implicit. It is a consequence of 1 not making a successful *accusation*, i.e. not declaring what the murder cards are (which is a win action, as explained next). For fully rational players ending your move corresponds to announcing that you are ignorant of the murder cards. Note that an endmove action is independent of the issue raised by the suspicion. It merely depends on the current epistemic state (which resulted of previous actions that depended on the suspicion). In example 1 it is unclear how endmove changes the knowledge of other players. We therefore present a different example in which it is obvious:

Example 2 *Player 1, who starts the game, reaches the kitchen in his first move. He voices the suspicion 'I think that Scarlett has committed the murder in the kitchen with a knife'. None of the other players can show a card. Player 1 (confidently) writes down an accusation, checks the murder cards and announces that he has won (*win*).*

Example 3 *Same as example 2. However, instead of making an accusation, player 1 now ends his move.*

In example 2 it so happened that the murder cards were 'kitchen, scarlett, knife' and that player 1, by mere incredible luck, accidentally asked for precisely these

cards. When nobody shows a card, he therefore *knows* what the murder cards are and wins. In example 3 however, the other players can deduce that player 1 holds at least one of the requested cards, because if he *hadn't*, he would have known what the murder cards are and would have made the accusation, as in example 2.

win Example 3 provided us with the one remaining sort of action occurring in Cluedo. Apart from the nocard, showcard, and endmove actions the only other sort of action occurring in Cluedo is that of winning it. The issue raised by an accusation results in an information change if it is successful. Otherwise we may assume that the accusing player didn't know the murder cards but merely guessed them, wrongly. By telling the other players that he made the wrong guess, he announces that he was ignorant of the murder cards at the moment of the accusation, so the information change is as in endmove. Its strategic role in the game is of course different: a false accusation can only be made once, whereas players end their moves every round.

Simplifications To describe Cluedo game actions in more detail we need to make some simplifications: we disregard the role of the board, dice and pawns, and we ignore that there are different types of cards.

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. To make an obvious point: although pawns are identied with guests, you don't even know whether you have committed the murder 'yourself'; they only serve as constraints on the suspicions that can be made in your move.[1]

Not just any suspicion can be made but only a suspicion consisting of a card of each category. 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 relevant card deals. We assume that all cards are different, and forget about categories.[2]

[1] Disregarding board, dice and pawns is less of a simplification than one might think, because one can generally reach a room, because one is allowed to make other suspicions about the room one already occupies, and because it is unclear why some suspicions should be preferred over others, even if they are about the same room (see section 6). How unlikely is it to reach a room? 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 $\frac{33}{36}$.

[2] How many deals are there in Cluedo? The murder cards come from different categories. so there are $6 \cdot 6 \cdot 9 = 324$ possible combinations of cards on the table. Without the restriction

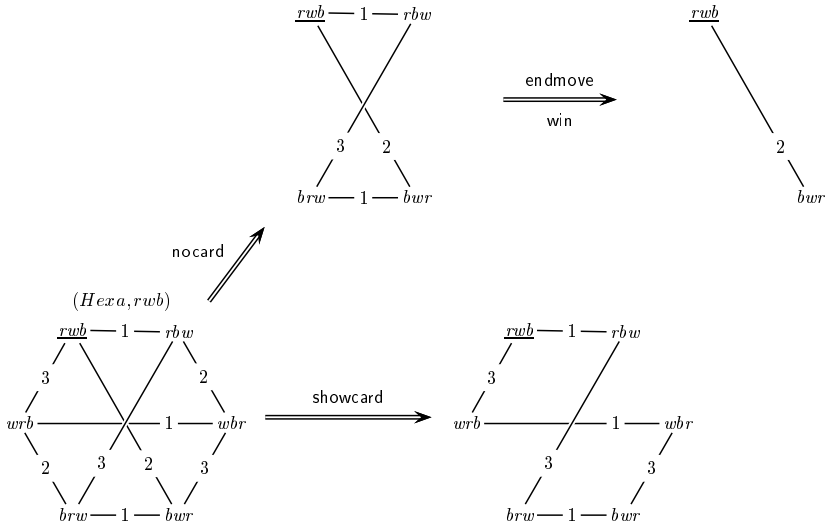

Figure 3: The results of executing nocard and showcard in the state $(Hexa, rwb)$, and the result of both endmove and win in the state resulting from nocard. Points of states are underlined. Worlds are named by the deals that (atomically) characterize them. Assume reflexivity of access.

Given the simplifications, what remains is a card game for 21 cards and six players, with game actions of the sort nocard, showcard, endmove and win. Because actions *only* consist of information change, we call this and similar games *knowledge games*. See [vD01b] and section 6. To describe these game actions in detail, we now move to the simplest game that still contains their essential features.

3 Three players and three cards

Three players each hold one card. Suppose player 1 holds a red card, 2 holds a white card and 3 holds a blue card. This initial game state is modelled by the equivalence state $(Hexa, rwb)$ in Figure 3. An equivalence state is a pointed multiagent Kripke-model where all accessibility relations are equivalences. The model called $Hexa$ (because it looks like a hexagon) consists of the six deals of three cards over three players. In a deal ijk player 1 holds card i, 2 holds j and 3 holds k. Two deals cannot be distinguished from each other by a player if he holds the same card in both, e.g. rwb and rbw are the same for player 1, which is indicated by a 1-labeled link in the figure. The point rwb of the state, which

on categories, there would have been $\binom{21}{3} = 1330$ to consider. After the murder cards have been drawn, the remaining cards are shuffled again. Therefore, the total number of relevant card deals is $324 \cdot \Pi_{i=1}^{6} \binom{3 \cdot i}{3}$.

stands for the 'actual world', is underlined in the figure. Some actions that can be executed in $(Hexa, rwb)$ are:

Example 4 (nocard) *Player 1 says that he doesn't have the white card.*

Example 5 (showcard) *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 6 (endmove) *After the* nocard *action, player 2 ends his move (implicitly says that he cannot win).*

Example 7 (win) *After the* nocard *action, player 3 says that he has won.*

We assume that only the truth is told and that it is publicly known what players can and cannot see. To win is (being the first) to announce that you know the deal of cards. Figure 3 pictures the states that result from updating the current state $(Hexa, rwb)$ with the information contained in the actions.

In nocard it suffices to eliminate some worlds: after 1's action, the two deals of cards where 1 holds white are eliminated. It is publicly known that they are no longer accessible. This update is a public announcement.

In showcard 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 showcard.

The endmove action is another public announcement, just as nocard. Again, only those deals of cards remain where the statement holds, i.e. where 2 still has an alternative. This is the case in rbw and wbr. These also happen to be the worlds where player 3 can win, therefore the win action results in the same state. Incidentally, after 1 has shown his card to 2 in $(Hexa, rwb)$, player 2 knows the deal of cards, and therefore can also perform a win action. Because it is already publicly known that 2 knows the deal of cards, this action does not result in an information change.

Finally note, that in the endmove action, because 2 announces that he cannot win, 1 can win, even though 1 couldn't win before. Because 2 said that he didn't know the deal of cards yet, 1 can conclude that 2 must have white, and therefore 1 knows that the deal of cards is rwb.

We can paraphrase some more of the structure of the actions. In nocard, all three players *learn* that player 1 does not hold the white card, where 'learning' should be regarded as the dynamic equivalent of 'common knowledge'. It is hard to give a more precise informal meaning to 'learning'. In particular, 'learning' is *not* the same as 'becoming common knowledge'. Imagine that instead of saying that he doesn't have white, 1 had said 'one of you doesn't know that I don't have white'. This is interpreted as: '1 does not have the white card, and 2 doesn't know that or 3 doesn't know that'. At the moment of utterance, this statement can be truthfully made in all worlds of $Hexa$ but wrb and wbr, so it

results in the same state as the execution of nocard. However, in that state is it *not* common knowledge that 2 doesn't know that 1 does not have white or that 3 doesn't know that. To the contrary: 2 and 3 *do* both know that 1 doesn't have white, it is even common knowledge.

We continue our conceptual analysis. The structure of endmove and win is similar to that of nocard, because all three are public announcements. Action showcard is more complex. In showcard, 1 and 2 learn that 1 holds red, whereas 1, 2 and 3 learn 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.

'Learning' is an operator in the dynamic language for actions that we will now present. 'Local choice' is another operator and is used to express that subgroup choices are known 'locally' only.

4 Knowledge actions

The area of dynamic epistemics has recently seen much progress [Par87, FHMV95, Pla89, GG97, Ger99, Bal99, BMS00, vD99, vD00, vD01a]. An early example of a dynamic epistemic logical language (as opposed to a metalevel treatment) is the elegant logic of public announcements presented in [Pla89]. Plaza models public announcements as binary operators that have a dynamic interpretation. An integrated approach including announcements to subgroups has been put forward in [GG97]. Gerbrandy's thesis, [Ger99], presents this dynamic epistemics in more generality. A treatment of epistemic actions as semantic objects, namely Kripke frames for actions, is found in [BMS00, Bal99]. Our own work provides both a relational action semantics [vD99, vD01a] and an action frame semantics for game actions [vD01b]. These two different interpretations are equivalent up to bisimilarity [vD00]. Here, we present the extension of the relational semantics to include concurrent actions. This builds upon work on concurrency in dynamic logic PDL [Pel87, HKT00, Gol92] and also appears to be related to game theoretical semantics for (extensions of) PDL [Par85, Pau00].

4.1 Syntax

To a standard multiagent epistemic language with common knowledge for a set A of agents and a set P of atoms [MvdH95, FHMV95, vdHV01], we add dynamic modal operators for programs that are called knowledge actions and that describe actions. The language $\mathcal{L}_A$ and the knowledge actions KA_A are defined by simultaneous induction.

Definition 1 (Dynamic epistemic logic – $\mathcal{L}_A$) *$\mathcal{L}_A(P)$ is the smallest set such*

that, if $p \in P, \varphi, \psi \in \mathcal{L}_A(P), a \in A, B \subseteq A, \alpha \in \mathsf{KA}_A(P)$*, then*

$$p, \neg\varphi, (\varphi \wedge \psi), K_a\varphi, C_B\varphi, [\alpha]\varphi \in \mathcal{L}_A(P)$$

Other propositional connectives and modal operators are defined by abbreviations. Outermost parentheses of formulae are deleted whenever convenient. As we may generally assume an arbitrary P, write $\mathcal{L}_A$ instead of $\mathcal{L}_A(P)$.

Definition 2 (Knowledge actions – KA_A) *Given a set of agents A and a set of atoms P, the set of* **knowledge actions** $\mathsf{KA}_A(P)$ *is the smallest set such that, if $\varphi \in \mathcal{L}_A(P), \alpha, \alpha' \in \mathsf{KA}_A(P), B \subseteq A$, then:*

$$?\varphi, L_B\alpha, (\alpha \ ; \ \alpha'), (\alpha \cup \alpha'), (\alpha \ ! \ \alpha'), (\alpha \cap \alpha') \in \mathsf{KA}_A(P)$$

Outermost parentheses of actions are deleted whenever convenient. We generally write KA_A instead of $\mathsf{KA}_A(P)$. The program constructor L_B is called **learning**. Action $?\varphi$ is a **test**, $(\alpha \ ; \ \alpha')$ is **sequential execution**, $(\alpha \cup \alpha')$ is **nondeterministic choice**, $(\alpha \ ! \ \alpha')$ is called **local choice**, and $\alpha \cap \alpha'$ is a **concurrent knowledge action**.

Actions without '!' operators are called **knowledge action types**, or just action types, actions without '$\cup$' operators are called **concrete knowledge actions**, or concrete actions. An inductively defined function $t : \mathsf{KA}_A \to \mathsf{KA}_A$ returns the type of a given action, with crucial clause $t(\alpha \ ! \ \alpha') = t(\alpha) \cup t(\alpha')$. An inductively defined function $C : \mathsf{KA}_A \to \mathcal{P}(\mathsf{KA}_A)$ returns the set of concretizations of a given action; its crucial clause is $C(\alpha \cup \alpha') = \{\beta \ ! \ \beta' \mid \beta \in C(\alpha) \text{ and } \beta' \in C(\alpha')\} \cup \{\beta' \ ! \ \beta \mid \beta \in C(\alpha) \text{ and } \beta' \in C(\alpha')\}$. Instead of $\alpha \ ! \ \alpha'$ we generally write $!\alpha \cup t(\alpha')$ or $t(\alpha') \cup !\alpha$. This more clearly expresses that given choice between α and α', the agents involved in those actions choose α, whereas that choice remains invisible to to the agents not involved (examples in the next section).

4.2 Examples

We describe the actions nocard, showcard, endmove, and win in $\mathcal{L}_{\{1,2,3\}}$ ($\mathcal{L}_{123}$).

Example 8 (nocard) *Player 1 says that he doesn't have the white card:* $L_{123}?\neg w_1$.

Player 1 may be lying or suffering from an exceptional case of colour-blindness. Therefore the description $L_{123}?K_1\neg w_1$. would be more accurate. However, as we assume truthfulness and normal vision, i.e. as we restrict ourselves to equivalence models, this amounts to the same as $L_{123}?\neg w_1$.

Example 9 (showcard) *Player 1 shows (only) player 2 his red card:* $L_{123}(!L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1)$.

Before describing the concrete action showcard, in which 2 shows 1 his *red* card, we describe the *type* of that action: 1 shows 2 *his* card. The action type can be paraphrased 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)'. Its description in $\mathcal{L}_A$ is $L_{123}(L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1)$. We assume associativity of $\cup$ (see section 4.5).

To describe the concrete action showcard we express what is known to agents 1 and 2, but not to agent 3, from the two choices to be made in $L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1$: between $(L_{12}?r_1 \cup L_{12}?w_1)$ and $L_{12}?b_1$, choose the first. So we get $L_{123}((L_{12}?r_1 \cup L_{12}?w_1) \,!\, L_{12}?b_1)$. Between $L_{12}?r_1$ and $L_{12}?w_1$, again choose the first: $L_{123}((L_{12}?r_1 \,!\, L_{12}?w_1) \,!\, L_{12}?b_1)$. In more readable notation this becomes $L_{123}(!(!L_{12}?r_1 \cup L_{12}?w_1) \cup L_{12}?b_1)$ and assuming associativity again we get $L_{123}(!L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1)$. The action of player 1 showing player 2 his blue card is described as $L_{123}(L_{12}?b_1 \,!\, (L_{12}?r_1 \,!\, L_{12}?w_1))$, which in more readable notation becomes $L_{123}(!L_{12}?b_1 \cup t(L_{12}?r_1 \,!\, L_{12}?w_1)) = L_{123}(!L_{12}?b_1 \cup (L_{12}?r_1 \cup L_{12}?w_1)) = L_{123}(L_{12}?r_1 \cup L_{12}?w_1 \cup !L_{12}?b_1)$.

Example 10 (endmove) *Player 2 cannot win:* $L_{123}?\neg K_2\delta$.

Here $K_2\delta$ is shorthand for $K_2\delta_{rwb} \cup K_2\delta_{rbw} \cup ...$, where δ_{ijk} is the atomic description of world (deal) ijk, e.g. $\delta_{rwb} := r_1 \wedge \neg r_2 \wedge \neg r_3 \wedge \neg w_1 \wedge w_2 \wedge \neg w_3 \wedge \neg b_1 \wedge \neg b_2 \wedge b_3$.

Example 11 (win) *Player 3 can win:* $L_{123}?K_3\delta$.

To illustrate the use of KA_A as a game action description language, we also give an example card action involving concurrency.

Example 12 (twocards) *There are three players (1,2,3) and four cards. The cards are shuffled and dealt to the players. Player 3 is dealt two cards. Player 3 now shows one card (only) to player 1, with his left hand, and (simultaneously) the other card (only) to player 2, with his right hand.*

Suppose the cards are called north, east, south and west (n, e, s, w). Atomic propositions c_a describe that player a holds card c. The action twocards is described by the knowledge action $L_{123}(\bigcup_{i \neq j \in \{n,e,s,w\}}(L_{13}?i_3 \cap L_{23}?j_3))$. In each game state twocards has two possible executions: player 3 may choose whether to show player 1 his i or his j card, the other card is then necessarily shown to player 2. Abstract simultaneous move games [OR94] may provide other examples of concurrent actions.

4.3 Semantics

Given a set of **agents** A and a set of **atoms** P, a (Kripke) **model** $M = \langle W, \{R_a\}_{a \in A}, V\rangle$ consists of a domain W of **worlds**, for each agent $a \in A$ a binary **accessibility relation** R_a on W, and a **valuation** $V : P \to \mathcal{P}(W)$. Given a model, the operator gr returns the set of agents: $gr(\langle W, \{R_a\}_{a \in A}, V\rangle) = A$; this is called the **group** of the model. The group of a set of models is the union of the groups of these models. In an **equivalence model** (commonly known as an $S5$ model) all accessibility relations are equivalence relations. We then write $\sim_a$ for the equivalence relation for agent a. If $w \sim_a w'$ we say that w

is the same as w' for a, or that w **is equivalent to** w' for a. Write $\sim_B$ for $(\bigcup_{a\in B} \sim_a)^*$. For a given model M, $\mathcal{D}(M)$ returns its domain. Instead of $w \in \mathcal{D}(M)$ we also write $w \in M$. Given a model M and a world $w \in M$, (M, w) is called a **state**, w the **point** of that state, and M the model **underlying** that state. Also, if M is clear from the context, write $\underline{w}$ for (M, w). Similarly, we *visually* point to a world in a figure by underlining it. If $s = (M, w)$ and $w \in \mathcal{D}(M)$ we also write $w \in s$. All notions for models are assumed to be similarly defined for states. We introduce the abbreviations $\mathcal{S}_A(P)$ for the class of equivalence states for agents A and atoms P and $\mathcal{S}_{\subseteq A}(P) := \bigcup_{B\subseteq A} \mathcal{S}_B(P)$. As before, drop the '$P$'. Write either *$s$ is an equivalence state* or, if the context requires more precision, $s \in \mathcal{S}_A$ or $s \in \mathcal{S}_{\subseteq A}$.

The semantics of $\mathcal{L}_A$ (on equivalence models) is defined as usual [MvdH95], plus an additional clause for the meaning of dynamic operators. The interpretation of a dynamic operator is a relation between an equivalence state and a set of equivalence states (to be defined in definition 5).

Definition 3 (Semantics of $\mathcal{L}_A$) *Let $(M, w) = s \in \mathcal{S}_A$ and $\varphi \in \mathcal{L}_A$, where $M = \langle W, \{\sim_a\}_{a\in A}, V\rangle$. We define $s \models \varphi$ by induction on the structure of φ.*

$$
\begin{array}{lcl}
M, w \models p & :\Leftrightarrow & w \in V(p) \\
M, w \models \neg\varphi & :\Leftrightarrow & M, w \not\models \varphi \\
M, w \models \varphi \wedge \psi & :\Leftrightarrow & M, w \models \varphi \text{ and } M, w \models \psi \\
M, w \models K_a\varphi & :\Leftrightarrow & \forall w' : w' \sim_a w \Rightarrow M, w' \models \varphi \\
M, w \models C_B\varphi & :\Leftrightarrow & \forall w' : w' \sim_B w \Rightarrow M, w' \models \varphi \\
M, w \models [\alpha]\varphi & :\Leftrightarrow & \forall S \subseteq \mathcal{S}_{\subseteq A} : (M, w)[\![\alpha]\!]S \Rightarrow \exists (M', w') \in S : M', w' \models \varphi
\end{array}
$$

The notion $\langle\alpha\rangle$ is dual to $[\alpha]$ and is defined as $s \models \langle\alpha\rangle\varphi \Leftrightarrow [\ \exists S \subseteq \mathcal{S}_{\subseteq A} : s[\![\alpha]\!]S$ and $\forall s' \in S : s' \models \varphi\ ]$. This may be intuitively more appealing: from the given state s, we can reach a set of of states S where φ holds everywhere ('simultaneously'). Our treatment of the dynamic operators is similar to that in dynamic logic [Pel87, Gol92].

We lift equivalence of worlds in a state to equivalence of states and to equivalence of sets of states. This is necessary because sets of states will occur as worlds in definition 5 of local interpretation, so that access between such worlds will be based upon properties of these sets of states.

Definition 4 (Equivalences of states and of sets of states)
Let $(M, w), (M, w'), (M'', w'') \in \mathcal{S}_A$, let $S, S' \subseteq \mathcal{S}_{\subseteq A}$, let $a \in A$. Then:

$$
\begin{array}{rcl}
(M, w) \sim_a (M, w') & :\Leftrightarrow & w \sim_a w' \\
(M, w) \sim_a (M'', w'') & :\Leftrightarrow & \exists v \in M : (M, v) \underline{\leftrightarrow} (M'', w'') \text{ and } (M, w) \sim_a (M, v) \\
S \sim_a S' & :\Leftrightarrow & (\forall s \in S : a \in gr(s) \Rightarrow \exists s' \in S' : s \sim_a s') \text{ and} \\
 & & (\forall s' \in S' : a \in gr(s') \Rightarrow \exists s \in S : s \sim_a s')
\end{array}
$$

In the second clause of the definition, $\underline{\leftrightarrow}$ stands for 'is bisimilar to' [BdRV01]. Bisimilarity is a notion of sameness between states that implies equivalence

of their logical descriptions (theories), though not vice versa. The implicit symmetric closure in third clause of the definition is needed to keep $\sim_a$ an equivalence relation.

We now continue with defining the *local interpretation* of knowledge actions. In the definition we use the following notations: let M be a model, then $\mathcal{D}(M)_\varphi = \{v \in \mathcal{D}(M) \mid M, v \models \varphi\}$; let $R, R' : W \to \mathcal{P}(W)$ be two binary relations from some domain W to subsets of that domain (such as $[\![\cdot]\!]$), then the **composition** $(R \circ R')$ of R and R' is defined as follows: let $v \in W, V \subseteq W$, then: $(R \circ R')(v, V) :\Leftrightarrow \exists V' : R(v, V')$ and $\forall v' \in V' : \exists V'' \subseteq V : R'(v', V'')$ and $V = \bigcup_{v' \in V'} \{V'' \mid R'(v', V'')\}$.

Definition 5 (Local interpretation of knowledge actions) *Let $\alpha \in \mathsf{KA}_A$ and $s = (M, w) \in \mathcal{S}_A$, where $M = \langle W, \{\sim_a\}_{a \in A}, V\rangle$. Let $S \subseteq \mathcal{S}_{\subseteq A}$ (and also all other sets of states in the definition). The* **local interpretation** *$[\![\alpha]\!]$ of α in s is defined by inductive cases:*

$$
\begin{array}{rcl}
s[\![?\varphi]\!]S & \Leftrightarrow & S = \{(\langle W_\varphi, \emptyset, V|W_\varphi\rangle, w)\} \\
s[\![L_B\alpha']\!]S & \Leftrightarrow & \exists S' : S = \{(\langle W', \{\sim'_a\}_{a\in B}, V'\rangle, S')\}, s[\![\alpha']\!]S', gr(W') \subseteq B \\
s[\![\alpha' \,;\, \alpha'']\!]S & \Leftrightarrow & s([\![\alpha']\!] \circ [\![\alpha'']\!])S \\
s[\![\alpha' \cup \alpha'']\!]S & \Leftrightarrow & s([\![\alpha']\!] \cup [\![\alpha'']\!])S \\
s[\![\alpha' \,!\, \alpha'']\!]S & \Leftrightarrow & s[\![\alpha']\!]S \\
s[\![\alpha' \cap \alpha'']\!]S & \Leftrightarrow & \exists S', S'' : s[\![\alpha']\!]S', s[\![\alpha'']\!]S'', \text{ and } S = S' \cup S''
\end{array}
$$

In the clause for interpreting $L_B\alpha'$, the model $\langle W', \{\sim'_a\}_{a\in B}, V'\rangle$ is defined as follows. Domain: $W' := \{S \mid \exists v \in M : v \sim_B w$ *and* $(M, v)[\![t(\alpha')]\!]S\}$*;* Valuation: *Let $s' = (\langle W_{s'}, \sim_{s'}, V_{s'}\rangle, w_{s'}) \in S' \in W'$, $p \in P$, then: $S' \in V'(p) \Leftrightarrow w_{s'} \in V_{s'}(p)$;* Access: *Let $S', S'' \in W', a \in B$, then:*

$$
\begin{array}{rcl}
S' \sim'_a S'' & \Leftrightarrow & S' \sim_a S'' \text{ and } [\ a \notin gr(S') \cup gr(S'') \Rightarrow \exists v', v'' \in M : \\
& & (M, v')[\![t(\alpha')]\!]S', (M, v'')[\![t(\alpha')]\!]S'' \text{ and } v' \sim_a v''\ ]
\end{array}
$$

If the interpretation of α in s is not empty, we say that α is **executable** in s. For all actions except concurrent knowledge actions it is more intuitive to think of their interpretation as a relation between states than as a relation between a state and a set of states: if $s[\![\alpha]\!]\{s'\}$, we like to think of s' as the result of executing α in s. The notational abbreviation $s[\![\alpha]\!]s' :\Leftrightarrow s[\![\alpha]\!]\{s'\}$ allows us to keep using this helpful intuition. Further, if the interpretation is functional as well, write $s[\![\alpha]\!]$ for the unique s' such that $s[\![\alpha]\!]s'$. If this is the case for arbitrary s, we call α a **state transformer**. Note that tests and learning actions are state transformers.

To execute an action $L_B\alpha'$ in a state s, we do not just have to execute the *actual* action α' in the *actual* state s, but also any *other* action of the same type $t(\alpha')$ as α' in any *other* state s' that is relevant for the agents in B: any state that is $\sim_B$ accessible from s. The results are the *worlds* in the state that results from executing $L_B\alpha'$ in s. Such worlds (that are sets of states) can be distinguished from each other by an agent $a \in B$ in two cases: either a occurs in

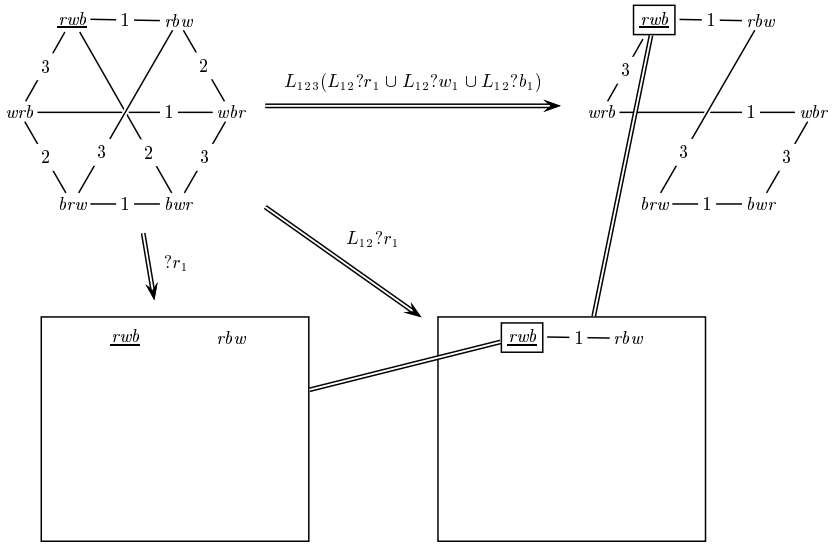

Figure 4: Stages in the computation of $(Hexa, rwb)[\![\mathsf{showcard}]\!]$. The linked frames visually emphasize identical objects: large frames enclose states that reappear as small framed worlds in the next stage of the computation.

both sets of states and he cannot distinguish between them, or a doesn't occur in these sets of states, so that he cannot distinguish between them anyway, and as well he could not distinguish their $[\![t(\alpha')]\!]$-origins either. The constraint $gr(W') \subseteq B$ in the clause for $L_B\alpha'$ is for technical reasons [vD01a].

4.4 Examples

We compute the interpretation of showred in $(Hexa, rwb)$. The interpretation of nocard, endmove and win proceeds along similar lines but is simpler. For an example interpretation of a truly concurrent action, see [vD01c].

Example 13 (Local interpretation of showcard) *In state $(Hexa, rwb)$, player 1 shows his red card (only) to player 2:* $L_{123}(!L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1)$.

All sets of states occurring in this computation are *singleton* sets, so assume from now that $[\![\cdot]\!]$ is a relation between states, not between states and sets of states.

We apply clause L_B of definition 5. To interpret showcard $= L_{123}(!L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1)$ in $\underline{rwb} = (Hexa, rwb)$, we *first* interpret the type $L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1$ of $!L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1$ in any state of $Hexa$ that is $\{1,2,3\}$-accessible from rwb, i.e. in all states of $Hexa$. The resulting states will

make up the domain of $\underline{rwb}[\![\mathsf{showcard}]\!]$ (we may write $\underline{rwb}[\![\mathsf{showcard}]\!]$ because showcard is a learning action and therefore a state tranformer). We *then* compute access on that domain, and, *finally*, the required image is $\underline{rwb}[\![!L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1]\!]$. We start with the first.

Action $L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1$ has a nonempty interpretation in all states of $Hexa$. We give two examples. Apply clause $\cup$ of definition 5 (assuming associativity again): $L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1$ can be interpreted in $\underline{rwb}$ because $L_{12}?r_1$ can be interpreted in that state. Similaryly, $L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1$ can be interpreted in $\underline{brw}$ because $L_{12}?b_1$ can be interpreted in that state. We compute the first.

Again, we apply clause L_B of definition 5. To interpret $L_{12}?r_1$ in $\underline{rwb}$, we interpret $?r_1$ in any state of $Hexa$ that is $\{1,2\}$-accessible from rwb, i.e. in all states of $Hexa$. The interpretation is not empty when 1 holds red, i.e. in $\underline{rwb}$ and in $\underline{rbw}$. We compute the first.

We now apply clause $?\varphi$ of definition 5. The state $\underline{rwb}[\![?r_1]\!]$ is the restriction of $Hexa$ to worlds where r_1 holds, i.e. rwb and rbw, with empty access, and with point rwb. Figure 4 pictures the result.

Having unravelled the interpretation of showcard to that of its atomic constituents, we can now start to compute access on the intermediate stages of our interpretation. The state $\underline{rwb}[\![?r_1]\!]$ is one of the worlds of the domain of $\underline{rwb}[\![L_{12}?r_1]\!]$ (as visualized in Figure 4 by linked frames) and is also the point of that state. The other world is $\underline{rbw}[\![?r_1]\!]$. As agent 1 does not occur in either of these, and their origins under the interpretation of $?r_1$ are the same to him ($rwb \sim_1 rbw$ in $Hexa$), therefore $\underline{rwb}[\![?r_1]\!] \sim'_1 \underline{rbw}[\![?r_1]\!]$ in $\underline{rwb}[\![L_{12}?r_1]\!]$. For the same reason, both worlds are reflexive for both 1 and 2 in $\underline{rwb}[\![L_{12}?r_1]\!]$. Further note that $\underline{rwb}[\![?r_1]\!] \not\sim'_2 \underline{rbw}[\![?r_1]\!]$, because in $rwb \not\sim_2 rbw$ in $Hexa$. The valuation of atoms does not change. Therefore world $\underline{rwb}[\![?r_1]\!]$ is named rwb and world $\underline{rbw}[\![?r_1]\!]$ is named rbw in Figure 4.

Similarly to the computation of $\underline{rwb}[\![L_{12}?r_1]\!]$, we compute the five other states where 1 and 2 learn 1's card. These form the domain of $\underline{rwb}[\![\mathsf{showcard}]\!]$. For a more detailed computation of access on that model, see [vD01a].

The point of $\underline{rwb}[\![\mathsf{showcard}]\!]$ is $\underline{rwb}[\![L_{12}?r_1]\!]$, because, applying the clause for '!' in definition 5, $\underline{rwb}[\![!L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1]\!] = \underline{rwb}[\![L_{12}?r_1]\!]$. We have now completed the interpretation. See Figure 4.

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.

4.5 Theory

We mention a few relevant theoretical results that have been proven for the language without concurrency [vD00, vD01a] but that can be easily extended to the more general language under consideration.

The class of equivalence states is closed under execution of knowledge actions; this trivially follows from definition 5. Various algebraic properties hold, such as e.g. $(\alpha \cup \alpha') \cup \alpha^* = \alpha \cup (\alpha' \cup \alpha^*)$ and $(\alpha \cup \alpha')\ ;\ \alpha^* = (\alpha\ ;\ \alpha^*) \cup (\alpha'\ ;\ \alpha^*)$.

It is indeed the case that concrete actions have a functional interpretation. An action's interpretation is included in that of its type: $[\![\alpha]\!] \subseteq [\![t(\alpha)]\!]$. If (s, S') is in the interpretation of α, then there is a concretization of α with precisely that interpretation: $s[\![\alpha]\!]S' \Rightarrow \exists \beta \in C(\alpha) : s[\![\beta]\!]S'$. Every action is equivalent to nondeterministic choice between all its concretizations: $[\![\alpha]\!] = [\![\bigcup_{\beta \in C(\alpha)} \beta]\!]$. The main theorems of interest (for proofs, see [vD01a]) are:

Theorem 1 (Bisimilarity implies modal equivalence) *Let $\varphi \in \mathcal{L}_A$. Let s, s' be equivalence states. If $s \underline{\leftrightarrow} s'$, then $s \models \varphi \Leftrightarrow s' \models \varphi$.*

Theorem 2 (Action execution preserves bisimilarity) *Let $\alpha \in \mathsf{KA}_A$. Let s, s' be equivalence states. For each set of ($\mathcal{S}_{\subseteq A}$) states S there is a set of states S' and a bijection $f : S \to S'$ such that: If $s \underline{\leftrightarrow} s'$ and $s[\![\alpha]\!]S$, then $s'[\![\alpha]\!]S'$ and for all $s'' \in S : s'' \underline{\leftrightarrow} f(s'')$.*

A corollary of theorem 2 is the following:

Corollary 3 *Let s, s' be equivalence states and α a state transformer that is executable in s. If $s \underline{\leftrightarrow} s'$, then $s[\![\alpha]\!] \underline{\leftrightarrow} s'[\![\alpha]\!]$.*

The *axiomatization* of the language has not been completed. Part of the problem is that there are well-formed $\mathcal{L}_A$ formulae that are uninterpretable. This is a result of the restriction of our semantics to *equivalence* ($S5$) states. Finally, it is as yet unclear how the *expressive power* of the current language relates to that of the language without $\cap$, as presented in [vD00, vD01a].

4.6 Action frames

We have described game actions as knowledge actions, i.e. in a logical language. We then computed the effect of a game action by determining the interpretation of that knowledge action. This may be considered a roundabout way to proceed: it is like working with the characteristic formula of a game state, instead of the Kripke state that this formula describes. Why not find a direct semantic representation of game actions? We outline the approach in [vD00]:

Just as a game state is represented by an equivalence state, a game action can be represented by an equivalence 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* a Kripke model and a Kripke frame: it resembles the computation of a direct product.

Definition 6 (Action frame) *Let $s = (M, w) = (\langle W, \{\sim_a\}_{a \in A}, V\rangle, w)$ be an equivalence (game) state. An* **action frame** *(relative to that state) is a pointed equivalence frame $\mu = (N, q) = (\langle Q, \{\sim_a\}_{a \in B}\rangle, q)$ such that $Q \subseteq \mathcal{P}(W)$, $B \subseteq A$, and $q \in Q$.*

Definition 7 (Executing an action frame) *Let $s = (M, v) = (\langle W, \{\sim_a\}_{a \in A}, V\rangle, v)$ be an equivalence state and let $\mu = (N, r) = (\langle Q, \{\sim_a\}_{a \in B}\rangle, r)$ be an action frame for s. The state $s \otimes \mu$ resulting from executing μ in s is defined as follows: $s \otimes \mu := (\langle W', (\sim_a)_{a \in B}, V'\rangle, (v, r))$, where $W' = \{(w, q) \in W \times Q \mid w \in q\}$,*

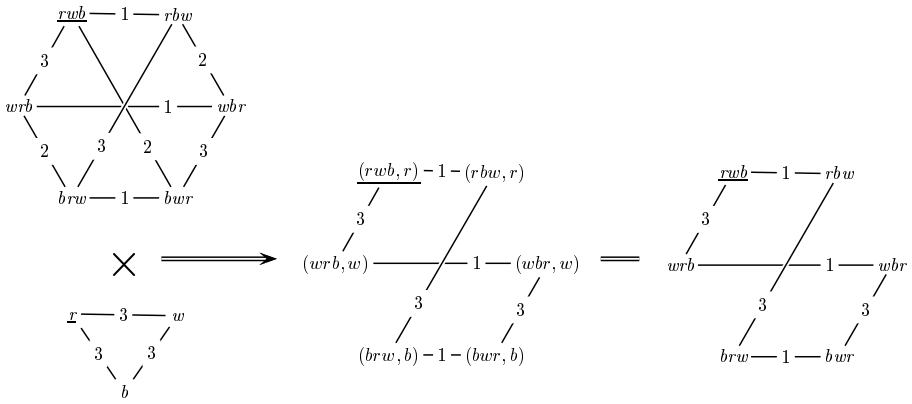

Figure 5: Executing action *Showcard* in state $(Hexa, rwb)$

$(v, r) \in W'$, *and* $\forall a \in B : \forall w, w' \in W : \forall q, q' \in Q : (w, q) \sim_a (w', q') :\Leftrightarrow (w \sim_a w'$ *and* $q \sim_a q')$, *and* $V'_{(w,q)} := V_w$

We then define a **game action** as an action frame that satisfies some additional frame properties [vD01b], corresponding to the roles of players. The general idea of this product construction is, that the next state of the game consists of all pairs (w, q) such that (N, q) 'could also have been' the action and (M, w) 'could also have been' the state, plus access appropriately defined.

Example 14 (Showcard) *Showcard is the game action* $(Q, \{rwb, rbw\}) = (\langle\{\{rwb, rbw\}, \{wrb, wbr\}, \{brw, bwr\}\}, \{\sim_1, \sim_2, \sim_3\}\rangle, \{rwb, rbw\})$ *where* $\sim_1$ *and* $\sim_2$ *are the identity '=' on* Q*, and* $\sim_3$ *is the universal relation* $Q \times Q$*. The execution of Showcard in* $(Hexa, rwb)$ *is pictured in Figure 5. In the figure, we abbreviate* $\{rwb, rbw\}$ *as* r*,* $\{wrb, wbr\}$ *as* w*, and* $\{brw, bwr\}$ *as* b*.* Domain*: e.g.* $(rwb, \{rwb, rbw\}) \in Q$ *because* $rwb \in \{rwb, rbw\}$*.* Access*: e.g.* $(rbw, r) \sim_3 (brw, b)$ *because* $rbw \sim_3 brw$ *and* $r \sim_3 b$ *(if player 3 holds white, he cannot distinguish 1 showing red from 1 showing blue).* Valuation*: Because valuations do not change (cards do not change hands) the names of world remain card deals, as on the right in Figure 5. Note that game action Showcard and knowledge action* showcard*, as in Figure 3 on page 6 and in Example 9 on page 9, induce the same relation between states.*

There is a precise relation between action frames and knowledge actions that are state transformers: instead of the local interpretation $[\![\alpha]\!]$ of a state transformer α we can also define a **product interpretation** $[\![\alpha]\!]^{\otimes}$ of that action as follows: $[\![\alpha]\!]^{\otimes} = (\langle T, \{\sim_a\}_{a \in A}\rangle, \alpha)$, where $T = \{\beta \mid \beta \in C(t(\alpha))\}$ (T is the set of all concrete actions of the same type as α), and $\sim_a$ is defined by induction on the structure of actions (no details) such that, e.g., in the $Hexa$ example player 1 showing the red card to player 2 is the same for player 3 as 1 showing the blue card or the white card to 2: $L_{123}(!L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1) \sim_3$

$L_{123}(L_{12}?r_1 \cup !L_{12}?w_1 \cup L_{12}?b_1) \sim_3 L_{123}(L_{12}?r_1 \cup L_{12}?w_1 \cup !L_{12}?b_1)$. Naturally for player 1 and 2, all these actions are different. Observe that the resulting frame is isomorphic to the game action in Figure 5.

Instead of using concrete actions as worlds, in $[\![\alpha]\!]^{\otimes}$, we may also, given a state, replace them by the set of worlds in that state where they can be executed (which is merely another way of saying that these worlds satisfy the *preconditions* of these actions). That completes the link between game actions (action frames) and knowledge actions. In [vD00] we prove the following result for the restricted action language without concurrency: Let s be an equivalence state, α a knowledge action that is a state transformer and that can be executed in that state, and let μ be the game action computed from α by the procedure above. Then:

$$s[\![\alpha]\!] \underline{\leftrightarrow} s \otimes [\![\alpha]\!]^{\otimes} \cong s \otimes \mu$$

We conjecture that this result can be generalized for the extended language (access between actions is more complex for concurrent actions). The notion of an action as a semantic object is similar to that in [BMS00, Bal99]. For the relation to Baltag, see also [vD00].

5 Knowledge actions in Cluedo

In section 2 we have seen that in Cluedo there are only four sorts of action, nocard, showcard, endmove and win. Compared to *Hexa*, instead of being asked for 'your card', which is basically the same as being asked for one of all (three) cards, in Cluedo you are being asked for one out of three (out of 21). Compared to *Hexa*, where 'winning' was defined as knowing the deal of cards, in Cluedo you only have to know the cards 'on the table': the murder cards. Let δ^0_d be the atomic description of the cards on the table for deal d of cards. This is a conjunction of 21 atoms or negations of atoms. For example, if 'Scarlett has done it with a knife in the kitchen' we get $\delta^0_d = scarlett_0 \wedge \neg plum_0 \wedge \neg white_0 \wedge .... \wedge knife_0 \wedge ...kitchen_0....$ Let $K_a\delta^0 = K_a\delta^0_d \vee K_a\delta^0_{d'} \vee ...$ express that a knows the murder cards. We suggest the following parameterized descriptions of the four action sorts, their interpretation will by now be obvious:

$\mathsf{nocard}^{a,\{c,c',c''\}}_b$	$L_{123456}?(\neg c_b \wedge \neg c'_b \wedge \neg c''_b)$
$\mathsf{showcard}^{a,\{c,c',c''\}}_{b,c}$	$L_{123456}(!L_{ab}?c_b \cup L_{ab}?c'_b \cup L_{ab}?c''_b)$
$\mathsf{endmove}^a$	$L_{123456}?\neg K_a\delta^0$
win^a	$L_{123456}?K_a\delta^0$

If we do not use the parameters (as we have done throughout until now) assume that they can take any value. Actual 'moves' in Cluedo are composed of various of these action constituents. A play of the game is a sequence of such moves, where the last move ends with a win action. Therefore, we can describe an entire play of the Cluedo game as a single KA_{123456} action; in the following, α^{*i} means a sequence of zero to i actions α, and α^* a arbitrary finite sequence of actions

α (where $(\alpha^0 \ ; \ \beta) = (\beta \ ; \ \alpha^0) = \beta$):

$$(\mathsf{nocard}^{*5} \ ; \ \mathsf{showcard}^{*1} \ ; \ \mathsf{endmove})^* \ ; \ (\mathsf{nocard}^{*5} \ ; \ \mathsf{showcard}^{*1} \ ; \ \mathsf{win})$$

This way we can describe an entire *strategy profile* (we have combined all actions of all players, not the actions of one player). A single player's strategy is then the subset of a given play that consists of the actions of that player (i.e., all actions with the same high index, as in $\mathsf{endmove}^a$). In both cases we disregard pruned subtrees of the game tree, and the choices that players are formally required to make there in a fully specified strategy.

6 Knowledge games

In Cluedo a finite number of cards is dealt over a finite number of players, and actions consist of either questions and answers about cards, or are announcements about (not) winning. We call Cluedo and similar games *knowledge games*. A knowledge game is informally defined by a deal of cards over players, a set of possible game actions of sort nocard, showcard, endmove, and nowin, and a protocol to determine the order of actions. 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. Although cards do not change hands, *knowledge* about cards does change during the game, and *only* knowledge: therefore the term *knowledge* games.

Knowledge games are competitive games of imperfect information, where the only final outcomes are that players can win or lose. Optimal strategies for the players will obviously be mixed strategies. See e.g. [OR94, Bin92]. Is the first player most likely to win a knowledge game? What is the value of Cluedo? Much work has to be covered before answers can be given. We present some of the obstacles on the way there.

We have already given a precise formal description of actions. One can also give a precise formal description of initial game states. For any given deal $d \in A^C$ of cards over players, the **initial knowledge game state** is the equivalence state $(\langle W, \{\sim_a\}_{a\in A}, V\rangle, d)$ with W the set of all deals where each player has the same number of cards as in deal d, and where deals d, d' are the same for player a ($d \sim_a d'$) if he holds the same cards in both, and V is the atomic description of deals (where the set of atoms P can be seen as the product $C \times A$: an atom c_a describes that agent a holds card c). See [vDvdHK01, vdHV01]. Now from a given game state and an 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 an action sequence. We *must* make a rather nice observation here: at *any* state of the game, the model underlying that state is *commonly known by all players*. The *only* thing they don't know, is 'where they live' in that model, i.e. what the point of the model is, the actual world.

The complexity of a game state is determined by the number of its (non-bisimilar) worlds. In general the execution of an action in a state may lead to a more complex state. In particular this is the case for showcard actions: if in a given world a player can choose which card to show, the next state will have as many successors of that world as there are choices. It is interesting to observe that *even* if the first action in a Cluedo game is a showcard action, the next state is not more but less complex, namely only $\frac{9}{21}$th the size of the initial state [vD00]. The other sorts of actions are public announcements, so obviously reduce the complexity. Had the rules been slightly different, a showcard action would have led to an increase. Does this explain why Cluedo is 'playable'?

Yet another matter of interest here: endmove actions, where you publicly announce that you cannot win yet, may result in information change. In Example 6 on page 7 we have even seen a case where a player could win because another player couldn't win. In that particular example, yet another player (3) could have won anyway, before the endmove action. Is there a knowledge game state where *none* of the players can win but where, as a result of one of them announcing that, a player can win? In technical words: Let $K\delta := \bigvee K_i\delta$ (one of the players knows the deal of cards – or some derived property, such as the cards on the table). Is there a game state s such that $\neg K\delta$ is an *unsuccessful update*: $s \models \neg K\delta$ and $s[\![L_{123}\neg K\delta]\!] \models K\delta$? An answer to that question would also be of strategic relevance.

Preference relation Because both actions and game states have so much internal structure, this is different from the usual picture [OR94] where these are abstract objects, and where payoffs and/or preferences are stipulated. In knowledge games, one has to *compute* the preference relation. You may prefer one suspicion over another, if its expected answers may 'on the long run' lead to a greater reduction *for you* in possible card deals, which is more or less the same as saying that you prefer the largest refinement of the current partition on the domain, as induced by *your* equivalence relation $\sim_a$. However, it is not so clear by what measure we should compare partitions. Smallest number of card deals? Smallest number of nonsimilar worlds? Games like Mastermind face similar questions that are not entirely answered, notwithstanding great efforts to solve them [Neu82]. It is not yet clear whether, e.g., 'asking for three cards that you do not have' is to be preferred over 'asking for three cards of which you hold one yourself'.

Optimal strategy Now you're *really* playing Cluedo. It's your move, you know the murderer and the weapon but you still hesitate between the kitchen and the conservatory. Naturally, all players are perfectly rational. You know that unless you guess the murder cards *now*, some other player may do so in the next round, and may win. Should you guess or not? (Note that unsuccessful guesses are endmove sorts of action. So this does not add an epistemic complexity.) Optimal strategies will be a function of the preference relation for the players. The preference relation was induced by the partition refinements as a consequence of actions. These partition refinements can be computed from

those actions and the current game state. It is therefore clear that in this article we have been laying some foundation stones for further game theoretical research on knowledge games.

7 Historical note

Cluedo was invented in 1943 by Anthony E. Pratt, a solicitor's clerk, and (his wife) Elva Pratt. Anthony Pratt said to have invented the game when he was temporarily laid off because of World War II and was instead doing a, mostly boring, fire brigade duty. Elva Pratt devised the board. The Pratts' original version was called 'Murder'. It had ten weapons instead of six, and some suspects had other names. In 1949 Cluedo was launched by Waddingtons Games in the UK. In the USA the game is called Clue. Apart from the original Cluedo, there are various other versions available. There is now even a Harry Potter 'Mystery at Hogwarts Game', that is obviously Cluedo-inspired.

Anthony Pratt died in 1994, in obscurity. His death only became generally known in 1996, after a public appeal by Waddingtons. His tombstone reads 'inventor of Cluedo'.

References

[Bal99] A. Baltag. A logic of epistemic actions. Manuscript, 1999.

[BdRV01] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, Cambridge, 2001. Cambridge Tracts in Theoretical Computer Science 53.

[Bin92] K. Binmore. *Fun and Games*. D.C. Heath, Lexington MA, 1992.

[BMS00] A. Baltag, L.S. Moss, and S. Solecki. The logic of public announcements, common knowledge and private suspicions. Revised manuscript, originally presented at TARK 98, submitted for publication, 2000.

[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, 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, 2 edition, 1992. CSLI Lecture Notes No. 7.

[HKT00] D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic* Foundations of Computing Series. MIT Press, Cambridge MA, 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, 1995.

[Neu82] E. Neuwirth. Some strategies for mastermind. *Zeitschrift für Operations Research*, 26:257–278, 1982.

[OR94] M.J. Osborne and A. Rubinstein. *A Course in Game Theory.* MIT Press, Cambridge MA, 1994.

[Par85] R. Parikh. The logic of games and its applications. In M. Karpinski and J. van Leeuwen, editors, *Topics in the theory of computation* Annals of Discrete Mathematics 24, pages 111–139, Amsterdam, 1985. Elsevier Science.

[Par87] R. Parikh. Knowledge and the problem of logical omniscience. In R. Zas and M. Zemankova, editors, *Proceedings of the 2nd international symposium on methodologies for intelligent systems*, pages 432–439, Amsterdam, 1987. North Holland.

[Pau00] M. Pauly. Game logic for game theorists. Technical report, CWI, Amsterdam, 2000. CWI Technical Report INS-R0017.

[Pel87] D. Peleg. Concurrent dynamic logic. *Journal of the ACM*, 34(2):450–479, 1987.

[Pla89] J.A. Plaza. Logics of public communications. In M.L. Emrich, M.S. Pfeifer, M. Hadzikadic, and Z.W. Ras, editors, *Proceedings of the 4th International Symposium on Methodologies for Intelligent Systems*, pages 201–216, 1989.

[vD99] H.P. van Ditmarsch. The logic of knowledge games: showing a card. In Eric Postma and Marc Gyssens, editors, *Proceedings of the BNAIC 99*, pages 35–42, Maastricht University, 1999.

[vD00] H.P. van Ditmarsch. *Knowledge games.* PhD thesis, University of Groningen, 2000. ILLC Dissertation Series DS-2000-06.

[vD01a] H.P. van Ditmarsch. Descriptions of game actions. *Journal of Logic, Language and Information*, 2001. To appear.

[vD01b] H.P. van Ditmarsch. Knowledge games. *Bulletin of Economic Research*, 53(4):249–273, 2001.

[vD01c] H.P. van Ditmarsch. The semantics of concurrent knowledge actions. In M. Pauly and G. Sandu, editors, *ESSLLI 2001 workshop on Logic and Games*, 2001.

[vdHV01] W. van der Hoek and L.C. Verbrugge. Epistemic logic: a survey. submitted to Game theory and applications VIII, 2001.

[vDvdHK01] H.P. van Ditmarsch, W. van der Hoek, and B.P. Kooi. Descriptions of game states. In I. van Loon, G. Mints, and R. Muskens, editors, *Proceedings of LLC9*, Stanford, 2001. CSLI Publications. To appear.