Department of Computer Science,
University of Otago

Technical Report OUCS-2001-03

Descriptions of game actions

Author:H.P. van Ditmarsch

Status: Accepted for the Journal of Logic, Language and Information. Revised version Oct 2001.

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

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

Descriptions of game actions

Hans P. van Ditmarsch (hans@cs.otago.ac.nz)
Computer Science, University of Otago, PO Box 56, Dunedin, New Zealand

Abstract. To describe simultaneous knowledge updates for different subgroups we propose an epistemic language with dynamic operators for actions. The language is interpreted on equivalence states ($S5$ states). The actions are interpreted as state transformers. Two crucial action constructors are *learning* and *local choice*. Learning is the dynamic equivalent of common knowledge. Local choice aids in constraining the interpretation of an action to a functional interpretation (state transformer). Bisimilarity is preserved under execution of actions. The language is applied to describe various actions in card games.

Keywords: multiagent systems, modal logic, dynamic epistemics, action language

1. Introduction

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' (Fagin et al., 1995; Parikh, 1987). In the 'runs and systems' approach of (Fagin et al., 1995), an update is a functional relation between two global states of an interpreted *system*, and a *run* is a sequence of such transformations. Such a global state corresponds in a natural way to a Kripke state where all relations are equivalence relations: an equivalence (or $S5$) state. They do not introduce an object (logical) language for both updates and epistemic statements. An early example of such a dynamic epistemic language is the elegant logic of public announcements as presented in (Plaza, 1989). Plaza models public announcements as binary operators that have a dynamic interpretation. An integrated approach including announcements to subgroups has been put forward in (Gerbrandy and Groeneveld, 1997). Gerbrandy's thesis, (Gerbrandy, 1999), 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, (Baltag et al., 2000) also treats epistemic actions in general. This is still being extended to an entire framework for dynamic epistemic logic in (Baltag, 1999).

Our research (van Ditmarsch, 1999; van Ditmarsch, 2000; van Ditmarsch, 2001a) 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 equivalence states. We base ourselves on standard Kripke semantics. Our contribution consists of a concise language to describe equivalence state transformations, and a new relational semantics to interpret these actions, namely one based on standard Kripke semantics. We describe and interpret combined updates for different subgroups by *local interpretation*: first complete the interpretation 'locally', for a given subgroup only, and then use that to determine the *global* interpretation, for the entire group of agents. Apart from the usual programming constructs: test, sequential execution, and nondeterministic choice (Harel, 1984; Harel et al., 2000; Goldblatt, 1992), we introduce as well: *learning* and *local choice*. Learning is the dynamic equivalent of common knowledge (of *reflexive* common knowledge (Meyer and van der Hoek, 1995)) and is related to truthful (factive) updating (Gerbrandy, 1999). Local choice aids in making the interpretation of an action functional. We start with some examples to illustrate the need for these operations.

Hexa 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 is modelled by a (hexagonal) equivalence state $(Hexa, rwb)$. See Figure 1. There are six deals of three cards over three players. The model $Hexa$ consists of these deals. In 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. The following actions can be executed in this state $(Hexa, rwb)$:

EXAMPLE 1 (table). *Player 1 puts the red card (face up) on the table.*

EXAMPLE 2 (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 3 (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 blue". Player 3 notices that the question is answered, but cannot hear the answer.*

We assume that only the truth is told. In show and whisper, we assume that it is publicly known what 3 can and cannot see or hear.

Figure 1 also pictures the states that result from updating the current state $(Hexa, rwb)$ with the information contained in the three actions. In table it suffices to eliminate some worlds: after 1's action, the four deals of cards where 1 does not hold red are eliminated. It

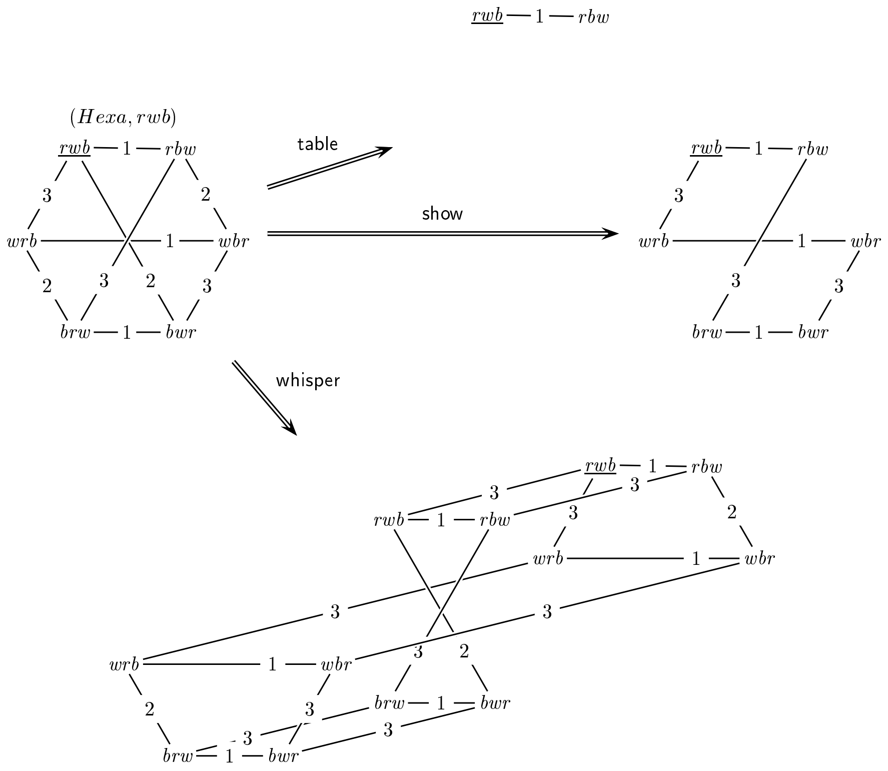

Figure 1. The results of executing table, show, and whisper in the state $(Hexa, rwb)$ where 1 holds red, 2 holds white and 3 holds blue. The points of the states are underlined. Worlds are named by the deals that (atomically) characterize them. Assume reflexivity and transitivity of access.

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 table, all three players *learn* that player 1 holds the red 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': indeed in *table* it becomes common knowledge that 1 holds red; however, imagine that instead of putting his card on the table, 1 had said to 2: 'You don't know that I have the red card' (interpreted as: 'I have the red card and you don't know that'). At the moment of utterance, this statement can be truthfully made in the worlds rwb and rbw of $Hexa$, so it results in the same state as execution of table. However, in that state is it *not* common knowledge that 2 doesn't know that 1 has red. To the contrary: after this announcement 2 knows that 1 has red.

We continue our conceptual analysis. 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. This can be expressed by the 'local choice' operator. The need for such an operator becomes more apparent in the case of the action $\mathsf{whisper}$: the action of 1 whispering in 2's ear a card that he doesn't have, has two different executions in any given state. 'Local choice' fixes one of those executions, in this case '1 and 2 learn that 1 doesn't have blue'.

In section 2 we define the logical language $\mathcal{L}_A$ and the knowledge actions KA_A. We give descriptions of the card game actions in the introduction. In section 3 we define the interpretation of $\mathcal{L}_A$. We also give some other game action descriptions. In section 4 we present some theoretical results. In section 5 we discuss extensions of $\mathcal{L}_A$ and compare our research to that of others.

2. Knowledge actions

To a standard multiagent epistemic language with common knowledge for a set A of agents and a set P of atoms (Meyer and van der Hoek, 1995; Fagin et al., 1995), 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*

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

Formula $K_a\varphi$ stands for a **knows** φ, $C_B\varphi$ stands for *group* B **commonly know** φ, and $[\alpha]\varphi$ stands for φ *holds after every execution of action* α. Other propositional connectives and modal operators are defined by abbreviations (let $p \in P$): $\varphi \vee \psi := \neg(\neg\varphi \wedge \neg\psi)$, $\varphi \to \psi := \neg\varphi \vee \psi$, $\varphi \leftrightarrow \psi := (\varphi \to \psi) \wedge (\psi \to \varphi)$, $E_B\varphi := \bigwedge_{a \in B} K_a\varphi$, $\top := p \vee \neg p$, $\bot := p \wedge \neg p$. Outermost parentheses of formulae are deleted whenever convenient. As we may generally assume an arbitrary P, write $\mathcal{L}_A$ instead of $\mathcal{L}_A(P)$. The set of agents A is called the **public**.

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:*

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

Outermost parentheses of actions are deleted whenever convenient. We generally write KA_A instead of $\mathsf{KA}_A(P)$. We name knowledge actions after their main constructor. Action $?\varphi$ is a **test**. The program constructor L_B is called the **learning** operator. $L_B\alpha$ stands for *group B learn α*. Instead of $L_{\{1,2,\ldots,i\}}$ write $L_{12\ldots i}$. Operator ';' stands for **sequential execution**; $\alpha\ ;\ \alpha'$ means *first execute α and then execute α'*. Operator '$\cup$' stands for **nondeterministic choice**; $\alpha \cup \alpha'$ means *execute either α or α'*. Operator '!' is called **local choice**; $\alpha\ !\ \alpha'$ means *from α and α', choose α (locally)*. Instead of $\alpha\ !\ \alpha'$, write either $!\alpha \cup \alpha'$ or $\alpha' \cup\ !\alpha$. This will make the relation between local and nondeterministic choice clearer, as we will also see in the examples. In combination with learning, local choice helps to constrain the interpretation of a (possibly nondeterministic) action to a functional interpretation.

The subclass of KA_A generated by all constructors except '!' is called the **knowledge action types** or **action types** (for A and P). The subclass of KA_A generated by all constructors except $\cup$ is called the **concrete knowledge actions** or **concrete actions** (for A and P).

We now give some examples of actions, related to the model *Hexa* from the introductory section. Assume (nine) atoms c_a describing that player a holds card c:

EXAMPLE 4 (Knowledge action for table). *Player 1 puts the red card on the table: $L_{123}?r_1$.*

EXAMPLE 5 (Knowledge action type for show). *Player 1 shows (only) player 2 his card: $L_{123}(L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1)$.*

Assume associativity of $\cup$ (see proposition 2, in section 4). The action 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)'. This *almost* describes the action show from example 2, where the *red* card was shown. Almost, but not quite: show can *not* be executed in a state $(Hexa, brw)$ where 1 holds blue, whereas the action of showing a card *can* be executed in that state. The last is a *knowledge action type*, and the first a *concrete knowledge action* (of that type).

EXAMPLE 6 (Knowledge action for show). *Player 1 shows (only) player 2 his red card:* $L_{123}(!L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1)$.

The type of show is $L_{123}(L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1)$. We must be more precise now and choose it, e.g., to be $L_{123}((L_{12}?r_1 \cup L_{12}?w_1) \cup L_{12}?b_1)$. We now express what is known to agents 1 and 2, but not to agent 3, from the two choices to be made: 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 the other notation that becomes $L_{123}(!(!L_{12}?r_1 \cup L_{12}?w_1) \cup L_{12}?b_1)$ and assuming associativity again we get $L_{123}(!L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1)$. There are two other concrete actions of the same type. These 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).

EXAMPLE 7 (Knowledge action type for whisper). *Player 1 whispers in 2's ear a card that he (1) doesn't have:* $L_{123}(L_{12}?\neg r_1 \cup L_{12}?\neg w_1 \cup L_{12}?\neg b_1)$.

EXAMPLE 8 (Knowledge action for whisper). *Player 1 whispers in 2's ear "I don't have blue":* $L_{123}(L_{12}?\neg r_1 \cup L_{12}?\neg w_1 \cup\ !L_{12}?\neg b_1)$.

In the case of whispering a card that you do not have, the three options are *not* having a card, instead of having a card. The action whisper is one of three concrete actions of that type.

Even though 3 knows that 1 can only have whispered 'not white' or 'not blue', this is not *publicly* known, e.g. 2 doesn't know that 3 knows that. The knowledge action describes the publicly known alternatives, therefore all three.

These example actions each 'involve' precisely all agents for which access is defined in $Hexa$. This is not accidental, because it apparently corresponds to our intuition of what a fully specified action is: for each agent occurring in a state of knowledge, we have to specify how his or her knowledge is updated.

We relate an action type to all concrete actions of that type by a simple operation $C : \mathsf{KA}_A \to \mathcal{P}(\mathsf{KA}_A)$ (C for *C*oncrete). It is inductively defined with crucial clause $C(\alpha \cup \alpha') = \{\beta ~!~ \beta' \mid \beta \in C(\alpha), \beta' \in C(\alpha')\} \cup \{\beta' ~!~ \beta \mid \beta \in C(\alpha), \beta' \in C(\alpha')\}$ (tests are concrete actions and the remaining clauses merely carry on results). We relate a concrete action to its type by the simple operation $t : \mathsf{KA}_A \to \mathsf{KA}_A$ (t for *t*ype). The crucial clause in the inductive definition of t is $t(\alpha ~!~ \alpha') = t(\alpha) \cup t(\alpha')$ (tests are types and the remaining clauses merely carry on results). In the next section we will see that the interpretation of an action $L_B\alpha$ is defined in terms of the interpretation of $t(\alpha)$. In section 4 we will show that the interpretation of an action α is equivalent to nondeterministic choice between all its concretizations: $\alpha = \bigcup_{\beta \in C(\alpha)} \beta$.

3. Local interpretation

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** (also known as an $S5$ / $S5_n$ / $S5_A$ 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 $s = (M, w)$, instead of $w \in \mathcal{D}(M)$ 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 '$P$'. We write either *$s$ is an equivalence state* or, if the context requires more precision, $s \in \mathcal{S}_A$ ($s \in \mathcal{S}_{\subseteq A}$).

The semantics of $\mathcal{L}_A$ (on equivalence models) is defined as usual (Meyer and van der Hoek, 1995), plus an additional clause for the meaning of dynamic operators. The interpretation of a dynamic operator is a relation between equivalence states (see also definition 5). These may be (and generally are) states for *different* groups of agents.

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.*

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

The notion $\langle\alpha\rangle$ is dual to $[\alpha]$ and is defined as $s \models \langle\alpha\rangle\varphi \Leftrightarrow \exists s' \in \mathcal{S}_{\subseteq A} : s[\![\alpha]\!]s'$ and $s' \models \varphi$.

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

DEFINITION 4 (Equivalence of states). *Let (M, w), (M, w'), (M'', w'') $\in \mathcal{S}_A$, let $a \in A$. Then:*

$$\begin{array}{rll}
(M, w) \sim_a (M, w') & :\Leftrightarrow & w \sim_a w' \\
(M, w) \sim_a (M'', w'') & :\Leftrightarrow & \exists v \in M : (M, v) \underline{\leftrightarrow} (M'', w'') \text{ and} \\
 & & (M, w) \sim_a (M, v)
\end{array}$$

In the second clause, $\underline{\leftrightarrow}$ stands for 'is bisimilar to', we refer to (Blackburn et al., 2001) for a definition. The overloading of the notation $\sim_a$ is justifiable: if s and s' are states for *different* (nonsimilar) underlying models, they can by definition *never* be the same for *any* agent. Therefore, when $s \sim_a s'$ we can see $\sim_a$ as the equivalence for a in the model (modulo bisimilarity) underlying both s and s'.

We now continue with defining the **local interpretation** of knowledge actions.

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

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

In the clause for $?\varphi$, W_φ is the restriction of W to the worlds where φ holds: $W_\varphi = \{v \in W \mid M, v \models \varphi\}$. 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* $s = (\langle W^s, \sim^s, V^s\rangle, w^s) \in W'$*,* $p \in P$*, then:* $s \in V'(p) \Leftrightarrow w^s \in V^s(p)$*;* Access: *Let* $s_1, s_2 \in W', a \in B$*, then:*

$$\begin{aligned} s_1 \sim'_a s_2 \;\Leftrightarrow\; & s_1 \sim_a s_2 \textit{ or } [\; a \notin gr(s_1) \cup gr(s_2) \textit{ and } \exists v_1, v_2 \in M : \\ & (M,v_1)[\![t(\alpha')]\!]s_1, (M,v_2)[\![t(\alpha')]\!]s_2 \textit{ and } v_1 \sim_a v_2 \;]. \end{aligned}$$

We start with general observations on the definition. We continue with introducing a notational abbreviation and additional terminology. After that we give examples of local interpretation.

In dynamic logic, a successful test does not change the current state. In our framework, a test removes all worlds in the current state where the test does not hold and removes all access between worlds. Therefore, a test generally results in a different state. What remains unchanged is merely the *point* of the current state.

To interpret an action $L_B\alpha'$ in a state s, we do not just have to interpret α' in s. We also have to interpret any action of the same type as α' in any other state s' that is $\sim_B$-accessible from s. The results are the *worlds* in the state that results from interpreting $L_B\alpha'$ in s. Such worlds can be distinguished from each other by an agent $a \in B$ in two cases: either a occurs in both states and he cannot distinguish between them, *or* a doesn't occur in either state but he could not distinguish their $[\![t(\alpha')]\!]$-origins.

The constraint that $gr(W') \subseteq B$ for interpreting $L_B\alpha'$ guarantees that agents in B learn *only* about groups of agents that already occur in $t(\alpha')$. Without this constraint some actions would be incorrectly interpreted, e.g., if 1 and 2 learn about an action involving 1 and 3, then in the resulting state 3 would not consider the actual state, where 2 also knows something, to be possible. The resulting state will therefore not be an equivalence state. However, also without this constraint the computations in definition 5 would result in an equivalence state, that therefore would be incorrect. (Alternatively to this semantic restriction, we could have made a syntactic restriction on the formation of $L_B\alpha'$ when defining class KA_A.)

The case α' ; α'' uses ordinary composition $\circ$ of binary relations, the case $\alpha' \cup \alpha''$ union of binary relations.

The interpretation of α' ! α'' is that of α'. However, the *function* of α' ! α'' is to constrain the interpretation of $\alpha' \cup \alpha''$ to that of α'. This is because the use of α' ! α'', even though its interpretation is compositional, depends on the context of a learning operator L_B that binds it (even: of *all* learning operators that bind it). The choice made for α' in $\alpha' \mathbin{!} \alpha''$ is *local*, i.e. for agents occurring in α' or α'' only. For agents in B not occurring in α' or α'', $\alpha' \mathbin{!} \alpha''$ is the same as $\alpha' \cup \alpha''$.

If the relation $[\![\alpha]\!]$ is functional, write $s[\![\alpha]\!]$ for the unique s' such that $s[\![\alpha]\!]s'$. Note that all actions $?\varphi$ and $L_B\alpha$ have a functional interpretation (are state transformers). An action α is **executable** in an equivalence state s, if the local interpretation of α in s is not empty.

Local interpretation is called *local*, because we only interpret the agents that are actually learning something in the action. In contrast to (Gerbrandy, 1999), we do not worry about what other agents have learnt at that stage of the interpretation, i.e. we postpone computing the *global* effects of learning. See section 5.

We illustrate definition 5 by computing in detail the interpretation of the action show in the state $(Hexa, rwb)$. After that, we remark shortly on the interpretation of table and whisper in that same state.

EXAMPLE 9 (Local interpretation of show). *In state* $(Hexa, rwb)$*, player 1 shows his red card (only) to player 2:* $L_{123}(!L_{12}?r_1 \cup\ L_{12}?w_1 \cup\ L_{12}?b_1)$.

We apply clause L_B of definition 5. To interpret $\mathsf{show} = L_{123}(!L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1)$ 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 $\underline{rwb}[\![\mathsf{show}]\!]$. 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 *any* state 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 2 pictures the result.

Having unravelled the interpretation of show to that of its atomic constituents, we can now start to compute access on the intermediate

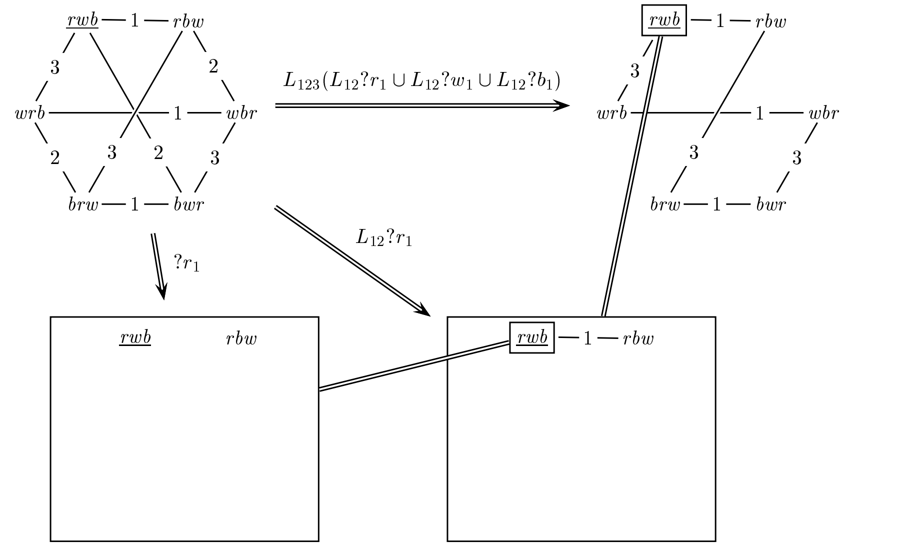

Figure 2. Stages in the computation of $(Hexa, rwb)\llbracket\mathsf{show}\rrbracket$. The linked frames visually emphasize identical objects: large frames enclose states that reappear as small framed worlds in the next stage of the computation.

stages of our interpretation. The state $\underline{rwb}\llbracket ?r_1\rrbracket$ is one of the worlds of the domain of $\underline{rwb}\llbracket L_{12}?r_1\rrbracket$ (as visualized in Figure 2 by linked frames) and is also the point of that state. The other world is $\underline{rbw}\llbracket ?r_1\rrbracket$. 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 $\underline{rwb}\llbracket ?r_1\rrbracket \sim'_1 \underline{rbw}\llbracket ?r_1\rrbracket$ in $\underline{rwb}\llbracket L_{12}?r_1\rrbracket$. For the same reason, both worlds are reflexive for both 1 and 2 in $\underline{rwb}\llbracket L_{12}?r_1\rrbracket$. Further note that $\underline{rwb}\llbracket ?r_1\rrbracket \not\sim'_2 \underline{rbw}\llbracket ?r_1\rrbracket$, because in $rwb \not\sim_2 rbw$ in $Hexa$. The valuation of atoms does not change. Therefore world $\underline{rwb}\llbracket ?r_1\rrbracket$ is named rwb, and world $\underline{rbw}\llbracket ?r_1\rrbracket$ is named rbw in Figure 2 that pictures the result.

Similarly to the computation of $\underline{rwb}\llbracket L_{12}?r_1\rrbracket$, compute the five other states where 1 and 2 learn 1's card. These form the domain of $\underline{rwb}\llbracket\mathsf{show}\rrbracket$. We compute access on the model in some typical cases. Again, reflexivity follows for all worlds: either because an agent occurs in that world and the first case applies, or because an agent doesn't occur in that world and the origins are identical, so obviously the same for that agent. We have that $\underline{rwb}\llbracket L_{12}?r_1\rrbracket \sim'_1 \underline{rbw}\llbracket L_{12}?r_1\rrbracket$ (as worlds), because $\underline{rwb}\llbracket L_{12}?r_1\rrbracket \sim_1 \underline{rbw}\llbracket L_{12}?r_1\rrbracket$ (as states), because, applying definition 4, the points $\underline{rwb}\llbracket ?r_1\rrbracket$ and $\underline{rbw}\llbracket ?r_1\rrbracket$ are the same for 1 in (the domain of the model underlying the) state $\underline{rwb}\llbracket L_{12}?r_1\rrbracket$. We also have that

$\underline{rwb}\llbracket L_{12}?r_1\rrbracket \sim'_3 \underline{wrb}\llbracket L_{12}?w_1\rrbracket$, because $3 \not\in \{1,2\}$ and $rwb \sim_3 wrb$ in $Hexa$. However, on the other hand $\underline{rwb}\llbracket L_{12}?r_1\rrbracket \not\sim'_2 \underline{bwr}\llbracket L_{12}?b_1\rrbracket$ (as worlds), because 2 occurs in both and $\underline{rwb}\llbracket L_{12}?r_1\rrbracket \not\sim_2 \underline{bwr}\llbracket L_{12}?b_1\rrbracket$ (as states), because $\underline{rwb}\llbracket L_{12}?r_1\rrbracket \ \underline{\not\leftrightarrow}\ \underline{bwr}\llbracket L_{12}?b_1\rrbracket$.

Again, the valuation of atoms in the worlds of $\underline{rwb}\llbracket\mathsf{show}\rrbracket$ does not change. Therefore world $\underline{rwb}\llbracket L_{12}?r_1\rrbracket$ is named rwb in Figure 2, etc. The point of $\underline{rwb}\llbracket\mathsf{show}\rrbracket$ is $\underline{rwb}\llbracket L_{12}?r_1\rrbracket$, because $\underline{rwb}\llbracket !L_{12}?r_1 \cup L_{12}?w_1 \cup L_{12}?b_1\rrbracket = \underline{rwb}\llbracket L_{12}?r_1\rrbracket$ (a more instructive point can be computed in whisper, next). We have now completed the interpretation. Figure 2 pictures the result.

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 10 (Local interpretation of table). *In state* $(Hexa, rwb)$*, player 1 puts the red card on the table:* $L_{123}?r_1$*.*

We do not show details of the computation. Figure 1 pictures the result. World rwb is actually state $\underline{rwb}\llbracket ?r_1\rrbracket$ and world rbw is actually state $\underline{rwb}\llbracket ?r_1\rrbracket$.

EXAMPLE 11 (Local interpretation of whisper). *In state* $(Hexa, rwb)$ *player 1 whispers in 2's ear 'I do not have the blue card':* $L_{123}(L_{12}?\neg r_1 \cup L_{12}?\neg w_1 \cup\ !L_{12}?\neg b_1)$*.*

We do not show details of the computation. Figure 1 pictures the result. Note that access is assumed to be transitive. Again, we have named the worlds by their atomic characterizations. We can distinguish worlds with the same name from each other, because they have different access to other worlds. Actually, e.g. the world rwb 'in front' is the state $\underline{rwb}\llbracket L_{12}?\neg w_1\rrbracket$ and the world rwb 'at the back' is the state $\underline{rwb}\llbracket L_{12}?\neg b_1\rrbracket$. The last is also the point. This can be observed by computing constraint $(M,w)\llbracket\alpha'\rrbracket w'$ in clause L_B in definition 5: $\underline{rwb}\llbracket L_{12}?\neg r_1 \cup L_{12}?\neg w_1 \cup\ !L_{12}?\neg b_1\rrbracket w' \Leftrightarrow \underline{rwb}\llbracket L_{12}?\neg b_1 \ ! \ (L_{12}?\neg r_1 \ ! \ L_{12}?\neg w_1)\rrbracket w' \Leftrightarrow \underline{rwb}\llbracket L_{12}?\neg b_1\rrbracket w'$ so $w' = \underline{rwb}\llbracket L_{12}?\neg b_1\rrbracket$.

In the 'back' rwb, that corresponds to the answer 'not blue', 2 knows that 1 holds red. In the 'front' rwb, that corresponds to the answer 'not white', 2 still considers bwr to be an alternative, so 2 does not know the card of 1. In both the 'back' and the 'front' rwb, neither 1 nor 3 know whether 2 knows 1's card!

We conclude with some other examples of actions in games.

EXAMPLE 12 (Win and pass). *Actions such as showing and telling other agents about your card(s), occur in card games where players* *also perform other actions. We call these games* knowledge games *(*van Ditmarsch, 2001a*). The goal of the game is to be the first to know (or guess rightly) the deal of cards, or some property derived from that. In Hexa, the condition of player 2 knowing the deal of cards can be described as* $\mathsf{win}_2 := K_2\delta_{rwb} \cup K_2\delta_{rbw} \cup \ldots$*. Here* δ_{ijk} *is the atomic description of world (deal)* ijk*, e.g.* $\delta_{rwb} := r_1 \wedge \neg r_2 \wedge \neg r_3 \wedge \neg w_1 \wedge w_2 \wedge \neg w_3 \wedge \neg b_1 \wedge \neg b_2 \wedge b_3$*. The action of player 2 winning is therefore described as the public announcement of that knowledge:* $L_{123}?\mathsf{win}_2$*.*

If the players are perfectly rational, ending one's move and passing to the next player also amounts to an action, namely announcing that you do not yet have enough knowledge to win. In the case of player 2: $L_{123}?\neg\mathsf{win}_2$*.*

EXAMPLE 13 (Cluedo). *The 'murder detection game' Cluedo is a concrete example of a knowledge game. The game consists of 21 cards and is played by six players. Each player has three cards and there are three cards on the table. The first player to guess those cards wins the game. The following actions are possible in Cluedo (and* only *those actions): showing (only to the requesting player) one of three requested cards (of different types, namely a murder suspect card, a weapon card, and a room card), confirming that you do not hold any of three requested cards (by public announcement), and 'ending your move', i.e. announcing that you cannot win. As each player has three cards, and there is no restriction on what cards are asked, a show action may involve actual choice, as in* whisper*. That 'ending your move' informs other perfectly rational players had previously not been noted.*

A play of the game Cluedo can therefore be described by a sequence of these different actions, so in a way by a single KA_A *action. See (*van Ditmarsch, 2000*) for details.*

Other standard applications of multiagent dynamics, such as the muddy children problem, also have simple descriptions in $\mathcal{L}_A$.

4. Theory

In this section we prove some properties of knowledge actions and their interpretation.

FACT 1 (Equivalence preservation). *The class of equivalence states is closed under execution of knowledge actions.*

This trivially follows from definition 5.

PROPOSITION 2 (Action algebra). *Let $\alpha, \alpha', \alpha^* \in \mathsf{KA}_A$. Then:*

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

Proof. By using simple relational algebra. We show (c), the rest is similar: $\llbracket (\alpha \cup \alpha') \ ; \ \alpha^* \rrbracket = \llbracket \alpha \cup \alpha' \rrbracket \circ \llbracket \alpha^* \rrbracket = (\llbracket \alpha \rrbracket \cup \llbracket \alpha' \rrbracket) \circ \llbracket \alpha^* \rrbracket = (\llbracket \alpha \rrbracket \circ \llbracket \alpha' \rrbracket) \cup (\llbracket \alpha \rrbracket \circ \llbracket \alpha^* \rrbracket) = \llbracket \alpha \ ; \ \alpha' \rrbracket \cup \llbracket \alpha \ ; \ \alpha^* \rrbracket = \llbracket (\alpha \ ; \ \alpha') \cup (\alpha \ ; \ \alpha^*) \rrbracket$.
■

We have not further investigated algebraic properties of action type operators. The next proposition relates concrete actions and action types to other actions.

PROPOSITION 3 (Concrete actions). *Let s, s' be equivalence models, let $\alpha \in \mathsf{KA}$. Then:*

(a) $\llbracket \alpha \rrbracket \subseteq \llbracket t(\alpha) \rrbracket$
(b) *concrete actions have a functional interpretation*
(c) $s \llbracket \alpha \rrbracket s' \Rightarrow \exists \beta \in C(\alpha) : s \llbracket \beta \rrbracket s'$
(d) $\llbracket \alpha \rrbracket = \llbracket \bigcup_{\beta \in C(\alpha)} \beta \rrbracket$

Proof.

(a) Induction on α. The only nontrivial case is $\alpha' \ ! \ \alpha''$. We have that: $\llbracket \alpha' \ ! \ \alpha'' \rrbracket = \llbracket \alpha' \rrbracket \subseteq \llbracket \alpha' \cup \alpha'' \rrbracket = \llbracket \alpha' \rrbracket \cup \llbracket \alpha'' \rrbracket \subseteq_{IH} \llbracket t(\alpha') \rrbracket \cup \llbracket t(\alpha'') \rrbracket = \llbracket t(\alpha') \cup t(\alpha'') \rrbracket = \llbracket t(\alpha' \ ! \ \alpha'') \rrbracket$.

(b) Induction on α. The only nontrivial case is nondeterministic choice. Let $\beta \in C(\alpha' \cup \alpha'')$. Then either $\beta = \beta' \ ! \ \beta''$ or $\beta = \beta'' \ ! \ \beta'$, with $\beta' \in C(\alpha')$ and $\beta'' \in C(\alpha'')$. In the first case, by induction $\llbracket \beta' \rrbracket$ is functional, and therefore also $\llbracket \beta' \ ! \ \beta'' \rrbracket = \llbracket \beta' \rrbracket$. In the second case, this follows from the functionality of $\llbracket \beta'' \rrbracket$.

(c) Induction on α. A typical case: If $s \llbracket \alpha' \cup \alpha'' \rrbracket s'$, then either $s \llbracket \alpha' \rrbracket s'$ or $s \llbracket \alpha'' \rrbracket s'$. If $s \llbracket \alpha' \rrbracket s'$ then, by induction, there is a $\beta' \in C(\alpha')$ such that $s \llbracket \beta' \rrbracket s'$. Let $\beta'' \in C(\alpha'')$ be arbitrary. Then $\beta' \ ! \ \beta'' \in C(\alpha' \cup \alpha'')$ and $s \llbracket \beta' \rrbracket s' = s \llbracket \beta' \ ! \ \beta'' \rrbracket s'$.

(d) Induction on α. Some cases. Case $\alpha' \ ; \ \alpha''$: use proposition 2.c and 2.d. Case $\alpha' \cup \alpha''$: $\llbracket \alpha' \cup \alpha'' \rrbracket =_{IH} \llbracket \bigcup_{\beta' \in C(\alpha')} \beta' \cup \bigcup_{\beta'' \in C(\alpha'')} \beta'' \rrbracket = \llbracket \bigcup_{\beta' \in C(\alpha'), \beta'' \in C(\alpha'')} (\beta' \ ! \ \beta'') \cup \bigcup_{\beta' \in C(\alpha'), \beta'' \in C(\alpha'')} (\beta'' \ ! \ \beta') \rrbracket = \llbracket \bigcup_{\beta \in C(\alpha)} \beta \rrbracket$. Case $L_B \alpha'$: use that $s \llbracket L_B \alpha' \rrbracket (M', w')$ implies $s \llbracket \alpha' \rrbracket w'$. ■

Proposition 3.a expresses that the interpretation of an action is contained in the interpretation of its type. Proposition 3.b expresses that concrete actions are state transformers. Proposition 3.c expresses that every state that results from action execution can be seen as the result of a concrete action. Proposition 3.d expresses that every action is equivalent to nondeterministic choice between all its concretizations (a kind of normal form, therefore). In particular, all clauses of proposition 3 hold when the arbitrary action is an action type or a concrete action. That suits the intuition even better: the interpretation of a concrete action is included in that of its type (a), an action type is equivalent to choice between all actions of that type (d), etc.

Preservation of bisimilarity We may expect that bisimilarity of states is preserved under execution of actions. This is indeed the case (theorem 5). However, to prove this we also need to show that bisimilar states have the same theory (theorem 4). This is not trivial, because modal formulas may contain dynamic modal operators for the effect of actions. We prove the theorems by simultaneous induction, assuming that φ is less complex than $?\varphi$ and that both α and φ are less complex than $[\alpha]\varphi$.

THEOREM 4 (Bisimilarity implies modal equivalence). *Let* $\varphi \in \mathcal{L}_A$. *Let* $(M, w), (M', w')$ *be equivalence states. If* $(M, w) \underline{\leftrightarrow} (M', w')$*, then* $M, w \models \varphi \Leftrightarrow M', w' \models \varphi$.

Proof. By induction of the structure of φ. The proof is standard except for the clause $\varphi = [\alpha]\psi$ that we therefore present in detail. Assume $M, w \models [\alpha]\psi$. We have to prove $M', w' \models [\alpha]\psi$. Let $(M^\bullet, w^\bullet)$ be arbitrary such that $(M', w')[\![\alpha]\!](M^\bullet, w^\bullet)$. By simultaneous induction hypothesis (theorem 5) it follows from $(M', w')[\![\alpha]\!](M^\bullet, w^\bullet)$ and $(M, w)\underline{\leftrightarrow}(M', w')$ that there is a (M^*, w^*) such that $(M, w)[\![\alpha]\!](M^*, w^*)$ and $(M^*, w^*)\underline{\leftrightarrow}(M^\bullet, w^\bullet)$. From $M, w \models [\alpha]\psi$ (given) and $(M, w)[\![\alpha]\!](M^*, w^*)$ follows $M^*, w^* \models \psi$. From $(M^*, w^*) \underline{\leftrightarrow} (M^\bullet, w^\bullet)$ and $M^*, w^* \models \psi$ follows, by induction, that $M^\bullet, w^\bullet \models \psi$. As $M^\bullet, w^\bullet$ was arbitrary, from $(M', w')[\![\alpha]\!](M^\bullet, w^\bullet)$ and $M^\bullet, w^\bullet \models \psi$ follows $M', w' \models [\alpha]\psi$.

THEOREM 5 (Action execution preserves bisimilarity). *Let* $\alpha \in \mathsf{KA}_A$. *Let* (M, w), (M', w') *be equivalence states. For every equivalence state* (M^*, w^*) *there is an equivalence state* $(M^\bullet, w^\bullet)$ *such that:*

If $(M, w)\underline{\leftrightarrow}(M', w')$ *and* $(M, w)[\![\alpha]\!](M^*, w^*)$*, then* $(M', w')[\![\alpha]\!](M^\bullet, w^\bullet)$ *and* $(M^*, w^*) \underline{\leftrightarrow} (M^\bullet, w^\bullet)$.

Proof. By induction on the structure of α. The proof consists of constructing a proper bisimulation $\mathfrak{R}^\alpha$ from a given bisimulation $\mathfrak{R}$, for each inductive case.

Case $?\varphi$: Suppose $\mathfrak{R} : (M, w) \underline{\leftrightarrow} (M', w')$ and $(M, w)[\![?\varphi]\!](M^*, w^*)$. Then $(M^*, w^*) = (M, w)[\![?\varphi]\!]$ and $w = w^*$. By simultaneous induction hypothesis (theorem 4) it follows from $(M, w)\underline{\leftrightarrow}(M', w')$ and $M, w \models \varphi$

that $M', w' \models \varphi$. Therefore $(M', w')\llbracket ?\varphi \rrbracket$ exists. For all worlds $v^* \in M^*$ and $v^\bullet \in (M', w')\llbracket ?\varphi \rrbracket$, define $\mathfrak{R}^{?\varphi}(v^*, v^\bullet) :\Leftrightarrow \mathfrak{R}(v^*, v^\bullet)$. Then $\mathfrak{R}^{?\varphi} : (M^*, w^*) \underline{\leftrightarrow} (M', w')\llbracket ?\varphi \rrbracket$, because (Points:) $\mathfrak{R}^{?\varphi}(w, w')$, (Back and Forth:) both states have empty access, and (Valuation:) $\mathfrak{R}^{?\varphi}(v^*, v^\bullet)$ implies $\mathfrak{R}(v^*, v^\bullet)$ implies, for all $p \in P$: $v^* \in V(p) \Leftrightarrow v^\bullet \in V(p)$.

Case $L_B\alpha'$: Suppose $\mathfrak{R} : (M, w) \underline{\leftrightarrow} (M', w')$ and $(M, w)\llbracket L_B\alpha' \rrbracket(M^*, w^*)$. Note that $(M', w')\llbracket L_B\alpha' \rrbracket$ *exists*, as its domain is not empty: by induction its point is a state bisimilar to w^*. We claim that $(M', w')\llbracket L_B\alpha' \rrbracket$ is the $(M^\bullet, w^\bullet)$ that we are looking for, and we define a $\mathfrak{R}^{L_B\alpha'}$ to establish the the required bisimulation.

The relation $\mathfrak{R}^{L_B\alpha'}$ between (M^*, w^*) and $(M', w')\llbracket L_B\alpha' \rrbracket$ is defined as follows: Let $w_1^* \in (M^*, w^*)$ and $w_1^\bullet \in (M', w')\llbracket L_B\alpha' \rrbracket$. According to the construction of $L_B\alpha'$, there is a $v_1 \in M$ such that $(M, v_1)\llbracket t(\alpha') \rrbracket w_1^*$ and $v_1 \sim_B w$, and there is a $v_1' \in M'$ such that $(M', v_1')\llbracket t(\alpha') \rrbracket w_1^\bullet$ and $v_1' \sim_B w'$. If $\mathfrak{R}(v_1, v_1')$, then by induction there is a $\mathfrak{R}^{t(\alpha')}$ such that $\mathfrak{R}^{t(\alpha')} : w_1^* \underline{\leftrightarrow} w_1^\bullet$ relates the points of w_1^* and $w_1^\bullet$. Define $\mathfrak{R}^{L_B\alpha'}(w_1^*, w_1^\bullet) :\Leftrightarrow \mathfrak{R}^{t(\alpha')} : w_1^* \underline{\leftrightarrow} w_1^\bullet$. It is important to observe that the definition is well-defined: because $w \sim_B v_1$ in M, world v_1 will have an $\mathfrak{R}$-image in M', and vice versa.

We now proceed to prove that $\mathfrak{R}^{L_B\alpha'} : (M^*, w^*) \underline{\leftrightarrow} (M', w')\llbracket L_B\alpha' \rrbracket$.

(Points:) $\mathfrak{R}^{L_B\alpha'}(w^*, w^\bullet)$, because $\mathfrak{R}^{t(\alpha')}(w^*, w^\bullet)$, because $\mathfrak{R}(w, w')$ (given).

(Forth:) Let $a \in B$, $w_2^* \in M^*$, $w_1^* \sim_a' w_2^*$, and $\mathfrak{R}^{L_B\alpha'}(w_1^*, w_1^\bullet)$. Assume $(M, v_2)\llbracket t(\alpha') \rrbracket w_2^*$. We distinguish case $a \in gr(w_1^*)$ (and, because of $w_1^* \sim_a' w_2^*$, therefore also $a \in gr(w_2^*)$) from case $a \notin gr(w_1^*) \cup gr(w_2^*)$. In the first case we use that $\mathfrak{R}^{t(\alpha')}$ is a bisimulation to establish the required world $w_2^\bullet$, in the second case we use that $\mathfrak{R}$ is a bisimulation and that $\mathfrak{R}^{t(\alpha')}$ preserves bisimilarity, to establish that.

If $a \in gr(w_1^*)$, then from $w_1^* \sim_a' w_2^*$ follows $w_1^* \sim_a w_2^*$ (i.e., as *states*), so the points of these states are the same for a as well. From $\mathfrak{R}^{L_B\alpha'}(w_1^*, w_1^\bullet)$ follows $\mathfrak{R}^{t(\alpha')} : w_1^* \underline{\leftrightarrow} w_1^\bullet$, therefore $\mathfrak{R}^{t(\alpha')}$ relates the points of w_1^* and $w_1^\bullet$. From that and from the fact that the points of w_1^* and w_2^* are the same for a, and because $\mathfrak{R}^{t(\alpha')}$ is a bisimulation, follows that there is a $w_2^\bullet$ such that the point of $w_1^\bullet$ is the same for a as the point of $w_2^\bullet$ and $\mathfrak{R}^{t(\alpha')} : w_2^* \underline{\leftrightarrow} w_2^\bullet$. But we now also have $\mathfrak{R}^{L_B\alpha'}(w_2^*, w_2^\bullet)$ and $w_1^\bullet \sim_a' w_2^\bullet$ (as worlds)!

If $a \notin gr(w_1^*) \cup gr(w_2^*)$, then $v_1 \sim_a v_2$ in M, by the definition of access in (M^*, w^*). From $v_1 \sim_a v_2$ and $\mathfrak{R}(v_1, v_1')$, and because $\mathfrak{R}$ is a bisimulation, follows that there is a $v_2' \in M'$ such that $v_1' \sim_a v_2'$ in M' and $\mathfrak{R}(v_2, v_2')$. By induction we may assume that $\mathfrak{R}^{t(\alpha')}$ preserves bisimilarity, therefore there is a $w_2^\bullet$ such that $(M', v_2')\llbracket t(\alpha') \rrbracket w_2^\bullet$ and

$\mathfrak{R}^{t(\alpha')} : w_2^* \underline{\leftrightarrow} w_2^\bullet$. We now have that $w_1^\bullet \sim_a' w_2^\bullet$ (from $v_1' \sim_a v_2'$) and $\mathfrak{R}^{L_B \alpha'}(w_2^*, w_2^\bullet)$ (from $\mathfrak{R}^{t(\alpha')} : w_2^* \underline{\leftrightarrow} w_2^\bullet$). Done.

(Back:) Similar to forth.

(Valuation:) Obvious.

Case $\alpha' ; \alpha''$: Suppose $\mathfrak{R} : (M, w) \underline{\leftrightarrow} (M', w')$ and $(M, w)[\![\alpha' ; \alpha'']\!](M^*, w^*)$. As $[\![\alpha' ; \alpha'']\!] = [\![\alpha']\!] \circ [\![\alpha'']\!]$, there is an (M_1, w_1) such that $(M, w)[\![\alpha']\!](M_1, w_1)$ and $(M_1, w_1)[\![\alpha'']\!](M^*, w^*)$. By induction we have an (M_1', w_1') such that $(M', w')[\![\alpha']\!](M_1', w_1')$ and $\mathfrak{R}^{\alpha'} : (M_1, w_1) \underline{\leftrightarrow} (M_1', w_1')$. Again by induction we have an $(M^\bullet, w^\bullet)$ and an $\mathfrak{R}^{\alpha', \alpha''}$ such that $(M_1', w_1')[\![\alpha'']\!](M^\bullet, w^\bullet)$ and $\mathfrak{R}^{\alpha', \alpha''} : (M^\bullet, w^\bullet) \underline{\leftrightarrow} (M^*, w^*)$. $\mathfrak{R}^{\alpha', \alpha''}$ is the required bisimulation $\mathfrak{R}^{\alpha' ; \alpha''}$, as we also have $(M', w')[\![\alpha' ; \alpha'']\!](M^\bullet, w^\bullet)$.

Case $\alpha' \cup \alpha''$: Suppose $\mathfrak{R} : (M, w) \underline{\leftrightarrow} (M', w')$ and $(M, w)[\![\alpha' \cup \alpha'']\!](M^*, w^*)$. Then either $(M, w)[\![\alpha']\!](M^*, w^*)$ or $(M, w)[\![\alpha'']\!](M^*, w^*)$. If $(M, w)[\![\alpha']\!](M^*, w^*)$, then by induction there is an $(M^\bullet, w^\bullet)$ and a $\mathfrak{R}^{\alpha'}$ such that $(M', w')[\![\alpha']\!](M^\bullet, w^\bullet)$ and $\mathfrak{R}^{\alpha'} : (M^*, w^*) \underline{\leftrightarrow} (M^\bullet, w^\bullet)$. From $(M', w')[\![\alpha']\!](M^\bullet, w^\bullet)$ follows $(M', w')[\![\alpha' \cup \alpha'']\!](M^\bullet, w^\bullet)$, so $\mathfrak{R}^{\alpha'}$ is the required bisimulation. Similarly, if $(M, w)[\![\alpha'']\!](M^*, w^*)$.

Case $\alpha' \; ! \; \alpha''$. Similar to $\alpha' \cup \alpha''$. ■

A direct consequence of theorem 5 is:

COROLLARY 6. *Let s, s' be equivalence states and α a concrete action. Then $s \underline{\leftrightarrow} s' \Rightarrow s[\![\alpha]\!] \underline{\leftrightarrow} s'[\![\alpha]\!]$.*

5. Further observations

The interpretation of some $\mathcal{L}_A$ formulas is undefined. An obvious example is the following: The formula $[?K_1 r_1]K_1 r_1$ – ‘after a test on 1 knowing red, 1 knows red’ – can *not* be interpreted on any state. Suppose $\underline{rwb} \models [?K_1 r_1]K_1 r_1$, then $\underline{rwb}[\![?K_1 r_1]\!] \models K_1 r_1$. However, $gr(\underline{rwb}[\![?K_1 r_1]\!]) = \emptyset$: it is an *equivalence* state with empty access, on which $K_1 r_1$ can therefore not be interpreted. For similar reasons, formulas as $[L_{12}?r_1 ; L_{123}?r_1]r_1$ are uninterpretable. Expanding the notion gr of ‘group’ to include actions may provide a solution. We then put a constraint $\varphi \in \mathcal{L}_{gr(\alpha)}$ on the formation of $[\alpha]\varphi$ in definition 2, so that formulas such as $[?K_1 r_1]K_1 r_1$ are no longer well-formed. With this syntax restriction we can derive validities as $[?\varphi]\psi \leftrightarrow (\varphi \rightarrow \psi)$. We have not yet completed the axiomatization of $\mathcal{L}_A$. It appears to become the axiomatization of a 'family' of logics $\mathcal{L}_B$ for all $B \subseteq A$ (given a global set of atoms P).

We have extended the language with the operation of *concurrent execution* (van Ditmarsch, 2001b). This relates to (Peleg, 1987; Goldblatt, 1992). Using concurrency, we can describe that a player shows two cards simultaneously to different players, say, one with his left hand and the other one with his right hand. The notion of local interpretation is 'lifted' from a relation between states to a relation between states and *sets* of states.

The action language would be further enhanced if we could refer not just to the current game state, but also the action *history*. We could then describe, e.g., that a player asks another player to show him 'another' card.

We make some closing remarks on the relation of our work to that of others.

A public update with formula φ in (Plaza, 1989) naturally corresponds to learning: $(\varphi + \psi)$ is equivalent to $[L_A?\varphi]\psi$ (where A is the public).

Learning is rather similar to updating in (Gerbrandy, 1999). The semantics of actions is also relational. However, in Gerbrandy outsiders to a group learning ('updating') something are assumed to learn nothing at all. We do not make that assumption. Because outsiders learn nothing, they cannot imagine the actual state of the world: no reflexivity. Indeed, his approach is more general than for equivalence states only. In (Gerbrandy, 1999) the crucial operator is the (sub)group update U_B. Actions $L_A\alpha$ (for the public A, i.e. fully specified actions) correspond to 'truthful updates' $(\alpha \; ; \; U_B\alpha)$ à la Gerbrandy.

Apart from interpreting an action as a relation between states, an action can also be interpreted as a semantic object corresponding to a Kripke frame, an 'action frame'. Executing an action in a state then amounts to computing a direct product of that state and that frame. This is the approach in (Baltag, 1999; Baltag et al., 2000). See also (van Ditmarsch, 2000; van Ditmarsch, 2001a). The different notions of interpretation correspond up to bisimilarity (van Ditmarsch, 2000). It is also interesting to observe that a concrete action corresponds to a pointed action frame (a state transformer), whereas an action type corresponds to an 'ordinary' action frame (i.e. with no point).

6. Conclusion

We proposed a dynamic epistemic language $\mathcal{L}_A$, that includes a language KA_A 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 taken from card games, to illustrate the language and its interpretation. We compared our research to that of others.

Acknowledgements

I thank the anonymous referees for their detailed reports. I thank Computer Science, University of Otago, for allowing me ample time to complete the revision.

References

Baltag, A.: 1999, 'A Logic of Epistemic Actions'. Manuscript.

Baltag, A., L. Moss, and S. Solecki: 2000, 'The Logic of Public Announcements, Common Knowledge and Private Suspicions'. Revised manuscript, originally presented at TARK 98, submitted for publication.

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

Fagin, R., J. Halpern, Y. Moses, and M. Vardi: 1995, *Reasoning about Knowledge*. Cambridge MA: MIT Press.

Gerbrandy, J.: 1999, 'Bisimulations on Planet Kripke'. Ph.D. thesis, University of Amsterdam. ILLC Dissertation Series DS-1999-01.

Gerbrandy, J. and W. Groeneveld: 1997, 'Reasoning about Information Change'. *Journal of Logic, Language, and Information* **6**, 147–169.

Goldblatt, R.: 1992, *Logics of Time and Computation*. Stanford: CSLI Publications, 2 edition. CSLI Lecture Notes No. 7.

Harel, D.: 1984, 'Dynamic Logic'. In: D. Gabbay and F. Guenthner (eds.): *Handbook of Philosophical Logic*, Vol. II. Dordrecht, pp. 497–604, Kluwer Academic Publishers.

Harel, D., D. Kozen, and J. Tiuryn: 2000, *Dynamic Logic* Foundations of Computing Series. Cambridge MA: MIT Press.

Meyer, J.-J. and W. van der Hoek: 1995, *Epistemic Logic for AI and Computer Science*. Cambridge Tracts in Theoretical Computer Science 41. Cambridge: Cambridge University Press.

Parikh, R.: 1987, 'Knowledge and the problem of logical omniscience'. In: R. Zas and M. Zemankova (eds.): *Proceedings of the 2nd international symposium on methodologies for intelligent systems*. Amsterdam, pp. 432–439, North Holland.

Peleg, D.: 1987, 'Concurrent Dynamic Logic'. *Journal of the ACM* **34(2)**, 450–479.

Plaza, J.: 1989, 'Logics of Public Communications'. In: M. Emrich, M. Pfeifer, M. Hadzikadic, and Z. Ras (eds.): *Proceedings of the 4th International Symposium on Methodologies for Intelligent Systems*. pp. 201–216.

van Ditmarsch, H.: 1999, 'The logic of knowledge games: showing a card'. In: E. Postma and M. Gyssens (eds.): *Proceedings of the BNAIC 99*. Maastricht University, pp. 35–42.

van Ditmarsch, H.: 2000, 'Knowledge games'. Ph.D. thesis, University of Groningen. ILLC Dissertation Series DS-2000-06.

van Ditmarsch, H.: 2001a, 'Knowledge games'. *Bulletin of Economic Research* **53(4)**, 249–273.

van Ditmarsch, H.: 2001b, 'The semantics of concurrent knowledge actions'. In: M. Pauly and G. Sandu (eds.): *ESSLLI 2001 workshop on Logic and Games*.