

Dynamic Epistemic Logic with Assignment

H.P. van Ditmarsch
Computer Science
University of Otago
New Zealand

hans@cs.otago.ac.nz

W. van der Hoek
Computer Science
University of Liverpool
United Kingdom

wiebe@csc.liv.ac.uk

B.P. Kooi
Philosophy
University of Groningen
the Netherlands

barteld@philos.rug.nl

ABSTRACT

We add assignment operators to languages for epistemic actions, so that change of knowledge and change of facts can be combined in specifications of multi-agent system dynamics. We make a distinction between ‘public assignment’ and ‘atomic assignment’. Public assignment means that the entire group of agents is aware of the factual change. This operation combines well with public announcement. We propose semantics for the logic of public announcements and public assignments, and we give a relevant valid principle. Atomic assignment means that *only* facts are changed, so that it can be expressed independently in the language *how* agents are aware of this factual change. Atomic assignment and ‘test’ (the truth of a formula) are the two basic constructs in a logic of epistemic actions. We propose semantics for this logic of epistemic actions and atomic assignment. The logic of public announcements and public assignments is a special case of this logic. Examples include card games where cards change hands.

Categories and Subject Descriptors

I.2.11 [Distributed Artificial Intelligence]: Multiagent systems

General Terms

Languages, Theory

Keywords

agent communication, assignment, dynamic epistemic logic

1. INTRODUCTION

This contribution combines dynamic epistemic logic, i.e., the logic of reasoning about change of knowledge, with a standard framework for assignment.

Epistemic logic, or the logic of knowledge, started with [14], and has been investigated since in philosophy [16], computer science [8], artificial intelligence [17], and game theory

[2]. The latter three application areas made it apparent that in multi-agent systems *higher-order information*, knowledge about other agents’ knowledge, is crucial.

Change of knowledge, and of belief, has been put on the agenda by Alchourrón et al. [1], and has since developed strongly. Standard issues include modelling differences between belief revision and belief updates [15], and the problem of iterated belief change [7]. In such approaches the dynamics is studied on a level above the informational level, and not incorporated in the logical language. This made it hard, if not impossible, to model the knowledge an agent has about other agents’ knowledge.

In dynamic epistemics one can describe knowledge change in the logical language, one can also describe knowledge an agent has about other agents’ knowledge, and one can describe change of knowledge, all in one. Publications in this area include [20, 9, 24, 22, 3, 25]. The basic idea behind knowledge change in this field is that it can be modelled as an *action* or program that induces an *information state transformation*. In other words, it can be modelled as an operation that, given a relational structure that represents what agents know, and what they know about each other, transforms that structure into a ‘new information state’, i.e., a similar relational structure, that represents the new state of information.

Assignment is a primitive operation in programming languages. Given program variables of a certain type, for example natural numbers or booleans, an assignment has the form $p := e$, for ‘variable p becomes expression e ’. This means that the value of p becomes (is assigned) the value of e , presuming that e can be evaluated in that programming language. See [10]. Highlights are the Hoare calculus of correctness of programs [6], in which assignment is the primitive program command, and the dynamic logic of Harel and collaborators [11], wherein programs such as assignments are the parameters of dynamic modal operators that express their pre- and postconditions.

In an epistemic setting the analogue of a ‘set of state variable values’, is a ‘valuation of atomic propositions’, i.e., a ‘state of the world’ or factual description. Now if it were *only* that, we could proceed as in dynamic logic. But instead of programs inducing (factual) state transitions, in dynamic epistemic logic we have programs inducing epistemic or information state transitions. An epistemic state is, roughly, nothing but a multi-set of factual state descriptions, with relations between them that describe how agents can distinguish between those states. The assignments now take the form $p := \varphi$, for ‘(in every state of the domain) the value of

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

AAMAS’05, July 25-29, 2005, Utrecht, Netherlands.
Copyright 2005 ACM 1-59593-094-9/05/0007 ...\$5.00.

atomic proposition p becomes the value of epistemic formula φ (in that state). The epistemic formula φ is interpreted in an epistemic state. The epistemic state transitions induced by assignments are similar to those induced by epistemic actions.

There are many multi-agent system scenarios where change of knowledge, i.e., actions or observations, and change of facts, interact and cannot be separated. Distributed systems, of which the ‘toy’ Muddy Children Problem is a very simple form, and card games where cards change hands, are introduced in the examples below. Detailed specifications of such systems are relevant for other aspects of their description as well, and possibly for other disciplines, such as the computation of optimal strategies for such card games [19]. Other fields of application are the modelling of speech acts and presuppositions in dialogue (www.fipa.com), and modelling negotiations and contractual agreement, in particular when that involves complex communicative acts [21].

We present two examples where assignment and knowledge interact: in the setting of the ‘Muddy Children Problem’ [18], we consider the action of *washing a child*; and in the setting of players holding cards [24], we consider the action of *swapping cards*.

Suppose that there are three children, Anne, Bill, and Cath, two of which have mud on their foreheads, suppose Anne and Bill. It is commonly known that children can only see the forehead of other children. Father comes forward and announces “At least one of you is muddy”. Father happens to be carrying a bucket of water and now empties it over Anne. (This is not part of the standard problem.) Anne is now clean, and this is publicly known. Father then says: “Let those who know whether they are muddy please step forward”. Instead of no-one stepping forward, which is the case in the standard version of the problem, Anne now steps forward, because she knows that she is not muddy. Father asks again: “Let those who know whether they are muddy please step forward”. What happens? And what if instead of just emptying the bucket over Anne, he had said at that same time “Anne, you dirty child, let’s do something about this”. What would have happened in this case after father’s second request? And what if Bill and Cath had been muddy instead of Anne and Bill, at the beginning?

Note that not just facts may change, such as the value of ‘Anne is muddy’, but also knowledge about facts, such as the value of ‘Bill knows that Anne is muddy’, but even knowledge about others’ knowledge, such as the value of ‘Anne knows that Bill knows whether she is muddy’.

We continue with the card game example. There are three players Anne, Bill, and Cath, and three cards Wheat, Flax, and Rye. The cards are shuffled and every player draws a card from the stack. Suppose Anne draws Wheat, Bill draws Flax, and Cath draws Rye. Players only know their own card. Anne and Bill now swap their cards. What do the players know about each other in the resulting game state? What if Anne had two cards instead of one? What is the difference, in the last case, between letting Bill *draw* a card from her hand, and Anne *handing* Bill a card? And in the first case?

Again, note that not just facts may change, such as the value of ‘Anne holds Wheat’, but also knowledge, such as the value of ‘Anne and Bill know the card deal’, and even knowledge about the knowledge of other players, such as the value of ‘Cath knows that Anne and Bill know the card

deal’.

An assignment operation can be added in a natural way to *both* the logic of public announcements *and* the ‘full’ dynamic epistemic logic. The first we call public assignment and the second atomic assignment. The first is easier to explain informally. It is an atomic program $p := \varphi$ that changes the value of atom p into that of the formula φ – and assumes that all the agents are aware of this change. In the case of father cleaning Anne in the first example, the assignment executed is $m_a := \perp$. Atom m_a stands for ‘Anne is muddy’, and \perp for the false proposition, so that $m_a := \perp$ stands for ‘all agents learn that it becomes false that Anne is muddy’. We can use this assignment in dynamic modal operators that express the result of epistemic state transformation, e.g., in the formula $[m_a := \perp]K_a \neg m_a$. This stands for ‘after all agents learn that it becomes false that Anne is muddy, Anne knows that she is not muddy’. This does not assume that Anne was muddy before the assignment.

Section 2 presents the logic of public announcements and public assignments. Section 3 presents the logic of epistemic actions and atomic assignment. We give no proof systems of the logics presented. We focus on examples and applications in multi-agent systems of actions involving both change of knowledge and factual change. Section 4 takes a cursory glance at concurrent actions, multiple assignments, and change of belief.

2. PUBLIC ASSIGNMENT

The logic of public announcements and public assignments contains both epistemic and dynamic modal operators. With epistemic operators we express individual knowledge, for an arbitrary agent, and public (common) knowledge, for the entire group of agents. With dynamic modal operators we express the effect of public announcements, i.e., public (and truthful) announcements of formulas in the language, and the effect of public assignments. The parameters that play a static role throughout the semantic and syntactic definitions are a set of agents N and a set of propositional atoms P .

An *epistemic model* $M = \langle S, \sim, V \rangle$ consists of a *domain* S of *factual states* or just states, *accessibility* $\sim: N \rightarrow \mathcal{P}(S \times S)$ which for each agent $n \in N$ defines a binary accessibility relation (that is an equivalence relation) \sim_n on S , and a *valuation* $V: P \rightarrow \mathcal{P}(S)$ which for each atom $p \in P$ defines a valuation $V_p \subseteq S$. If M is an epistemic model, and $s \in \mathcal{D}(M)$ (s is in the domain of M), then the pointed model (M, s) is an *epistemic state*.

Definition 1 (Logical language) The formulas are inductively defined as

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

We assume familiarity with the propositional connectives and with notational abbreviations. Formula $K_n\varphi$ stands for ‘agent n knows φ ’, and $C_N\varphi$ stands for ‘it is common knowledge in group N that φ ’. The construct $[\varphi]\psi$ stands for ‘after truthful public announcement of φ , it holds that ψ ’. Instead of ‘ φ is a public and truthful announcement’ we say ‘ φ is an announcement’. The expression $[p := \varphi]\psi$ stands for ‘after (all agents learn that) p becomes φ , it holds that ψ ’.

Definition 2 (Semantics) Given an epistemic model $M = \langle S, \sim, V \rangle$ and a state $s \in S$. We define $(M, s) \models \varphi$ by induction on φ . Write $\llbracket \varphi \rrbracket_M$ for $\{s \in S \mid (M, s) \models \varphi\}$.

$M, s \models p$	iff	$s \in V_p$
$M, s \models \neg \varphi$	iff	$M, s \not\models \varphi$
$M, s \models \varphi \wedge \psi$	iff	$M, s \models \varphi$ and $M, s \models \psi$
$M, s \models K_n \varphi$	iff	for all $t : s \sim_n t$ implies $M, t \models \varphi$
$M, s \models C_N \varphi$	iff	for all $t : s \sim_N t$ implies $M, t \models \varphi$
$M, s \models [\varphi] \psi$	iff	$M, s \models \varphi$ implies $M \varphi, s \models \psi$
$M, s \models [p := \varphi] \psi$	iff	$M_{p:=\varphi}, s \models \psi$

where $\sim_N := \bigcup_{n \in N} \sim_n^*$ (i.e., the reflexive transitive closure of the union of all \sim_n), where $M|\varphi := \langle S', \sim', V' \rangle$ is defined as

$$\begin{aligned} S' &= \llbracket \varphi \rrbracket_M \\ \sim'_n &= \sim_n \cap (\llbracket \varphi \rrbracket_M \times \llbracket \varphi \rrbracket_M) \\ V'_p &= V_p \cap \llbracket \varphi \rrbracket_M \end{aligned}$$

and where $M_{p:=\varphi} = \langle S, \sim, V' \rangle$ is as M except that $V'_p = \llbracket \varphi \rrbracket_M$ (i.e., for all $q \neq p$, we have that $V'_q = V_q$). \dashv

In other words: the model $M|\varphi$ is the model M restricted to all the states where φ holds, including access between states, and the valuation of states. The model $M_{p:=\varphi}$ is the model M but wherein the value of the atom p takes the extension of the formula φ in the model M (and where all else remains the same). Unlike public announcements, that can only be executed when true, the public assignment action is always executable. We can therefore see it as an operation on the model underlying the epistemic state. We write $M_{p:=\varphi}$ for the model resulting from that assignment. Validity, and validity in models, are defined as usual.

2.1 Example involving public assignment

Public assignment is an action that changes facts that describe states in the model, in a way that is public to all agents. Consider the introductory example of father emptying a bucket of water over Anne, in view of all children, after his announcement that at least one of them is muddy, but before his request to step forward if you know whether you are muddy. Atom m_a describes that Anne is muddy, etc. The top left model (*Cube*, 110) in Figure 1 represents the epistemic state where Anne and Bill are muddy, before Father's announcements. The name 101 stands for the state where Anne and Cath are muddy, and Bill is not muddy, etc. The part $011-a-111$ means that Anne cannot distinguish between the states 011 and 111 (because she cannot see her own forehead), etc. The actual state 110 is underlined: Actually, Anne and Bill are muddy.

As the result from father's announcement that at least one child is muddy, state 000 (where no-one is muddy) is eliminated from public consideration. As the result of executing assignment $m_a := \perp$ in that epistemic state only the valuation of fact m_a is changed, namely into 'false': all 1s in the first position of all states now have become 0s. Note that the state 010 where Anne was formerly clean is epistemically different from the state 010 where Anne was formerly muddy: in the first case Bill knows that he is muddy, but in the second case he does not know that. After father's request, only Anne steps forward. Not surprisingly: it is now common knowledge that Anne is clean. Therefore she knows *whether* she is muddy: namely 'not'. We have

$$\begin{aligned} \text{Cube}, 110 &\models [m_a \vee m_b \vee m_c] \neg (K_a m_a \vee K_a \neg m_a) \\ \text{Cube}, 110 &\models [m_a \vee m_b \vee m_c][m_a := \perp] K_a \neg m_a \end{aligned}$$

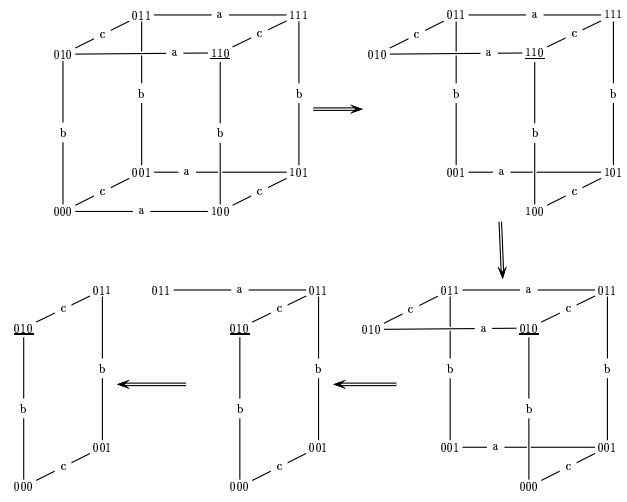


Figure 1: A version of the muddy children problem that includes cleaning them. States that cannot be distinguished by agent n have an n -link between them. The actual states are underlined. Clockwise, from the top left corner: the initial configuration; after father says that at least one is muddy; after father hosed Anne with water; after those who know whether they are muddy step forward (and Anne steps forward); after that request is repeated (eliminating that only Bill and Cath were muddy).

Because Bill and Cath do not step forward at the same time as Anne, the leftmost state 010 and 001 in the bottom-right model are eliminated. Unlike in the 'traditional' analysis of the Muddy Children Problem, state 000 (formerly 100) can *not* be eliminated from the model, as the reason for Anne to step forward is unknown to Bill and Cath: she may have stepped forward because she already knew that she was muddy – for the original case 100 – or she may have stepped forward because she has just been cleaned. After father repeats his request, the left-most 011 in the bottom-middle model of the figure is eliminated. This corresponds to the situation where instead of Anne and Bill, Bill and Cath had been muddy at the outset. No further repetitions of Father's request will change that epistemic state: it *remains* unclear to Bill and Cath whether they are muddy. Without the 'cleaning', Bill would have learnt that he is muddy after the first request, and step forward (with Anne) at its repetition.

What 'sort' of actions that regularly recur in multi-agent systems can be made by combining public announcements and public assignments? We defer a discussion of that to Section 3.2, where they can be contrasted with other epistemic actions. More complex actions often involve subgroup interaction that cannot be described with public announcements and assignments only, such as the action where father throws a bucket with *dirty* water over Anne, so that she may or may not become clean.

2.2 Reduction axiom for assignment

What is the principle relating truth of formulas before and after the assignment? Let $\psi(p/\varphi)$ stand for the substitution of φ for all occurrences of p in ψ . The Hoare-calculus for

program correctness contains a rule for assignment

$$\{\psi(p/\varphi)\} \quad p := \varphi \quad \{\psi\}$$

to which corresponds a dynamic principle

$$\psi(p/\varphi) \rightarrow [p := \varphi]\psi$$

which *suggests* the reduction axiom

$$\psi(p/\varphi) \leftrightarrow [p := \varphi]\psi$$

But this principle is unsound (invalid), because of the interaction of assignment and announcement, and the peculiar phenomenon that some formulas expressing ignorance become false *because* they are announced. (In other words, unlike ‘standard’ program semantics, the evaluation of an expression depends on *more* than the set of state values, but just like the standard setting, the result is after all *only* the reset of a state value. This discrepancy causes the problem.)

For example, imagine that I were to say “You do not know that I am going to the movies tonight”. The superficial logical structure of that announcement is $\neg Kp$, where p describes the atomic proposition ‘I am going to the movies tonight’ and where K is the epistemic operator describing your knowledge. A presupposition in such a dialogue is also that p is true, therefore the actual announcement is $p \wedge \neg Kp$. As a result of this announcement, you *know* that I am going to the movies tonight: Kp . In other words, the formula of the announcement became false, *because* it was announced.

We can capture this scenario in an epistemic state $(M, 1)$ such that $M = \langle \{0, 1\}, \sim, V \rangle$ with $V_p = \{1\}$ and $\sim = \{0, 1\} \times \{0, 1\}$. The instance

$$p \wedge \neg Kp \leftrightarrow [p := p \wedge \neg Kp][p]p$$

of the supposed reduction axiom (wherein we have applied the substitution) is *false* in $(M, 1)$, so that it must be invalid. One can easily see that

$$(M, 1) \not\models p \wedge \neg Kp$$

but that

$$(M, 1) \models [p := p \wedge \neg Kp][p]p$$

The result of the announcement $p \wedge \neg Kp$ is the single-state model consisting of state 1 only, in which the announced formula is false; whereas the result of the assignment $p := p \wedge \neg Kp$ is $(M, 1)$ itself again, in which after announcement of p , p is still true.

The ‘principle of public assignment’ $\psi(p/\varphi) \leftrightarrow [p := \varphi]\psi$ is valid if φ is a *preserved* formula. The preserved formulas are those preserved under restriction to arbitrary submodels, i.e., if $M' \subseteq M$, then for all $s \in \mathcal{D}(M)$: if $M, s \models \varphi$ then $M', s \models \varphi$. This entails that for arbitrary formulas χ that are true in s , if $M, s \models \varphi$ then $M|\chi, s \models \varphi$.

We consider completeness and proof theory – although fascinating and important topics – outside the scope of this contribution, and merely report on progress in this matter. The principle of public assignment appears not useful for a proof system for this logic, because a syntactic characterization of the rather semantic notion of ‘preservation’ is not available. Instead, one adds reduction axioms by case distinction on the structure of ψ in $[p := \varphi]\psi$.

The logic of public announcements *without* assignment is proved sound and complete in [3], with precursors in [20, 9]. The logic of multi-agent epistemic logic (i.e., without announcements) *with* public assignment can easily be seen

to be complete, with or without common knowledge. Reduction axioms are sufficient in that case. The logic *without* common knowledge, *with* public announcements, and *with* public assignments, is also complete. We then use that there is a ‘rewrite system’ from the logic of public announcements without common knowledge to that logic without announcements. But the completeness of the logic presented in this paper is still a conjecture: the interaction between common knowledge, announcement, and assignment, creates one difficulty too much to be easily resolved.

3. ATOMIC ASSIGNMENT

We now introduce a more expressive logic, of which the logic of public announcements and public assignments is a special case. To a standard multiagent epistemic language with common knowledge, for a set N of agents and a set P of atoms [17, 8], we add dynamic modal operators for epistemic actions. Actions may change the knowledge of the agents involved, and, if the actions involve assignments, they may also change the value of facts. The structures on which this logic is interpreted, are the same as for the logic of public announcements and public assignments. We need one more concept: given an epistemic model $M = \langle S, \sim, V \rangle$, the set of agents N is called the group (*gr*) of this model, for which we write $gr(M) = N$.

Definition 3 (Logical language) The *formulas* are defined by

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

The *actions* are defined by

$$\alpha ::= ?\varphi \mid p := \varphi \mid L_G\alpha \mid (\alpha ! \beta) \mid (\alpha \mid \beta) \mid (\alpha ; \beta) \mid (\alpha \cup \beta)$$

The formulas and actions in bold carry extra constraints, that have been deleted in this presentation. In $C_G\varphi$ and in $L_G\alpha$, group G is an arbitrary subset of the set N of all agents. Remove brackets whenever convenient. The construct $[\alpha]\psi$ stands for ‘after every execution of action α , it holds that ψ ’. The dual of $[\alpha]$ is $\langle \alpha \rangle$, so that $\langle \alpha \rangle\psi$ stands for ‘after *some* execution of α , it holds that ψ ’.

Action $?\varphi$ is a *test*. Action $p := \varphi$ is an *atomic assignment*. We write this the same as public assignment, as there is no source of ambiguity in action expressions. Action $L_G\alpha$ is pronounced as ‘group G learn α ’ and this is called a *learning* action. Action $(\alpha ; \beta)$ is *sequential execution*, and $(\alpha \cup \beta)$ is *nondeterministic choice*. Action $(\alpha ! \beta)$ is called (left) *local choice*, and $(\alpha \mid \beta)$ is its dual, right local choice. Local choice $(\alpha ! \beta)$ may be seen as ‘from α and β , choose the first *locally*’ and local choice $(\alpha \mid \beta)$ as ‘from α and β , choose the second *locally*’. In an action of the form $L_G(\alpha ! \beta)$, everybody in G and not in α or β (i.e., not in learning operators occurring in α or β) is unaware of the choice for α . Therefore we call the choice ‘local’: only agents in α or β may be aware of it. Instead of $(\alpha ! \beta)$ we generally write $(! \alpha \cup \beta)$. This expresses more clearly that given choice between α and β , the agents involved in those actions choose α , whereas that choice remains invisible to the agents that learn about these alternatives but are not involved. Similarly, instead of $(\alpha \mid \beta)$ we generally write $(\alpha \cup ! \beta)$. If we replace all ‘!’ and ‘|’ operators in an action α by \cup operators, we get the *type* $t(\alpha)$ of that action. The use of ‘!’ and ‘|’ operators merely allows us to constrain the interpretation of a possibly non-deterministic action to a deterministic one.

To interpret actions and formulas we need a notion of equivalence between epistemic states. We lift equivalence of factual states in the domain of an epistemic state to equivalence of epistemic states, in the obvious way:

$$\begin{aligned} (M, s) \sim_n (M, t) & \text{ iff } s \sim_n t \\ (M, s) \sim_n (M', s') & \text{ iff there is a } t \in M \text{ such that } (M, t) \\ & \quad \Leftrightarrow (M', s') \text{ and } (M, s) \sim_n (M, t) \end{aligned}$$

The symbol \Leftrightarrow stands for ‘is bisimilar to’, see [17, 5]. Epistemic states occur as factual states in clause $L_G\alpha$ of the following definition of the interpretation of formulas and actions. Equivalence of such epistemic states determines equivalence of the corresponding factual states.

Definition 4 (Semantics) Given a model $M = \langle S, \sim, V \rangle$ and an $s \in M$. The interpretation \models of a formula φ in (M, s) and the interpretation $\llbracket \cdot \rrbracket$ of an action α between epistemic states are defined by simultaneous induction.

$$\begin{aligned} (M, s) \models p & \text{ iff } s \in V_p \\ (M, s) \models \neg\varphi & \text{ iff } (M, s) \not\models \varphi \\ (M, s) \models \varphi \wedge \psi & \text{ iff } (M, s) \models \varphi \text{ and } (M, s) \models \psi \\ (M, s) \models K_n\varphi & \text{ iff for all } t : s \sim_n t \text{ implies } (M, t) \models \varphi \\ (M, s) \models C_G\varphi & \text{ iff for all } t : s \sim_G t \text{ implies } (M, t) \models \varphi \\ (M, s) \models [\alpha]\varphi & \text{ iff for all } (M', s') : (M, s)[\alpha](M', s') \\ & \text{ implies } (M', s') \models \varphi \end{aligned}$$

$$\begin{aligned} (M, s)[?\varphi](M', s') & \text{ iff } M' = \langle \llbracket \varphi \rrbracket_M, \emptyset, V \cap \llbracket \varphi \rrbracket_M \rangle \\ & \text{ and } s' = s \\ (M, s)[p := \varphi](M', s') & \text{ iff } M' = \langle S, \emptyset, V' \rangle \text{ and } s' = s \\ (M, s)[L_G\alpha](M', s') & \text{ iff } M' = \langle S', \sim', V' \rangle \text{ and} \\ & \quad (M, s)[\alpha]s' \\ \llbracket \alpha ; \beta \rrbracket & = \llbracket \alpha \rrbracket \circ \llbracket \beta \rrbracket \\ \llbracket \alpha \cup \beta \rrbracket & = \llbracket \alpha \rrbracket \cup \llbracket \beta \rrbracket \\ \llbracket \alpha ! \beta \rrbracket & = \llbracket \alpha \rrbracket \end{aligned}$$

In the clause for ‘assignment’: V' is as V except that $V_p = \llbracket \varphi \rrbracket_M$. In the clause for ‘learning’, M' is such that: $S' = \{(M'', t'') \mid \exists u \in M : (M, u)[t(\alpha)](M'', t'')\}$; for an arbitrary agent n : if $(M, s)[t(\alpha)](M'_1, s'_1)$ and $(M, t)[t(\alpha)](M'_2, t'_2)$, then $(M'_1, s'_1) \sim_n (M'_2, t'_2)$ iff $(M'_1, s'_1) \sim_n (M'_2, t'_2)$ (where \sim_n means equivalence of epistemic states) or $(n \notin gr(M'_1) \cup gr(M'_2))$ and $s \sim_n t$; and for an arbitrary atom p and factual state (M'', s'') (with valuation V'') in the domain of M' : $(M'', s'') \in V'_p$ iff $s'' \in V_p$. \dashv

A test results in an epistemic state without access for any agent. This is appropriate: how knowledge changes is only expressed in ‘learning’, so before we encounter an L operator we cannot say anything at all about the knowledge of the agents in the epistemic state resulting from action execution. One might as well say that, while compositionally interpreting an action, the computation of agents’ knowledge is *deferred* until L operators are encountered. Note that this is different from public announcement, where all access is retained on the restriction of the domain to the φ -states.

Assignments always succeed and change the value of atoms. Just as for tests, and for the same reason, it results in an epistemic state without access. Note that this is different from public assignment, where all access is retained.

Learning $L_G\alpha$ is defined in terms of $t(\alpha)$. This is how local choice constructions $\alpha ! \beta$ get their ‘real’ meaning,

even though $\llbracket \alpha ! \beta \rrbracket = \llbracket \alpha \rrbracket$ suggests that the ‘!’ connective is superfluous. Note that $\llbracket L_G(\alpha ! \beta) \rrbracket$ is computed from $\llbracket t(\alpha ! \beta) \rrbracket = \llbracket t(\alpha) \cup t(\beta) \rrbracket = \llbracket t(\alpha) \rrbracket \cup \llbracket t(\beta) \rrbracket$, and therefore from *both* $\llbracket \alpha \rrbracket$ and $\llbracket \beta \rrbracket$. To execute an action $L_G\alpha$ in an epistemic state (M, s) , we do not just have to execute the *actual* action α in the *actual* epistemic state (M, s) , but also any *other* action of the same type $t(\alpha)$ as α in any *other* epistemic state (M, t) with the same underlying model M . The results are epistemic states that serve as factual states in the domain of the epistemic state that results from executing $L_G\beta$ in (M, s) . Such factual states cannot be distinguished from each other by an agent $n \in G$, if they are either indistinguishable as epistemic states, or if the agent does not occur in these epistemic states but could not distinguish between the epistemic states before execution of the actions.

Bisimilarity of epistemic states is preserved under execution of epistemic actions. A proof system for a version of this logic *without* assignments and with concurrency is proposed in [25]. The logic of epistemic actions *with* assignments is under investigation and a proof system is yet to be provided. We also mention that *public assignment* $p := \varphi$ is equivalent to the epistemic action $L_N(p := \varphi)$ (where $p := \varphi$ is now *atomic assignment*), and that *public announcement* of φ is equivalent to the epistemic action $L_N?\varphi$.

3.1 Example involving atomic assignment

For an example, we model that Anne and Bill swap their cards, in a situation where Anne, Bill, and Cath (a, b, c) each hold one of the cards Wheat, Flax, and Rye (w, x, y), respectively, and where players only know their own card. Atoms q_n describe that the card named q is held by player n , e.g., w_a describes that Anne holds Wheat. This situation is modelled by the epistemic state $(Hexa, w.x.y)$, see Figure 2.

The action where Anne swaps a card q for a card q' of Bill, in such a way that Cath cannot see which cards are swapped, is described as follows.

$$\begin{aligned} Swap(a, b)(q, q') & = ?(q_a \wedge q'_b); q_a := \perp; q_b := \top; \\ & \quad q'_b := \perp; q'_a := \top \\ LearnSwap(a, b)(w, x) & = L_{abc}(\ !L_{ab}Swap(a, b)(w, x) \cup \\ & \quad L_{ab}Swap(a, b)(w, y) \cup \\ & \quad L_{ab}Swap(a, b)(x, w) \cup \\ & \quad L_{ab}Swap(a, b)(x, y) \cup \\ & \quad L_{ab}Swap(a, b)(y, w) \cup \\ & \quad L_{ab}Swap(a, b)(y, x)) \end{aligned}$$

We read this action as follows: Anne, Bill and Cath learn (L_{abc}) that one from six alternatives is executed: either Anne and Bill learn (L_{ab}) that ‘they swap one card namely w and x ’ ($Swap(a, b)(w, x)$), or ..., or ...; and the first of these six alternatives really happens (but *only* Anne and Bill learn that). The $Swap$ action consists of five parts, take $Swap(a, b)(w, x)$ for example. We then get $?(w_a \wedge x_b)$; $w_a := \perp$; $w_b := \top$; $x_b := \perp$; $x_a := \top$. The first part tests whether Anne holds Wheat and Bill holds Flax. If so, then four assignments are executed. That is the actual swapping of those two cards. The order of the four assignments is irrelevant. In this case, it reads as: make it false that Anne holds Wheat, then make it true that Bill holds Wheat. (That completes transferring the ownership of Anne’s Wheat card.) Then, make it false that Bill holds Flax, and after that make it true that Anne holds Flax. The four assignments *must* be kept together, so that the cards are traded

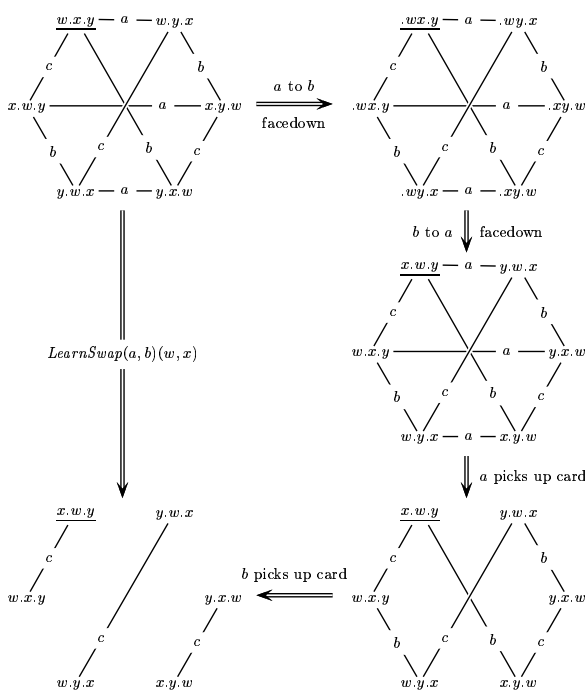


Figure 2: Top-left: the epistemic state ($Hexa, w.x.y$) where three players Anne, Bill and Cath (a, b, c) each hold *one* card namely Wheat, Flax, and Rye (w, x, y), respectively. The actual deal $w.x.y$ is underlined. Clockwise, from top-left: Anne puts her Wheat card facedown on the table in front of Bill; Bill puts the card that he knows, i.e., his Flax card, facedown on the table in front of Anne. Then, Anne looks at her card. Then, Bill looks at his card. We can also model this sequence as one action $LearnSwap(a, b)(w, x)$.

simultaneously and not one after the other: if a player gives a card to another player and *then* that other player returns a card, *that may have been the card that the other player had just been given*. The *Swap* parts *must* be bound by L_{ab} : from the point of view of Anne and Bill, the action is public (to them), only from Cath's point of view it isn't.

The result of the action $LearnSwap(a, b)(w, x)$ is pictured in the transition between the top-left and the bottom-left model in Figure 2 (unrelated to the other transitions in the figure). We refrain from giving details of the computation of the interpretation of this action and only mention that, e.g., the factual state $x.w.y$ in the resulting model is actually the epistemic state ($Hexa, w.x.y$) $[L_{ab}Swap(a, b)(w, x)]$ (we use a postfix functional notation for the unique epistemic state resulting from executing that action in that state). As agent c does not occur in this state, nor in factual state $w.x.y$ of the resulting model that really is epistemic state ($Hexa, x.w.y$) $[L_{ab}Swap(a, b)(x, w)]$, these two epistemic states are the same for Cath *as factual states* in the resulting model, because before execution of these actions, $w.x.y$ was the same for Cath as $x.w.y$ (because she holds Rye in both deals). Etc.

We can also model the action of trading cards as a sequence of four different actions. This is also pictured in Figure 2. For the sake of generality, we do this from a perspective where players might hold *more than one* card from

a set of cards Q . First, Anne puts her card facedown in front of Bill (and for a general perspective, we let her choose the cards that she gives to Bill, and not blindly draw one of her cards); then Bill puts *a card that he knows to hold* (in that way excluding the card just handed to him by Anne, that he also 'holds' but that he does not know yet) facedown in front of Anne; then Anne picks up her new card (this is the card 'that she does not know yet'); then Bill picks up his new card. The types of these actions are

$$\begin{aligned}
 &L_{abc} \bigcup_{q \in Q} L_a(?q_a ; q_a := \perp ; q_b := \top) \\
 &L_{abc} \bigcup_{q \in Q} L_b(?K_b q_b ; q_b := \perp ; q_a := \top) \\
 &L_{abc} \bigcup_{q \in Q} L_a?(q_a \wedge \neg K_a q_a) \\
 &L_{abc} \bigcup_{q \in Q} L_b?(q_b \wedge \neg K_b q_b)
 \end{aligned}$$

If Anne *blindly* chooses one of her cards to give to Bill the description is

$$L_{abc} \bigcup_{q \in Q} (?q_a ; q_a := \perp ; q_b := \top)$$

The difference between the previous description is the absence of the L_a constructor. Without it, Anne is not aware of the card she is handing to Bill. In other words, this has become the action where Bill *draws* one of Anne's cards.

3.2 Ontology of epistemic assignment actions

If we abstract from the formulas occurring in actions, and therefore regard their communicative structure only, most epistemic assignment actions in multi-agent systems fall in one of a small number of categories. A tentative overview is the following. We only give action *types*, and for simplicity assignments are to false or true only. They are on occasion followed by examples. The first two action types have equivalent descriptions in the logic of public announcements and public assignments.

1. A fact ('de re') becomes false:
 $\bigcup_{p \in P} L_N(p := \perp)$
Father throws a bucket of water over Anne:
 $L_{abc}(m_a := \perp)$
2. A true fact ('de re') becomes false:
 $\bigcup_{p \in P} L_N(?p ; p := \perp)$
3. A fact ('de dicto') becomes false:
 $L_N \bigcup_{p \in P}(p := \perp)$
4. A true fact ('de dicto') becomes false:
 $L_N \bigcup_{p \in P}(?p ; p := \perp)$
Father says "I just saw (all) mud fall from one of your foreheads":
 $L_{abc} \bigcup_{n=a,b,c}(?m_n ; m_n := \perp)$
5. A subgroup of agents observes an action:
 $L_N \bigcup_{\alpha} L_{G\alpha}$
Anne and Bill swap a card:
 $L_{abc} \bigcup_{q, q'} L_{ab}Swap(a, b)(q, q')$
Father throws a bucket of dirty water over Anne:
 $L_{abc}(L_{bc}(m_a := \perp) \cup L_{bc}(m_a := \top))$

The difference between 'de re' and 'de dicto' in items 1. and 3. can be paraphrased as follows: in the 'de re' case, there is a fact, such that all agents learn that that fact becomes false. In the 'de dicto' case, all agents learn that there is a fact that becomes false (but they might not know which fact). Similarly for the difference between items 2. and 4.

We close this section with a two slightly more detailed examples.

Two agents Alice (a) and Bob (b) fix and communicate a private key q from a set of possible keys P in the presence of an eavesdropper Eve (e):

$$L_{abe}(!L_{ab}(q := \top) \cup \bigcup_{p \neq q \in P} L_{ab}(p := \top))$$

Anne and Bill reach contractual agreement by intermediation of a solicitor in the following way: Anne (a) signs the contract (p) in the presence of witnesses a_1 and a_2 and solicitor c . Then Bill (b) signs the contract in the presence of witnesses b_1 and b_2 and the solicitor:

$$\begin{aligned} &L_{aa_1a_2bb_1b_2c} ? \bigwedge_{p \in P} \neg p ; \\ &L_{aa_1a_2bb_1b_2c} \bigcup_{p \in P} L_{aa_1a_2c}(p := \top) ; \\ &L_{aa_1a_2bb_1b_2c} \bigcup_{p \in P} L_{bb_1b_2c} ? p \end{aligned}$$

After the action, it is public knowledge (i.e., legally established) that Anne and Bill share common knowledge of the content of the contract ('the value of p ').

$$C_{abq} \wedge C_{aa_1a_2bb_1b_2c} (\bigvee_{p \in P} C_{abp})$$

In other words, the contract has become in force. This has been achieved, even though Anne, but not Bill, actively participated in the second of the three actions (a occurs in $L_{aa_1a_2c}$), and Bill, but not Anne, actively participated in the third of the three actions (b occurs in $L_{bb_1b_2c}$).

4. CONCLUSIONS, FURTHER RESEARCH

Multiple assignment A language with multiple (concurrent) assignment – optionally including precondition(s) – is an alternative to the action primitives 'test' and 'atomic assignment' in the above. But this amounts to the same thing. A precondition for an assignment is nothing but a test preceding it. In other words, a construct $(\varphi : p := \psi)$ is equivalent to a sequence $?\varphi ; p := \psi$. Multiple assignments can be made sequential by interleaving and adding a new atomic variable r to the language. For example, the multiple assignment $p, q := q, p$ where the values of two atoms are swapped, can be simulated the sequence of three actions $r := p ; p := q ; q := r$, after introducing a new atom r in the language. A description of $Swap(a, b)(w, x)$ with multiple assignment is

$$?(w_a \wedge x_b) ; (w_a, w_b, x_a, x_b := w_b, w_a, x_b, x_a)$$

Concurrent actions There is no relation between concurrent epistemic actions [25] and 'concurrent' assignments. The first is 'true' concurrency, i.e., the execution of an action in an epistemic state can result in a set of concurrent epistemic states. But the second is, as already mentioned, 'intersection' or interleaving concurrency. For the difference between the two types of concurrency, see also [12]. An elegant description in concurrent dynamic epistemic logic of (the type of the action of) swapping cards, before picking up those cards, is

$$L_{abc} \left(\bigcup_{q \in Q} L_a(?q_a ; q_a := \perp ; q_b := \top) \cap \bigcup_{q \in Q} L_b(?q_b ; q_b := \perp ; q_a := \top) \right)$$

Belief We have only modelled the interaction of changing knowledge and changing facts. Belief is weaker than knowledge: one may believe something that is not true. To model private actions, we also need to incorporate belief. For example, Anne and Bill swap Wheat and Flax without Cath

noticing it, so that Cath incorrectly *believes* that Anne (still) holds Wheat. There are obvious generalization of the logics presented here in that direction.

Other approaches Various work investigates the combination of knowledge, change of knowledge, and change of facts. We mention results by Jérôme Lang and others [13], by Alexandru Baltag and others [4], and by Jan van Eijck and others [26, 23].

Conclusions We gave a language and a semantics for an epistemic logic with public announcements and public assignments. We also gave a language and a semantics for an epistemic logic with epistemic actions and atomic assignments, of which the first logic is a special case. We gave an overview of types of epistemic action involving assignment including examples in distributive systems and games.

5. ACKNOWLEDGMENTS

We thank the anonymous AAMAS referees for their comments.

6. REFERENCES

- [1] C. Alchourrón, P. Gärdenfors, and D. Makinson. On the logic of theory change: partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50:510–530, 1985.
- [2] R. Aumann and A. Brandenburger. Epistemic conditions for nash equilibrium. *Econometrica*, 63:1161–1180, 1995.
- [3] A. Baltag and L. Moss. Logics for epistemic programs. *Synthese*, 139:165–224, 2004. Knowledge, Rationality & Action 1–60.
- [4] A. Baltag, L. Moss, and S. Solecki. The logic of public announcements, common knowledge, and private suspicions. Technical report, Centrum voor Wiskunde en Informatica, Amsterdam, 1999. CWI Report SEN-R9922.
- [5] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, Cambridge, 2001. Cambridge Tracts in Theoretical Computer Science 53.
- [6] O. Dahl, E. Dijkstra, and C. Hoare. *Structured Programming*. Academic Press, London, 1972.
- [7] A. Darwiche and J. Pearl. On the logic of iterated belief revision. *Artificial Intelligence*, 89(1-2):1–29, 1997.
- [8] R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning about Knowledge*. MIT Press, Cambridge MA, 1995.
- [9] J. Gerbrandy. *Bisimulations on Planet Kripke*. PhD thesis, University of Amsterdam, 1999. ILLC Dissertation Series DS-1999-01.
- [10] D. Gries. *The Science of Programming*. Springer, Berlin, 1981.
- [11] D. Harel. Dynamic logic. In D. Gabbay and F. Guenther, editors, *Handbook of Philosophical Logic*, volume II, pages 497–604. Dordrecht, 1984. Kluwer Academic Publishers.
- [12] D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic*. MIT Press, Cambridge MA, 2000. Foundations of Computing Series.

- [13] A. Herzig, J. Lang, and P. Marquis. Revision and update in multiagent belief structures. Manuscript, 2005.
- [14] J. Hintikka. *Knowledge and Belief*. Cornell University Press, Ithaca, NY, 1962.
- [15] H. Katsuno and A. Mendelzon. On the difference between updating a knowledge base and revising it. In *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning*, pages 387–394, 1991.
- [16] W. Lenzen. Recent work in epistemic logic. *Acta Philosophica Fennica*, 30:1–219, 1978.
- [17] J.-J. 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.
- [18] Y. O. Moses, D. Dolev, and J. Y. Halpern. Cheating husbands and other stories: a case study in knowledge, action, and communication. *Distributed computing*, 1(3):167–176, 1986.
- [19] M. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, Cambridge MA, 1994.
- [20] J. Plaza. Logics of public communications. In M. Emrich, M. Pfeifer, M. Hadzikadic, and Z. Ras, editors, *Proceedings of the 4th International Symposium on Methodologies for Intelligent Systems*, pages 201–216, 1989.
- [21] M. Schoop. A language-action approach to electronic negotiations. In H. Weigand, G. Goldkuhl, and A. de Moor, editors, *Proceedings of the 8th International Working Conference on the Language-Action Perspective on Communication Modelling (LAP 2003)*, pages 143–160, Tilburg, NL, 2003.
- [22] J. van Benthem. Logics for information update. In J. van Benthem, editor, *Proceedings of TARK VIII*, pages 51–88, Los Altos, 2001. Morgan Kaufmann.
- [23] J. van Benthem, J. van Eijck, and B. Kooi. Logics of communication and change. manuscript, 2005.
- [24] H. van Ditmarsch. Descriptions of game actions. *Journal of Logic, Language and Information*, 11:349–365, 2002.
- [25] H. van Ditmarsch, W. van der Hoek, and B. Kooi. Concurrent dynamic epistemic logic. In V. Hendricks, K. Jørgensen, and S. Pedersen, editors, *Knowledge Contributors*, pages 45–82, Dordrecht, 2003. Kluwer Academic Publishers. Synthese Library Volume 322.
- [26] J. van Eijck. Guarded actions. Technical report, Centrum voor Wiskunde en Informatica, Amsterdam, 2004. CWI Report SEN-E0425.