ARJEN HOMMERSOM, JOHN-JULES MEYER and ERIK DE VINK

UPDATE SEMANTICS OF SECURITY PROTOCOLS

ABSTRACT. We present a model-theoretic approach for reasoning about security protocols, applying recent insights from dynamic epistemic logics. This enables us to describe exactly the subsequent epistemic states of the agents participating in the protocol, using Kripke models and transitions between these based on updates of the agents' beliefs associated with steps in the protocol. As a case study we will consider the SRA Three Pass protocol and discuss the Wide-Mouthed Frog protocol.

1. INTRODUCTION

In today's world of e-commerce and the Internet, the role of security protocols is getting increasingly important. The design of these security protocols is difficult and error-prone (Lowe 1996; Schneier 2000; Anderson 2001), which makes (automatic) verification of protocols of crucial importance. Since the late 1980s, one line of research, amongst others, for reasoning about security protocols is based on the use of the so-called BAN logic, proposed by Burrows, Abadi and Needham in (Burrows et al., 1990). This is an epistemic logic augmented by constructs that are relevant for reasoning about security, such as the property of having the disposal of a cryptographic key to be able to decode a message and therefore to know its contents. Although many useful results have been reported (e.g., Kessler and Neumann 1998; Agray et al., 2001; Stubblebine 2002), due to their complexity and their semantic underpinning the use of BAN logics to prove the correctness of security protocols has so far been of limited success (cf. Abadi and Tuttle 1991; Wedel and Kessler 1996; Bleeker and Meertens 1997).

In this paper we will apply insights from dynamic epistemic logics as recently developed by Gerbrandy (1997, 1999), Baltag and Moss (Baltag et al., 1998; Baltag 2002; Baltag and Moss 2004), van Ditmarsch (2000, 2001), and Kooi (2003). Moreover, contrary to the traditional BAN logic approach, our approach is semantic or model-theoretic. We use Kripke models to represent the epistemic state of

Synthese **142:** 229–267, 2004.
Knowledge, Rationality & Action 289–327.

the agents involved in a protocol, similarly to the S5 preserving approach of Van Ditmarsch to analyze certain kinds of games involving knowledge. From the action models of Baltag and Moss we import the idea to describe belief updates of the agents by semantic operators transforming the Kripke models at hand by copying and deleting parts of these models, although we use traditional Kripke models rather than action models. To this end, we need also operations for unfolding models, which is in its turn inspired by Gerbrandy's work on possibilities. The difference being that in our approach only *partial* unfolding is called for. We furthermore propose a language to express belief updates in the context of security protocols as well as properties of these updates, and give a semantics of this language in terms of the models mentioned and the operators on them. Since our approach is model-theoretic, we believe that it may serve as a starting point for the automatic verification of (properties of) security protocols.

As a case study illustrating our approach we will consider the so-called SRA Three Pass protocol. It is not our intention to prove that the protocol is completely secure (as it is not in full generality), but we will prove that if the agents participating in the protocol are honest, then an intruder watching the communication does not learn anything about the plain-text messages in a single run. Furthermore, we show what the intruder is able to learn about the agents participating. We also discuss the Wide-Mouthed Frog protocol to illustrate the operation developed in the sequel for updating the beliefs of agents.

2. PRELIMINARIES

In this section we briefly discuss some preliminaries and background regarding the semantic updates we will handle and the epistemic models we will use. First, we define the notion of an objective formula and introduce so-called o-seriality. The set of propositional variables in a model is denoted as $\mathscr{P}$.

DEFINITION 2.1. The class of objective formulas is the smallest class such that

- all propositional variables and atoms $p \in \mathscr{P}$ are objective;
- if ϕ is objective, then $\neg\phi$ is objective;
- if ϕ_1 and ϕ_2 are objective, then $\phi_1 \wedge \phi_2$ is objective.

So, objective formulas do not involve beliefs. For our purposes it is important that every agent, at every world, distinguishes a world with the same 'objective' information. This leads to the notion of an o-serial model. The operations on Kripke structures discussed in the sequel degenerate for models that are not o-serial.

DEFINITION 2.2. A model $M = \langle S, \pi, R_1, \ldots, R_m \rangle$ is *o-serial* iff for all agents i and $w \in S$, there exists $v \in S$ such that $R_i(w, v)$ and for all objective formulas ϕ it holds that $(M, w) \models \phi \Leftrightarrow (M, v) \models \phi$.

We use a, b, c, etc. and i, j as typical agents, taken from a class $\mathscr{A}$. Furthermore, B is used as a doxastic modal operator. For example, $B_a\phi$ should be read as 'a believes ϕ'. We interpret formulas on standard Kripke models $(M, s) = (\langle S, \pi, R_1, \ldots, R_m \rangle, s)$, where $(M, s) \models B_i\phi$ iff $\forall t \in S$: $R_i(s, t) \rightarrow (M, t) \models \phi$.

We require the relations R_i to be o-serial, transitive and euclidean. This yields a class of models that we will call Kt45, a proper subset of the class of models of the well-known doxastic logic *KD45*. The lower case t refers to the axiom

$$B_i\phi \Rightarrow \phi \tag{t}$$

where the formula ϕ ranges over objective formulas. The system Kt45 is sound with respect to the class of o-serial, transitive and euclidean models (Hommersom 2003). (We conjecture that Kt45 is complete as well for this class.) We will show that the operations we introduce preserve Kt45. The point is that in worlds of Kt45 models, we cannot both have $B_i\phi$ and $\neg\phi$, for an objective formula ϕ. This is reasonable from the assumption that agents are conscious about the protocol. Therefore, they will not infer objective falsehoods. This objectivity is captured locally for each state. As a consequence, the operations that we introduce can restrict the set of states without destroying objective information.

For the analysis of security protocols below, we assume that we are omniscient about the values of the variables in different runs of a protocol. For example, the program variable p in a protocol run has the value $[\![p]\!]$. In the real world it is, obviously, always true that $p = [\![p]\!]$. However, it is cumbersome to keep track of what is the real world in the operations on Kripke structures that we employ below. Therefore, we assume that an interpretation $[\![\cdot]\!]$ is given, that provides the 'real' values (not necessarily boolean) of the program variables when needed. It might very well be the case that $p \neq [\![p]\!]$ in a certain state. Often, we will abbreviate $p = [\![p]\!]$ to p on (thus transforming a program expression into a propositional variable). Similarly, $\neg p$ is an abbreviation of

$p \neq [\![p]\!]$. For example, agent a that learns $B_b p \vee B_b \neg p$, learns that agent b has assigned a value to the program variable p.

The types of updates we consider are

- public announcement of a variable,
- private learning of a variable, and
- private learning about the knowledge of other agents.

The first type of update typically runs as follows: In an open network, agent a sends a message to agent b. From a security perspective, it is customary in the so-called Dolev–Yao framework (Dolev and Yao 1983), to assume that all agents in the network can read this message too. In contrast, the second type of update, describes private learning. For example, agent b receives a message $\{x\}_k$ from agent a. (Here, we use the notation $\{x\}_k$ to denote a message x encrypted with the cryptographic key k.) If b possesses the key k, then b privately learns the message content x. The final type of update is probably the most interesting. It is realistic to assume that the steps in a protocol run are known to all agents. Therefore, observing that an agent receives a message will increase the knowledge of the other agents. For example, if agent a sends a message $\{x\}_k$ to agent b, then agent c learns that b has learned the information contained in the message $\{x\}_k$, but typically, c does not learn x if c does not possess the key k.

Stronger types of updates we do not consider here. For example, we will not update the beliefs of an honest agent such that it learns that an intruder has learned about others. In the present paper, we restrict ourselves to beliefs about objective formulas and the updating of such beliefs.

3. UPDATE CONSTRUCTIONS

In this section we describe various types of updates in detail. We will start by defining an update for propositions in Section 3.1. In Section 3.2 we will define a belief update for agents that learn something about the belief of others. We do this in two slightly different ways by varying in the operations that describe a side-effect for an agent.

3.1. *Objective Updates*

The belief update of objective formulas we will use is based on the work reported in (Baltag et al., 1998; Roorda et al., 2002). The construction works as follows: We will make copies of the states of the model such that the *old* worlds correspond to the information in the original model and the *new* worlds correspond to the new information.

DEFINITION 3.1. Let a model $(M, w) = (\langle S, \pi, R_1, \dots, R_m \rangle, w)$, a group of agents $\mathcal{B}$, and an objective formula ϕ be given. Then $\text{UPDATE}_{(\phi,\mathcal{B})}(M, w)$, the update of (M, w) for agents in $\mathcal{B}$ and formula ϕ, is given by $\text{UPDATE}_{(\phi,\mathcal{B})}(M, w) = (\langle S', \pi', R'_1, \dots, R'_m \rangle, w')$, where

- $S' = \{old(s), new(s) \mid s \in S\}$
- $w' = new(w)$
- for all $p \in \mathcal{P} : \pi'(old(u))(p) = \pi'(new(u))(p) = \pi(u)(p)$
- for $a \in \mathcal{A}$, the binary relation R'_a on S' is minimal such that

$$\begin{array}{lcll} R'_a(old(u), old(v)) & \Leftrightarrow & R_a(u, v) & \\ R'_a(new(u), new(v)) & \Leftrightarrow & R_a(u, v) \wedge (M, v) \models \phi & \text{if } a \in \mathcal{B} \\ R'_a(new(u), old(v)) & \Leftrightarrow & R_a(u, v) & \text{if } a \notin \mathcal{B} \end{array}$$

In order to distinguish the two copies of the states, the tagging function *old* and *new* are used. In the new part of the model, agents in $\mathcal{B}$ will only consider possible worlds that verify ϕ. Therefore, states can become unreachable from the actual world $new(w)$, and can be dropped.

The following example shows how this works on a concrete model.

EXAMPLE 3.1 (updating). Consider the model (M, s) in Figure 1, where we have $\pi(s)(p) = \texttt{true}$ and $\pi(t)(p) = \texttt{false}$. The operation we execute is $\text{UPDATE}_{(p,\{b\})}$, i.e. b learns p. This results in the model (M, u) in Figure 2, where $\pi(u)(p) = \pi(v)(p) = \texttt{true}$ and $\pi(w)(p) = \texttt{false}$ and $new(s) = u$, $old(s) = v$ and $old(t) = w$. The world $new(t)$ is unreachable and is omitted.

We can see that the belief of agent a has not changed: it still considers its old worlds possible. The belief of agent b, however, has changed. It now only considers the state u possible where p holds. Note that agent b is aware that agent a does not know about p.

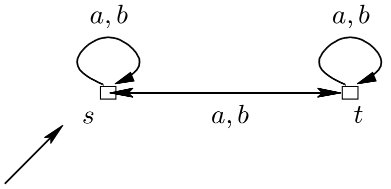

Figure 1. (M, s).

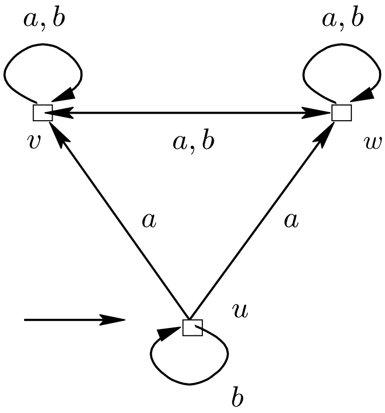

Figure 2. (M, u).

The update operation $\textsc{update}_{(\phi,\mathscr{B})}$ is based on a formula ϕ and a set of agents $\mathscr{B}$. Roorda et al. (2002) propose a characterization of the formulas that are altered by such an operation with a single learning agent. Here, we extend their definition for multi-agent purposes.

DEFINITION 3.2. An update function $(\cdot)[\phi, \mathscr{B}]$ is called proper if

$$\begin{array}{lcl}
(M,w)[\phi,\mathscr{B}] \models p & \Leftrightarrow & (M,w) \models p \\
(M,w)[\phi,\mathscr{B}] \models \alpha \wedge \beta & \Leftrightarrow & (M,w)[\phi,\mathscr{B}] \models \alpha \text{ and } (M,w)[\phi,\mathscr{B}] \models \beta \\
(M,w)[\phi,\mathscr{B}] \models \neg\alpha & \Leftrightarrow & (M,w)[\phi,\mathscr{B}] \not\models \alpha \\
(M,w)[\phi,\mathscr{B}] \models B_a\alpha & \Leftrightarrow & (M,w) \models B_a\alpha \text{ if } a \notin \mathscr{B} \\
(M,w)[\phi,\mathscr{B}] \models B_a\alpha & \Leftrightarrow & \forall v : ((R_a(w,v) \text{ and } (M,v) \models \phi) \Rightarrow \\
 & & \qquad (M,v)[\phi,\mathscr{B}] \models \alpha) \text{ if } a \in \mathscr{B}
\end{array}$$

for every model M and state w.

Following Roorda et al. (2002) we have that $\textsc{update}_{(\phi,\mathscr{B})}$ is proper (cf. Roorda et al., 2002, Proposition 3.2). Moreover, $\textsc{update}_{(\phi,\mathscr{B})}$ is uniquely characterized by Definition 3.2 up to elementary equivalence, i.e., if $(\cdot)[\phi, \mathscr{B}]$ is a proper update function, then $(M, w)[\phi, \mathscr{B}]$ and $\textsc{update}_{(\phi,\mathscr{B})}(M, w)$ are elementary equivalent. We collect the following properties of $\textsc{update}_{(\phi,\mathscr{B})}$.

THEOREM 3.1.

(a) For any objective formula ϕ and set of agents $\mathscr{B}$, it holds that

$$\textsc{update}_{(\phi,\mathscr{B})}(M, w) \models B_b\phi$$

for all $b \in \mathscr{B}$.

(b) If (M, w) satisfies the Kt45 properties, the formula ϕ is objective and $(M, w) \models \phi$, then $\text{UPDATE}_{(\phi,\mathscr{B})}(M, w)$ satisfies the Kt45 properties as well.

(c) Updating is commutative, i.e. the models

$$(M^1, w^1) = \text{UPDATE}_{(\psi,\mathscr{C})}(\text{UPDATE}_{(\phi,\mathscr{B})}(M, w)), \quad \text{and}$$
$$(M^2, w^2) = \text{UPDATE}_{(\phi,\mathscr{B})}(\text{UPDATE}_{(\psi,\mathscr{C})}(M, w))$$

for objective formulas ϕ, ψ and sets of agents $\mathscr{B}, \mathscr{C}$, are bisimilar.

(d) Update is idempotent, i.e., the two models

$$(M^1, w^1) = \text{UPDATE}_{(\psi,\mathscr{B})}(\text{UPDATE}_{(\phi,\mathscr{B})}(M, w)), \quad \text{and}$$
$$(M^2, w^2) = \text{UPDATE}_{(\phi,\mathscr{B})}(M, w)$$

for an objective formula ϕ and a set of agents $\mathscr{B}$, are bisimilar.

Proof. We prove parts (a) to (c). The proof of part (d) is similar to that of part (c). For part (a) we need to prove, that for any objective ϕ, set of agents $\mathscr{B}$ and $b \in \mathscr{B}$, it holds that $(M', w') = \text{UPDATE}_{(\phi,\mathscr{B})}(M, w) \models B_b\phi$. Take $b \in \mathscr{B}$. Since w' is $new(w)$, we have, by definition, for all v, if $R_b(w, v)$ then $(M', v) \models \phi$. Hence, $(M', w') \models B_b\phi$.

We prove part (b) by checking each of the properties for a Kt45 model. Assume that $(M, w) = \langle S, R_1, \ldots, R_m, \pi \rangle$ is transitive, and that

$$\text{UPDATE}_{(\phi,\mathscr{B})}(M, w) = (\langle S, R'_1, \ldots, R'_m, \pi' \rangle, w')$$

is not. Then there is, for some agent $i, (s, t) \in R'_i, (t, u) \in R'_i$ and $(s, u) \notin R'_i$. Because, by definition, there are no arrows from *old* to *new* states and no i has both a relation from *new* to *new* states and from *new* to *old* states, there are only three cases. Firstly, suppose $old(s), old(t)$ and $old(u)$. Then, by definition, we have $(s, t) \in R_i$, $(t, u) \in R_i$ and by transitivity $(s, u) \in R_i$. Then $(old(s), old(u)) \in R'_i$. Contradiction. Now suppose $new(s), old(t)$ and $old(u)$. Then $(s, t) \in R_i, (t, u) \in R_i$ and $i \notin \mathscr{B}$. Again, this implies $(new(s), old(u)) \in R'_i$, which is a contradiction. Finally, suppose $new(s), new(t), new(u)$, then $(s, t) \in R_i, (t, u) \in R_i$ and $(M, u) \models \phi$. Thus, $(new(s), new(u)) \in R'_i$, which is again a contradiction and completes the proof for transitivity. For proving that R'_i is euclidean (for all agents i), we need to consider exactly the same cases and for all these cases a similar argument can be made. For o-seriality, the proof obligation is to show that for all $s \in S'$, there is a $t \in S'$ such that $(s, t) \in R_i \wedge (M, s) \models \psi \leftrightarrow (M, t) \models \psi$, for all objective ψ. Suppose $i \notin \mathscr{B}$, then this follows directly from the definition, since we will have some t such that $(s, old(t)) \in R_i$. Suppose $i \in \mathscr{B}$, then by assumption of M being o-serial, there is a t such that $(s, t) \in R_i$ which agree on the objective formulas. In particular, they agree on ϕ. Suppose $(M, s) \models \phi$, then $(M, t) \models \phi$ and therefore $(new(s), new(t)) \in R_i$. Suppose $(M, s)\ \phi$, then both s and t will be unreachable from w' and will thus have no corresponding state in M'. Thus, in all cases o-seriality is preserved.

For part (c), we observe that four copies of the original states are made. In both cases we have an original copy (call them the *old* ones), a new copy that is made in the first step is called *middle*, and the last copy that is made (so a copy from both the `old` and *middle*) is called *new*. Furthermore, the predicates have a superscript of 1 or 2 depending on the model they belong to. In addition, the elements of the model will have superscripts according to their model. We construct a bisimulation $\mathscr{R} \subseteq S^1 \times S^2$ such that it is minimal with respect to

$$\begin{array}{lcl} \mathscr{R}(old^1(s), old^2(s')) & \Leftrightarrow & s = s' \\ \mathscr{R}(middle^1(s), new^2(s')) & \Leftrightarrow & s = s' \\ \mathscr{R}(new^1(s), middle^2(s')) & \Leftrightarrow & s = s' \end{array}$$

(referring to equality in the original model).

1. $\mathscr{R}$ satisfies forward-choice: Suppose $\mathscr{R}(s, s') \wedge R_i^1(s, t), s,\ t \in S^1$, $s' \in S^2$, then there is a $t' \in S^2$ such that $\mathscr{R}(t, t'), (s', t') \in R^1$. If $old^1(s)$, it is trivially satisfied (take a $old^2(t')$, and it is satisfied, since $old(t)$). Say $middle^1(s), new^2(s')$. If $i \in \mathscr{C}$, then t is apparently a copy from the old^1 states to the $middle^1$ states. But then, it is also copied to the new^2 states and it is reachable for the same reason that is was reachable in the $middle^1$ state. If $i \in \mathscr{B}$, the same argument applies as for $\mathscr{C}$. If $i \notin \mathscr{C}$ and $i \notin \mathscr{B}$, then one can go back to the *old* world again as these worlds are still considered possible for these agents i.
2. $\mathscr{R}$ satisfies backward-choice: The reasoning is similar to the case of forward choice.
3. For all $s \in S,\ s' \in S', (\mathscr{R}(s, s') \Rightarrow \pi(s) = \pi'(s'))$: from the definition of $\mathscr{R}$ and the definition of $\text{UPDATE}_{(\phi,\mathscr{B})}(\cdot, \cdot)$ this is trivial. □

The update operation of Definition 3.1 is restricted to objective formulas. In principle, one can do the same constriction for non-objective ones. However, for a non-objective formula $B_i\phi$, it can happen that, unintendedly, an agent increases the objective knowledge encapsulated by the formula ϕ. This is illustrated by the next example.

EXAMPLE 3.2 (updating of non-objective formula). Suppose we are interested in agent a learning the formula $B_b p \vee B_b \neg p$, but not the property p itself. So, agent a learns that agent b knows about p without getting information about p itself. Consider the Kripke model (M, s) in Figure 3, where $\pi(s)(p) = \pi(u)(p) = \texttt{true}$ and $\pi(t)(p) = \texttt{false}$. This models the state where b knows that p is $\texttt{true}$. Agent a does not know p or $\neg p$,and it does not know if b knows p.

If we apply the definition of the update operation, it results in the model (M', v) from Figure 4, where $\pi'(v)(p) = \pi'(s)(p) = \pi'(u)(p) = \texttt{true}$ and $\pi'(t)(p) = \texttt{false}$. In (M', v) it holds that $B_a p$, as $\pi'(v)(p) = \texttt{true}$, since v was copied from the state s in M. Figure 4 illustrates that a has learned $B_b p \vee B_b \neg p$, but also that a has learned p itself, which we wanted to avoid. The reason that it turns out like this, is because the only state in M where $B_b p \vee B_b \neg p$ holds, is the state s. Thus, all the other states have no corresponding *new* states.

In the next subsection we will define a side-effect function such that a will learn about others, but does not learn any objective formulas itself.

3.2. *Side-effects*

The main reason that an update of $B_b p \vee B_b \neg p$ for agent a has undesired consequences, is that it actually does not include the right arrows between the copies of the original states. The construction, in the case of the non-objective formula $B_b p \vee Bb \neg p$, deletes arrows of a to gain the states that satisfy the updating formula. However, for the rest, we want a to keep all the states it considers possible. Moreover, we do not want to change the knowledge of the other agents. In this subsection we define the functions that accomplish these requirements.

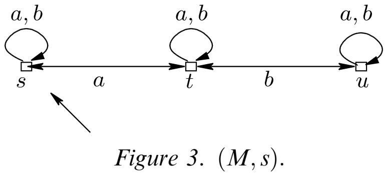

Figure 3. (M, s).

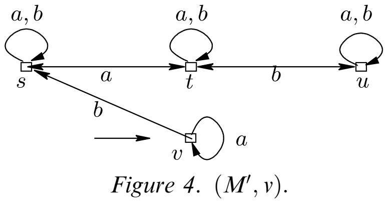

Figure 4. (M', v).

A technical obstacle is that states can be shared among agents. It is obvious that if we change a state with the intention to change the belief of one agent, then the belief of the other agents that consider this state possible, is changed as well. Therefore, the first thing to do, is to separate the states of learning agents from the states of agents that do not learn. This procedure will be called *unfolding*. The tag $new_{\mathscr{B}}$ is a generalization of *new* and *old* from the previous section; the tag *orig* is only used for the point of the model, i.e. the actual world.

DEFINITION 3.3. Given a model (M, w) with $M = \langle S, \pi, R_1, \ldots, R_m\rangle$, and a partitioning $\mathscr{X}$ of $\mathscr{A}$, we define the operation $\textsc{unfold}_{\mathscr{X}}(M, w)$, the unfolding of (M, w) with respect to $\mathscr{X}$, by $\textsc{unfold}_{\mathscr{X}}(M, w) = (\langle S', \pi', R'_1, \ldots, R'_m\rangle, w')$, where

- $S' = \{new_{\mathscr{B}}(s) | s \in S, \mathscr{B} \in \mathscr{X}\} \cup \{orig(w)\}$
- $w' = orig(w)$
- $\pi'(new_{\mathscr{B}}(s))(p) = \pi(s)(p)$ and $\pi'(orig(w))(p) = \pi(w)(p)$ for all $s \in S, p \in \mathscr{P}, \mathscr{B} \in \mathscr{X}$
- for $a \in \mathscr{A}$, the binary relation R'_a on S' is minimal such that

$$R'_a(new_{\mathscr{B}}(s), new_{\mathscr{B}}(t)) \Leftrightarrow R_a(s, t)$$
$$R'_a(orig(w), new_{\mathscr{B}}(s)) \Leftrightarrow R_a(w, s) \text{ and } a \in \mathscr{B}$$

where $\mathscr{B}$ ranges over $\mathscr{X}$.

So, for every group of agents $\mathscr{B}$ there is copy of the original states (viz. $new_{\mathscr{B}}(s)$ for every $s \in S$). The unfold operation does indeed preserve our Kt45 properties and it models the same knowledge, which is captured by the following theorem.

THEOREM 3.2.

(a) If (M, w) is a Kt45 model and $\mathscr{X}$ a partitioning of $\mathscr{A}$, then it holds that $\textsc{unfold}_{\mathscr{X}}(M, w)$ is a Kt45 model too.
(b) For every model (M, w) and partitioning $\mathscr{X}$, it holds that (M, w) and $\textsc{unfold}_{\mathscr{X}}(M, w)$ are bisimilar.

Proof. Part (a) First, we prove R'_a is euclidean under the assumption that R_a is euclidean, for any $a \in \mathscr{A}$. Assume that $R'_a(s', t') \wedge R'_a(s', u'), s', t', u' \in S'$. The proof obligation is that $R'_a(t', u')$ where s', t' and u' are either in one of the partitions or in `orig`. Suppose $s' = new_{\mathscr{W}}(s), t' = new_{\mathscr{Y}}(t)$ and $u' = \mathtt{new}_{\mathscr{Z}}(u)$ for $\mathscr{W}, \mathscr{Y}, \mathscr{Z} \in \mathscr{X}$. From the definition of $R'_a(s', t')$, it follows that

$R_i(s,t) \wedge \mathscr{W} = \mathscr{Y}$, from $R'_a(s',u')$ follows that $R_a(s,u) \wedge \mathscr{W} = \mathscr{Z}$. Since R_a is euclidean, we have $R_a(t,u)$. From $\mathscr{W} = \mathscr{Y}$ and $\mathscr{W} = \mathscr{Z}$ we have $\mathscr{Y} = \mathscr{Z}$. Thus we conclude $R'_a(new_{\mathscr{Y}}(t), new_{\mathscr{Z}}(u)) = R'_a(t',u')$. The only other case is that $s' = orig(w), t' = new_{\mathscr{W}}(t)$ and $u' = new_{\mathscr{W}}(u)$ for some $\mathscr{W} \in \mathscr{X}$. Then $R_a(w,u)$ and $R_a(w,u)$. Thus $R'a(new_{\mathscr{W}}(t), new_{\mathscr{W}}(u))$. The proof that R'_a is transitive is similar to the euclidean proof. O-seriality can be proven directly, by observing that each $new_{\mathscr{W}}$ is a copy of the original model, so the property is preserved inside a partition. Since $\mathscr{A} \neq \emptyset$, it also holds in *orig*, because the world which is a copy of the *orig* in each partition is accessible (which, clearly, have the same valuation).

Part (b) Construct a bisimulation $\mathscr{R} \subseteq S \times S'$ such that, for $u, v \in S$ and $\mathscr{W} \in \mathscr{X}$,

$$\mathscr{R}(u, new_{\mathscr{W}}(v)) \Leftrightarrow u = v \text{ and } \mathscr{R}(u, orig(w)) \Leftrightarrow u = w$$

We check the various properties.

$\mathscr{R}$ satisfies forward-choice: Suppose $\mathscr{R}(s,s')$ and $R_i(s,t), s,t \in S$, $s' \in S'$, $a \in \mathscr{A}$. If $s' = new_{\mathscr{W}}(s)$, then by the definition of $\text{UNFOLD}_{\mathscr{X}}(\cdot,\cdot)$ and $R_a(s,t)$ we have $R'_a(new_{\mathscr{W}}(s), new_{\mathscr{W}}(t))$. If $s' = orig(s)$, we have $R'_a(orig(w), new_{\mathscr{W}}(t))$ for some $\mathscr{W} \in \mathscr{X}$ with $a \in \mathscr{W}$. Furthermore, for all cases of t' we get $\mathscr{R}(t,t')$.

$\mathscr{R}$ satisfies backward-choice: Suppose $\mathscr{R}(s,s')$ and $R_a(s',t'), s \in S$, $s',t' \in S'$. For all cases of $R'_a(s',t')$, we immediately get $R_a(s,t)$. Also, for all cases of t' we have $\mathscr{R}(t,t')$.

$\mathscr{R}(s,s') \Rightarrow \pi(s) = \pi'(s')$ for all $s \in S$, $s' \in S'$: This is immediate, from the definition of $\mathscr{R}$ and the definition of $\text{UNFOLD}_{\mathscr{X}}(\cdot,\cdot)$. □

EXAMPLE 3.3 (UNFOLDING). Consider the Kripke model (M,s) in Figure 5 with $\pi(s)(p) = \pi(u)(p) =$ `true`, $\pi(t)(p) =$ `false`. So, b knows that p is `true`, while a does not. Furthermore, a does not know if b knows p. Now the operation we perform is $\text{UNFOLD}_{\{\{a\},\{b\}\}}(M,s)$, thus $\{a,b\}$ is split into $\{a\}$ and $\{b\}$, which results in the model (M',s) in Figure 6.

So, we have separated the knowledge of a and b. In Figure 6, the state s is the original state, the primed states model a's knowledge and the double primed states model b's knowledge. Thus, the upper half

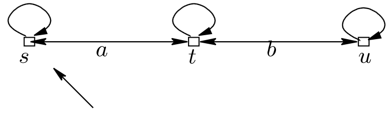

Figure 5. (M,s).

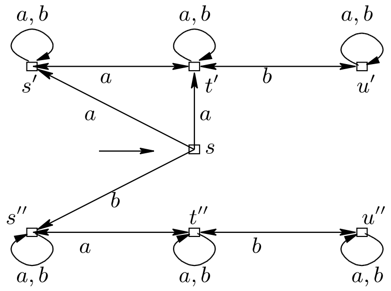

Figure 6. (M', s).

of the model represents the knowledge of a, and the lower half represents the knowledge of b. Note that no states are shared, in particular because the point of the model is not reflexive.

Now, we give some preparatory definitions leading to the formulation of a side-effect in Definition 3.9. First, we define the notion of a partial submodel.

DEFINITION 3.4. A model $M = \langle S, \pi, R_1, \ldots, R_m \rangle$ is a partial submodel of $M' = \langle S', \pi', R'_1, \ldots, R'_m \rangle$, notation $M \sqsubseteq M'$, iff $S \subseteq S'$, $\pi(s)(p) = \pi'(s)(p)$ for all $s \in S$, $p \in \mathscr{P}$ and $R_i \subseteq R'_i$.

Note that a partial submodel is not pointed. Our notion of a partial submodel is slightly more liberal compared to the standard notion of a submodel, as here we allow to drop arrows. It is for technical reasons, viz. the handling of the atom split operation and the operation for side-effects below, that we have occasion to consider partial submodels here.

Next, we construct a partial submodel that represents the knowledge of a group of agents $\mathscr{B}$.

DEFINITION 3.5. Given a model (M, w) such that

$$(M, w) = (\langle S, \pi, R_1, \ldots, R_m \rangle, w) = \textsc{unfold}_{\mathscr{B}, \mathscr{A}/\mathscr{B}}(M', w')$$

for some (M', w'), define $\textsc{sub}_{\mathscr{B}}(M)$, the submodel of M for $\mathscr{B}$, by $\textsc{sub}_{\mathscr{B}}(M) = \langle S', \pi', R'_1, \ldots, R'_m \rangle$ where

- $S' = \{new_{\mathscr{B}}(s) | s \in S\} \cup \{orig(w)\}$
- $\pi'(s)(p) \Leftrightarrow \pi(s)(p)$ for all $s \in S', p \in \mathscr{P}$,
- for all $a \in \mathscr{A}, R'_a(s, t) \Leftrightarrow R_a(s, t) \wedge s, t \in S'$.

Clearly a $\mathscr{B}$-submodel is a partial submodel in the sense of Definition 3.4. The restmodel is the complementary part of the model that is the complement with respect to the accessibility relation of a given partial submodel.

DEFINITION 3.6. Given a model $M = \langle S, \pi, R_1, \ldots, R_m \rangle$ and a partial submodel $N = \langle S'', \pi'', R_1'', \ldots, R_m'' \rangle$ of M, define $\textsc{rest}_\mathrm{N}$, the restmodel of N in M, by $\textsc{rest}_N(M) = \langle S', \pi', R_1', \ldots, R_m' \rangle$ where

- $s \in S' \Leftrightarrow s \in S \wedge \exists a \in \mathscr{A}\, \exists (u, v) \in R_a'(u = s \vee v = s)$
- $\pi'(s)(p) \Leftrightarrow \pi(s)(p)$ for all $s \in S', p \in \mathscr{P}$
- for all $a \in \mathscr{A}, R_a'(s, t) \Leftrightarrow R_a(s, t) \wedge \neg R_a''(s, t)$

We can see the partial submodel and restmodel definitions in action by taking the model of Example 3.3 and applying the above definitions (see Figure 7). This exactly corresponds to the idea of two submodels that represent the belief of different agents.

Now, we would like to update the belief of some agents. To this end, we want to replace the submodel that represents their belief by a new model. We will apply the following definition.

DEFINITION 3.7. Given a model $N = \langle S, \pi, R_1, \ldots, R_m \rangle$, a model M, a model N' such that $N \sqsubseteq N' \sqsubseteq M$ with $\textsc{rest}_{N'}(M) = \langle S', \pi', R_1', \ldots, R_m' \rangle$, we define the operation $\textsc{replace}_{N'}(N, M)$, the replacement of N' by N in M, by $\textsc{replace}_{N'}(N, M) = \langle S'', \pi'', R_1'', \ldots, R_m'' \rangle$ where

- $s \in S'' \Leftrightarrow s \in S \vee s \in S'$,
- $\pi''(s)(p) \Leftrightarrow \pi(s)(p)$ for all $s \in S'', p \in \mathscr{P}$
- for all $a \in \mathscr{A}, R_a''(s, t) \Leftrightarrow (s, t) \in R_a \vee (s, t) \in R_a'$.

The idea is, that once the belief is completely separated, we cannot only safely change the belief of certain agents, but also preserve the

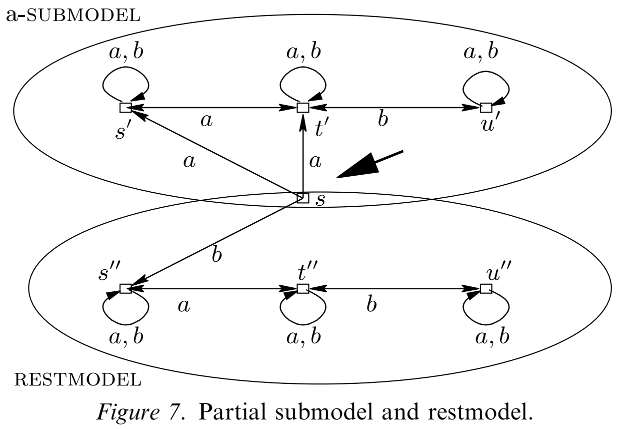

Figure 7. Partial submodel and restmodel.

Kt45 properties. The operation $\text{ATOMSPLIT}_{(\phi,\mathscr{B})}$ removes the arrows for agents in the group $\mathscr{B}$ between states that have a different valuation for the objective formula ϕ.

DEFINITION 3.8. Given a model $M = \langle S, \pi, R_1, \ldots, R_m \rangle$ and objective formula ϕ, we define an operation $\text{ATOMSPLIT}_{(\phi,\mathscr{B})}(M) = \langle S', \pi', R'_1, \ldots, R'_m \rangle$ as follows.

- $S = S'$,
- $\pi'(s)(p) \Leftrightarrow \pi(s)(p)$ for all $s \in S', p \in \mathscr{P}$,
- for $a \in \mathscr{B}, R'_a(s,t) \Leftrightarrow R_a(s,t) \wedge M, s \models \phi \Leftrightarrow M, t \models \phi$,
- for $a \notin \mathscr{B}, R'_a(s,t) \Leftrightarrow R_a(s,t)$.z

Finally we are in a position to define the actual side-effect function that ties these things together.

DEFINITION 3.9. For a model (M', w'), a set of agents $\mathscr{B}$ and an objective formula ϕ such that $(M', w') = \text{UNFOLD}_{\{\mathscr{B},\mathscr{A}/\mathscr{B}\}}(M, w)$ and $N = \text{SUB}_{\mathscr{B}}(M')$ we define the operation $\text{SIDE-EFFECT}_{(\phi,\mathscr{B},\mathscr{C})}(M, w)$, the side-effect for agents in $\mathscr{B}$ with respect to the agents in $\mathscr{C}$ and the formula ϕ, by

$$\begin{aligned}&\text{SIDE-EFFECT}_{(\phi,\mathscr{B},\mathscr{C})}(M, w)\\&\quad = (\text{REPLACE}_N(\text{ATOMSPLIT}_{(\phi,\mathscr{C})}(N), M'), w').\end{aligned}$$

Note, that the formula ϕ in Definition 3.9 is required to be objective (cf. Example 3.2).

EXAMPLE 3.4. We continue Example 3.3. Consider the a-submodel of M. We now apply $\text{ATOMSPLIT}_{(p,b)}$ on this model which results in the model (M'', s) in Figure 8. The arrow (t', u') has disappeared, since $\pi(t)(p) \neq \pi(u)(p)$. Therefore, u is not reachable anymore, and can be

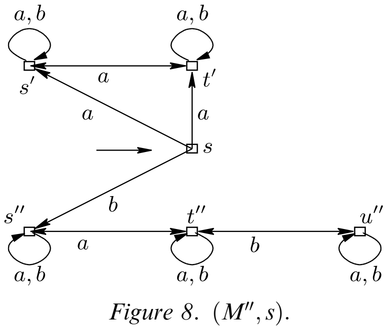

Figure 8. (M'', s).

dropped. Notice, that a believes $B_b p \vee B_b \neg p$, while a has learned nothing about p itself, as was the case for Example 3.2.

A typical application of the side-effect function is of the form SIDE-EFFECT$_{(p,\mathscr{A},b)}$ where all agents collectively learn that agent b knows about the atom p.

We introduce the notion of interconnection of relations, that comes in handy for a proof of the preservation of the Kt45 properties by the side-effect operation. Two binary relations A and B are called interconnected iff there is a $(w, v) \in A$ and $(s, t) \in B$ such that $w = s, w = t, v = s$ or $v = t$. If two binary relations are not interconnected, we call them separated. Separateness is useful because of its following properties.

LEMMA 3.1.

(a) If binary relations A and B are separated and are both Kt45, then $A \cup B$ is also Kt45.
(b) If A and B are separated and $A \cup B$ has the Kt45 properties then both A and B have the Kt45 properties.

Proof. (a) We restrict ourselves only to the proof that union preserves the euclidean property. Assume $C = A \cup B$ not euclidean. Then there is $(s, t) \in C$ and $(s, u) \in C$ and $(t, u) \notin C$. Observe that (1) (s, t) and (s, u) cannot both come from A or both come from B, since those relations were both euclidean, and that would mean $(t, u) \in C$ and (2)(s, t) and (s, u) cannot come from the distinct subsets since that would contradict the interconnection property. So, in conclusion (s, t) and (s, u) cannot be elements of C. This directly contradicts the assumption. Hence C is euclidean.

(b) Because of symmetry we only have to prove this for A. Suppose $(s, t) \in A, (s, u) \in A$. Since $C = A \cup B$ is euclidean, C must contain (t, u). But because A and B are not interconnected, (t, u) must be part of A. Therefore, A is euclidean. Proofs for transitivity and o-seriality are similar. □

We have seen that sets that are separated can be split and joined together without changing the Kt45 properties. In the next lemma we apply this for the replace operation.

LEMMA 3.2. Let $f : \mathscr{M} \rightarrow \mathscr{M}$ be an operation on the class of models such that $(i) f(M) \sqsubseteq M$ for any model $M \in \mathscr{M}$ and (ii) preserves Kt45 properties. Suppose model $(M, w) = \textsc{unfold}_{\{\mathscr{B}, \mathscr{A} \setminus \mathscr{B}\}}(M', w')$ and $N = \textsc{sub}_{\mathscr{B}}(M)$. Then the operation defined by $\textsc{replace}_N(f(N), M)$ preserves the Kt45 properties too.

Proof. From the definition of UNFOLD, it is quite easy to see, that N is separated (for all $R_a, a \in A$) with the rest, since for a there's only a relation between `orig` and the partition where a is in, and for the other agents the relation from `orig` to the partition of a does not exist. So from Lemma 3.1 and the fact that M has the Kt45 properties, we must conclude that both the $\mathscr{B}$-submodel (N) and the restmodel have the Kt45 properties. Since f preserves the Kt45 properties, $f(N)$ also has the Kt45 properties too. By observing that doing a $\textsc{replace}_N$ is the same as doing a union of the accessibility-relations of the $\mathscr{B}$-restmodel with the new replacement, we can now use Lemma 3.1, and conclude that $\textsc{replace}_N(f(N), M)$ has the Kt45 properties as well. □

In order to apply the above lemma we check that splitting preserves Kt45.

LEMMA 3.3. Given a Kt45-model $(M, w) = (\langle S, \pi, R_1, \ldots, R_m \rangle, w)$, then, for an objective formula ϕ and subset of agents $\mathscr{B}$, the model $\textsc{atomsplit}_{(\phi, \mathscr{B})}(M) = \langle S', \pi', R'_1, \ldots, R'_m \rangle$ has all the Kt45 properties too.

Proof. For proving the new model is euclidean, suppose R_i is euclidean and $(s, t) \in R'_i \wedge (s, u) \in R'_i$. From $(s, t) \in R'_i$ follows that $\pi(s)(p) = \pi(t)(p)$ and $(s, t) \in R_i$. From $(s, u) \in R'_i$ follows that $\pi(s)(p) = \pi(u)(p)$ and $(s, u) \in R_i$. Thus, $\pi(t)(p) = \pi(u)(p)$ and therefore if $(t, u) \in R_i$, then $(t, u) \in R'_i$. Hence, $(t, u) \in R'_i$. The proof for transitivity is similar. For preservation of o-seriality we suppose $s \in S'$. By definition $s \in S$ and there is some $t \in S$ such that $(s, t) \in R_i$ and $(M, s) \models \varphi \leftrightarrow (M, t) \models \varphi$, where φ is objective. In particular then it holds that $\pi(s)(p) = \pi(t)(p)$. By definition $(s, t) \in R'_i$. □

We are now in a position to prove a number of properties of the side-effect operation.

THEOREM 3.3.

(a) If (M, w) is a Kt45-model, then $\textsc{side-effect}_{(\phi, \mathscr{B}, \mathscr{A})}(M, w)$, for any objective formula ϕ and sets of agents $\mathscr{B}, \mathscr{C}$, is a Kt45-model as well.

(b) (commutativity of side-effect) Given a model (M, w), sets of agents $\mathscr{B}, \mathscr{C}, \mathscr{D}, \mathscr{E}$ and two formulas ϕ, ψ, it holds that

$$\text{SIDE–EFFECT}_{(\phi,\mathscr{D},\mathscr{E})}(\text{SIDE–EFFECT}_{(\psi,\mathscr{B},\mathscr{C})}(M, w))$$

and $\text{SIDE-EFFECT}_{(\psi,\mathscr{B},\mathscr{C})}(\text{SIDE-EFFECT}_{(\phi,\mathscr{D},\mathscr{E})}(M, w))$ are bisimilar.

(c) (swapping update and side-effect) Given a model (M, w), sets of agents $\mathscr{B}, \mathscr{C}, \mathscr{D}$, *a* formula ϕ and an objective formula ψ, it holds that the models

$$\text{SIDE–EFFECT}_{(\phi,\mathscr{C},\mathscr{D})}(\text{UPDATE}_{(\psi,\mathscr{B})}(M, w))$$

and

$$\text{UPDATE}_{(\psi,\mathscr{B})}(\text{SIDE–EFFECT}_{(\phi,\mathscr{C},\mathscr{D})}(M, w))$$

are bisimilar.

(d) (idempotency of side-effect) Given a model (M, w), sets of agents $\mathscr{B}, \mathscr{C}, \mathscr{D}$ and *a* formula ϕ, it holds that

$$\text{SIDE–EFFECT}_{(\phi,\mathscr{B},\mathscr{C})}(\text{SIDE–EFFECT}_{(\phi,\mathscr{B},\mathscr{C})}(M, w))$$

and

$$\text{SIDE–EFFECT}_{(\phi,\mathscr{B},\mathscr{C})}(M, w)$$

are bisimilar.

Proof. (a) Clearly, $\text{ATOMSPLIT}_{(p,\mathscr{C})}(M) \sqsubseteq M$ holds. Therefore, the statement follows from Lemma 3.2 and Lemma 3.3.

(b) There are several tagged states. We have $orig, new_{\mathscr{B}}, new_{\mathscr{A}\setminus\mathscr{B}}, new_{\mathscr{D}}, new_{\mathscr{A}\setminus\mathscr{D}}$ in both models.

Construct a bisimulation $\mathscr{R} \subseteq S \times S'$ such that

$$\begin{array}{lcl}
\mathscr{R}(orig(w), orig(v)) & \Leftrightarrow & w = v \\
\mathscr{R}(new_{\mathscr{B}}(u), new_{\mathscr{B}}(v)) & \Leftrightarrow & u = v \\
\mathscr{R}(new_{\mathscr{A}\setminus\mathscr{B}}(u), new_{\mathscr{A}\setminus\mathscr{B}}(v)) & \Leftrightarrow & u = v \\
\mathscr{R}(new_{\mathscr{D}}(u), new_{\mathscr{D}}(v)) & \Leftrightarrow & u = v \\
\mathscr{R}(new_{\mathscr{A}\setminus\mathscr{D}}(u), new_{\mathscr{A}\setminus\mathscr{D}}(v)) & \Leftrightarrow & u = v
\end{array}$$

We can now follow the reasoning from Theorem 3.1 to see that this bisimulation has all the desired properties for bisimulation. For example, $\mathscr{R}$ satisfies forward-choice: suppose $\mathscr{R}(s, s') \wedge R_a(s, t)$, $s, t \in S, s' \in S'$. Suppose $orig(s)$, then t could be $new_{\mathscr{B}}$ or $new_{\mathscr{A}\setminus\mathscr{B}}$. But this copy of t is present in S' as well, since that was created after the first execution of SIDE-EFFECT on the original model. And indeed, there's a $orig(w)$ such that $(w, t) \in S'$. The checks for all the other options are similar.

(c) Construct a bisimulation $\mathscr{R} \subseteq S \times S'$ such that:

$$\begin{array}{lcl} \mathscr{R}(orig(w), orig(v)) & \Leftrightarrow & w = v \\ \mathscr{R}(old(u), old(v)) & \Leftrightarrow & u = v \\ \mathscr{R}(new(u), new(v)) & \Leftrightarrow & u = v \\ \mathscr{R}(new_{\mathscr{B}}(u), new_{\mathscr{B}}(v)) & \Leftrightarrow & u = v \\ \mathscr{R}(new_{\mathscr{A}/\mathscr{B}}(u), new_{\mathscr{A}/\mathscr{B}}(v)) & \Leftrightarrow & u = v \end{array}$$

Checking all the properties is similar to (b).

(d) The result of applying the same side-effect operation twice is the same model as applying it just once modulo a number of unreachable states. Again, construct a bisimulation $\mathscr{R}$ similar to all the previous proofs such that the relation exists if they are a copy of each other and reachable from the point of the model. The unprimed variables belong to $\text{SIDE-EFFECT}_{(\phi,\mathscr{B},\mathscr{C})}(M, w)$ and the primed ones to $\text{SIDE-EFFECT}_{(\phi,\mathscr{B},\mathscr{C})}(\text{SIDE-EFFECT}_{(\phi,\mathscr{B},\mathscr{C})}(M, w))$. Now, $\mathscr{R}$ satisfies forward-choice: suppose $\mathscr{R}(s, s') \wedge R_a(s, t), s, t \in S, s' \in S'$. Suppose $new_{\mathscr{B}}(s)$, then we indeed have such a copy $new_{\mathscr{B}}(t')$ with $\mathscr{R}(t, t')$, since no more arrows inside $new'_{\mathscr{B}}$ were deleted, because only the states from $new_{\mathscr{B}}$ are reachable in $new'_{\mathscr{B}}$. For the other cases it it trivial. It can also be shown that it satisfies the other properties that are required. □

Next, we consider how the formulas are altered by the side-effect operation. We will partially answer this by presenting a few interesting formulas that hold in the resulting model. We distinguish

1. the group of agents $\mathscr{B}$ that learn about other agents, ranged over by b;
2. the group of agents $\mathscr{C}$ that is learned about, ranged over by c;
3. other agents in the group $\mathscr{D}$, ranged over by d.

The fact that the agents in $\mathscr{B}$ are the only agents that learn at all, is clear. The other agents consider exactly (copies of) their old worlds possible; their belief has not changed. With this in mind, we present a few properties of the side-effect operation. Below $C_{\mathscr{B}\mathscr{C}\mathscr{D}}$ expresses common knowledge of agents in the sets $\mathscr{B}, \mathscr{C}$ and $\mathscr{D}$.

LEMMA 3.4. Given a model (M, w), disjoint sets of agents $\mathscr{B}, \mathscr{C}, \mathscr{D}$ and a formula ϕ, put $(M', w') = \text{SIDE–EFFECT}_{(\phi,\mathscr{B},\mathscr{C})}(M, w)$. Then it holds that

(a) $(M', w') \models B_b(B_c\phi \vee B_c\neg\phi)$;
(b) $(M', w') \models B_b\psi$ iff $(M, w) \models B_b\psi$ for any objective formula ψ;
(c) $(M', w') \models B_bC_{\mathscr{B}\mathscr{C}\mathscr{D}}(B_c\phi \vee B_c\neg\phi)$;

(d) $(M', w') \models B_a\psi$ iff $(M, w) \models B_a\psi$ for any formula ψ and any agent $a \notin \mathscr{B}$.

Proof. We prove the typical cases of part (b) and (c).

(b) The unfolded model of (M, w) is bisimilar with (M', w'), and no arrows of b were deleted afterwards. Hence, the knowledge of b about objective formulas has not changed.

(c) We have already seen that in every state of $\textsc{sub}_{\mathscr{B}}(M')$ it holds that $B_c\phi \vee B_c\neg\phi$. Now, what we need to prove is that the path $w' \xrightarrow{b} s_1 \xrightarrow{i} s_2 \xrightarrow{j} \ldots$ with $i, j, \ldots, \in \mathscr{A}$ is a path to a state where it holds that $B_b p \vee B_b\neg p$. But since $s_1 \in new_{\mathscr{B}}$, and since there are no arrows from $new_{\mathscr{A}}$ to other partitions, all s_k are elements of $new_{\mathscr{B}}$. Thus any s_k is part of $\textsc{sub}_b(M')$. By the construction of $\textsc{side-effect}_{(\phi,\mathscr{B},\mathscr{C})}(M, w)$, in any state in $new_{\mathscr{B}}$, we either have all arrows of c to a world where ϕ holds (at least one, by o-seriality) or to a world where $\neg\phi$ holds. Furthermore, $B_c\phi$ or $B_c\neg\phi$ holds. Hence, $B_c\phi \vee B_c\neg\phi$ holds. □

In part (a) of the above lemma, an agent b obtains derived knowledge of an agent c. Part (b) states that no objective knowledge is learned. Part (c) phrases that an agent b considers the rest of the agents as smart itself. Finally, part (d) captures that other agents do not learn.

Property (c) may or may not be a reasonable assumption of b about the other agents. If one agent believes that another agent knows the value of ϕ, then it is reasonable to assume that another agent will believe the same. On the other hand common knowledge might be too strong to assume.

Next, we address the issue that an agent b shares its belief about an agent c with only some other agents. We represent this by linking b's beliefs of those other agents back to the original (unmodified) states. We distinguish four different type of groups of agents.

1. the groups $\mathscr{B}$ and $\mathscr{C}$ are as before;
2. the group $\mathscr{D}$ of agents of which agents in $\mathscr{B}$ believe they have learned in common about agents in group $\mathscr{C}$, ranged over by d;
3. the group $\mathscr{E}$ of agents of which agents in $\mathscr{B}$ believe they have learned nothing about, ranged over by e.

We define the new side-effect operation 0-$\textsc{unfold}$ that handles this refinement. Here, 0 refers to zero-knowledge for the group of agents $\mathscr{E}$. As before, we define an unfolding operation first.

DEFINITION 3.10. Given a model (M, w), with $M = \langle S, \pi, R_1, \ldots, R_m\rangle$, and a partitioning $\mathscr{X} = \{\mathscr{B}, \mathscr{C}, \mathscr{D}, \mathscr{E}\}$ of the set of agents $\mathscr{A}$, we define the operation 0-UNFOLD$_{\mathscr{X}}(M, w)$, the zero-knowledge unfolding of (M, w) with respect to $\mathscr{X}$, by 0-UNFOLD$_{\mathscr{X}}(M, w) = (\langle S', \pi', R'_1, \ldots, R'_m\rangle, w')$, where

- $S' = new_{\mathscr{B}}(S) \cup new_{\mathscr{C}\mathscr{D}\mathscr{E}}(S) \cup \{orig(w)\}$
- $w' = orig(w)$
- $\pi'(new_{\mathscr{Y}}(v))(p) = \pi(v)(p)$ and $\pi'(orig(w))(p) = \pi(w)(p)$ for all $p \in \mathscr{P}, \mathscr{Y} \in \{\{\mathscr{B}\}, \{\mathscr{C}\mathscr{D}\mathscr{E}\}\}$
- for $a \in \mathscr{A}, R'_a$ on S' is the minimal binary relation such that

$$\begin{array}{lcl} R'_a(orig(w), new_{\mathscr{B}}(v)) & \Leftrightarrow & R_a(w, v) \wedge a \in \mathscr{B} \\ R'_a(orig(w), new_{\mathscr{C}\mathscr{D}\mathscr{E}}(v)) & \Leftrightarrow & R_a(w, v) \wedge a \notin \mathscr{B} \\ R'_a(new_{\mathscr{B}}(u), new_{\mathscr{C}\mathscr{D}\mathscr{E}}(v)) & \Leftrightarrow & R_a(u, v) \wedge a \in \mathscr{E} \\ R'_a(new_{\mathscr{B}}(u), new_{\mathscr{B}}(v)) & \Leftrightarrow & R_a(u, v) \wedge a \notin \mathscr{D} \\ R'_a(new_{\mathscr{C}\mathscr{D}\mathscr{E}}(u), new_{\mathscr{C}\mathscr{D}\mathscr{E}}(v)) & \Leftrightarrow & R_a(u, v) \end{array}$$

So, instead of completely separating the knowledge of agents in $\mathscr{B}$ with the other agents, we share this knowledge with the other agents. Since the other agents do not learn anything, agents in $\mathscr{B}$ does not gain knowledge about $\mathscr{E}$. We present a theorem similar to Theorem 3.2.

THEOREM 3.4.

(a) If (M, w) is a Kt45 model, then so is 0-UNFOLD(M, w).
(b) For every model (M, w), it holds that (M, w) and 0-UNFOLD(M, w) are bisimilar.

Proof. (a) The accessibility relations of every agent other than those in $\mathscr{E}$ are constructed exactly the same way as in Definition 3.3. Since that operation preserves the Kt45 properties, we do not have to prove it for those agents. A proof for the agents in $\mathscr{E}$ now follows. For transitivity assume $(s, t) \in R'_a, (t, u) \in R'_a$. By the definition, we immediately see that u is of the form $new_{\mathscr{C}\mathscr{D}\mathscr{E}}(\cdot)$. Also we see that if there is a $R_a(s, u)$ then there's a $R'_a(s, u)$. Thus, we can conclude $R'_a(s, u)$. The proof that R'_a euclidean is similar. For o-seriality observe that every arrow of an agent in $\mathscr{E}$ ends in a state that used to be its old state. Thus, o-seriality is preserved.

(b) Construct a bisimulation $\mathscr{R} \subseteq S \times S'$ such that

$$\mathscr{R}(u, new_{\mathscr{Y}}(v)) \Leftrightarrow u = v \text{ and } \mathscr{R}(u, orig(v)) \Leftrightarrow u = v.$$

We prove that

$\mathscr{R}$ satisfies forward-choice: Suppose $\mathscr{R}(s,s')$ and $R_a(s,t), s,t \in S$, $s' \in S'$. Suppose $orig(s)$, then we have $new_{\mathscr{B}}(t)$ if $a \in \mathscr{B}$, else we have $new_{\mathscr{B}\mathscr{C}\mathscr{D}}(t)$ if $a \notin \mathscr{B}$. Suppose $new_{\mathscr{C}\mathscr{D}\mathscr{E}}(s)$, then we immediately have $new_{\mathscr{C}\mathscr{D}\mathscr{E}}(t)$ for all agents a. Suppose $new_{\mathscr{B}}(s)$, then we have $new_{\mathscr{B}}(t)$ if $a \notin \mathscr{E}$, else $new_{\mathscr{C}\mathscr{D}\mathscr{E}}(t)$.

$\mathscr{R}$ satisfies backward-choice: Suppose $\mathscr{R}(s,s')$ and $R_a(s',t')$, $s \in S, s',t' \in S'$. We have that $R_a(s',t')$ immediately implies $R_a(s,t)$ for some the s,t such that s',t' are copies from s,t. This implies $\mathscr{R}(t,t')$.

$\mathscr{R}(s,s') \Rightarrow \pi(s) = \pi'(s')$ for $s \in S, s' \in S'$: Direct from the definition of $\mathscr{R}$. □

Due to the case distinction for arrows leaving the point $orig(w)$, in Definition 3.10 above, it holds that the knowledge of $\mathscr{B}$-agents about $\mathscr{C}$-agents is separated, with the knowledge of other agents about agents in $\mathscr{C}$ or of $\mathscr{C}$-agents themselves. So we can 'cut out' the submodel containing the c arrows from the belief of b. In this definition we can omit the point of the model in our partial submodel, which makes it slightly easier.

DEFINITION 3.11. Given a model (M',w'), with $(M',w') = \langle S', \pi', R'_1, \ldots, R'_m \rangle$, an objective formula ϕ and a partitioning $\mathscr{X}$ of $\mathscr{A}$ (as in Definition 3.10) such that $(M',w') =$ 0-UNFOLD$_{\mathscr{X}}(M,w)$, for some (M,w), we define SUB$_{\mathscr{B}}(M') = \langle S'', \pi'', R''_1, \ldots, R''_m \rangle$ where

- $S'' = \{new_{\mathscr{B}}(s) | s \in S\}$
- $\pi''(s)(p) \Leftrightarrow \pi(s)(p)$ for all $s \in S', p \in \mathscr{P}$
- $R''_c(s,t) \Leftrightarrow R'_c(s,t)$ for $s,t \in S''$
- $R''_a = \emptyset$ $(a \notin \mathscr{C})$.

The operation 0-SIDE-EFFECT$_{(\phi,\mathscr{X})}$, the zero-knowledge side-effect of ϕ with respect to the partitioning $\mathscr{X}$, is then given by

$$\begin{aligned}&0\text{-SIDE-EFFECT}_{(\phi,\mathscr{X})}(M,w)\\&\quad = (\text{REPLACE}_N(\text{ATOMSPLIT}_{(\phi,\mathscr{C})}(N), M'), w')\end{aligned}$$

where $N =$ SUB$_{\mathscr{B}}(M')$.

The operation 0-SIDE-EFFECT has algebraic properties comparable to those of the previous side-effect operation (cf. Theorem 3.3). We also have to following result, corresponding to Lemma 3.4.

LEMMA 3.5. Given a model (M, w), a partitioning $\mathscr{X}$ of $\mathscr{A}$ and an objective formula ϕ such that the model $(M', w') = 0\text{-SIDE-EFFECT}_{(\phi,\mathscr{X})}(M, w)$, it holds that

(a) $(M', w') \models B_b(B_c\phi \vee B_c\neg\phi)$.
(b) $(M', w') \models B_b\psi$ iff $(M, w) \models B_b\psi$ for any objective formula ψ.
(c) $(M', w') \models B_bC_{\mathscr{B}\mathscr{C}\mathscr{D}}(B_c\phi \vee B_c\neg\phi)$.
(d) $(M', w') \models B_a\psi$ iff $(M, w) \models B_a\psi$ for $a \notin \mathscr{B}$ for any formula ψ.
(e) $(M', w') \models B_bB_e\psi$ iff $(M, w) \models B_bB_e\psi$ for any formula ψ.

Proof. We provide the proof for (e). $(\Rightarrow)$ Assume $(M', w') \models B_bB_e\psi$, then we have for all paths $w \xrightarrow{b} s \xrightarrow{e} t, t \models \psi$. But, since all arrows of e in $new_{\mathscr{B}}$ point back to (a copy of) the original model, we have that $(M, t) \models \psi.(\Leftarrow)$ Assume $(M, w) \models B_bB_e\psi$. Now, we have a path in the original model $w \xrightarrow{b} s \xrightarrow{e} t$. In the construction of (M, w) no arrows of b and e are ever added nor deleted. □

Parts (a) to (d) are correspond to the properties given in Lemma 3.4. Part (e) states that the knowledge of agent b about agent e has not changed, which is exactly as desired.

EXAMPLE 3.5 (alternative side-effect function). Recall the model (M, s) from Example 3.3 (Figure 5). We now present this model with four agents $\{b, c, d, e\}$ in Figure 9, with a partitioning into singletons, such that $\pi(s)(p) = \pi(u)(p) = \texttt{true}$ and $\pi(t)(p) = \texttt{false}$. Now, apply $0\text{-SIDE-EFFECT}_{(p,b,c)}(M, w)$ and we gain the model (M', s) from Figure 10 such that $\pi(s)(p) = \pi(s')(p) = \pi(s'')(p) = \pi(u')(p) = \texttt{true}$, $\pi(t)(p) = \pi(t')(p) = \texttt{false}$. Note that in this model, b still knows exactly the same about e as it did before.

3.3. *Comparison with the Action Model Approach*

Baltag and Moss (Baltag et al., 1998; Baltag and Moss 2004) propose a framework for describing epistemic actions using action models. Similar to Kripke models that describe the uncertainty of agents

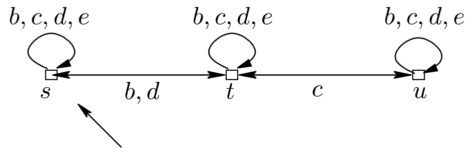

Figure 9. (M, s).

about which world they are in, they use Kripke models to describe the uncertainty of agents about the action that is being performed.

Formally, an epistemic action model is a triple $\Sigma = \langle \Sigma, \overset{\mathscr{A}}{\rightarrow}, pre \rangle$, where Σ is a set of *simpleactions*, $\overset{\mathscr{A}}{\rightarrow}$ is an accessibility relation of agents on actions and the precondition pre is a mapping pre: $\Sigma \rightarrow \Phi$ with Φ being the collection of all epistemic propositions. The central operation of updating an epistemic model $M = \langle S, R_1, \ldots, R_m, \pi \rangle$ as we have used so far with such an action model Σ is defined as $M \otimes \Sigma = \langle S \otimes \Sigma, R'_1, \ldots, R'_m, \pi' \rangle$, where

- $S \otimes \Sigma = \{(s, \sigma) \in S \times \Sigma | (M, s) \models \text{pre}\ (\sigma)\}$
- $R'_i((s, \sigma), (s', \sigma'))$ iff $R_i(s, s')$ and $\sigma \overset{\mathscr{A}}{\rightarrow} \sigma'$
- $\pi'((s, \sigma))(p)$ iff $\pi(s)(p)$

In Baltag and Moss (2004), Baltag and Moss provide, based on this notion of updating, illuminating examples and study several interesting applications of this idea such as the public and private learning of, what we would call, objective formulas. This poses the question if we can describe the more complicated updates as well.

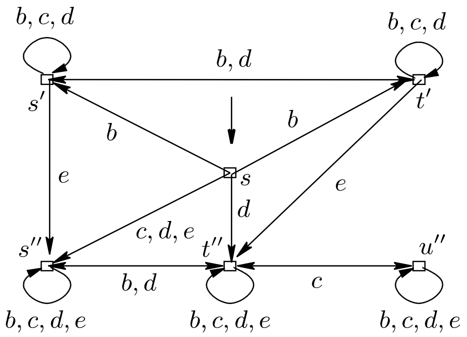

Figure 10. (M', s).

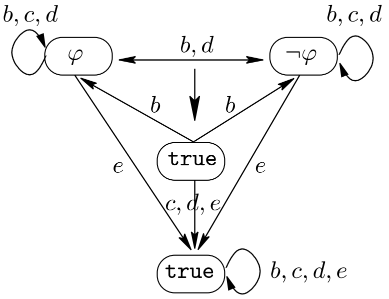

Figure 11. 0-SIDE-EFFECT action model.

In Figure 11 the ovals depict a precondition with φ some objective formula. Intuitively, this action model corresponds to the operation of 0-SIDE-EFFECT.

Indeed, the update product of the Kripke structure of Example 3.5 in Figure 9 and the action model of Figure 11 results in the model of Figure 10. Also, omitting the *e*-arrows from the action model in Figure 11 yields an action model that for the concrete examples discussed above corresponds to the side-effect operation of Definition 3.9. However, currently we have no proof that such a correspondence holds in general.

4. A LOGICAL LANGUAGE FOR SECURITY PROTOCOLS

In this section we exploit the ideas of the previous section for a logical language to reason about security protocols. The UPDATE and SIDE-EFFECT operations are used for its semantics. We introduce so-called transition rules for the modeling of security protocols, that we discuss in the next section.

DEFINITION 4.1. Fix a set of proposition $\mathscr{P}$, ranged over by p, and a set of agents $\mathscr{A}$ of m elements, ranged over by i,j. The language $\mathscr{L}_{\mathscr{C}}$ is given by

$$\phi ::= p|\neg\phi|\phi_1 \wedge \phi_2|B_i\phi|[\sigma]\phi$$
$$\sigma ::= Priv(i \rightarrow j, p)|Pub(i,p)|\sigma;\sigma'$$

where $\mathscr{C}$ is a collection of so-called transition rules.

The σ symbol denotes a (possibly composed) communication action. The action $Priv(i \rightarrow j, p)$ is a private or peer-to-peer message p from i to j; the action $Pub(i,p)$ means a public announcement or a broadcast by i about p. In the latter, every agent on the network learns p, whereas in the former, only j learns p. The bracket operator $[\sigma]\phi$ has the interpretation that after executing the communication action σ, ϕ holds.

The subscript in $\mathscr{L}_C$ refers to a set of so-called transition rules $\mathscr{C}$. The transition rules capture the updates, i.e., the expansions and side-effects, necessary for the interpretation of indirect effect of the constructs $Priv(i \rightarrow j, p)$ and $Pub(i,p)$. The transitions rules enforce consistency among the propositions that hold. For example, if an agent believes that the value of a message m is $[\![m]\!]$ and possesses a key k, then it must believe that the value of the encryption $\{m\}_k$ of m has a value that corresponds with $[\![m]\!]$.

A transition rule has the form $B_i p \Rightarrow \beta$. The condition $B_i p$ expresses that p must be believed by agent i. The body β of a transition rule is a sequence of actions $\alpha_1; \ldots ; \alpha_n$. Actions come in three flavours, viz. $L_{\mathscr{B}}p, S_{\mathscr{B},\mathscr{C}}p$ and $S^0_{\mathscr{B},\mathscr{C},\mathscr{D}}p$. Here, $L_{\mathscr{B}}p$ expresses that p is learned among the agents in the set $\mathscr{B}$ and corresponds to belief expansion, whereas $S_{\mathscr{B},\mathscr{C}}p$ expresses the side-effect that the agents in the set $\mathscr{B}$ have learned that the agents in the set $\mathscr{C}$ now know about p. Similarly, $S^0_{\mathscr{B},\mathscr{C},\mathscr{D}}p$ is used for the side-effect where agents in $\mathscr{B}$ assume that the agents in $\mathscr{D}$ have learned as well.

As an example, we will have the transition rule $B_b\{x\}_k \Rightarrow L_{ab}x$, when agents a and b share the key k, and a sends b the message $\{x\}_k$. In the situation described above, agent a sends the message x to agent b and agent b returns the message $\{x\}_k$. Since it is shared, a already can compute $\{x\}_k$ itself, so the delivery of $\{x\}_k$ does not teach a anything about this value. However, the transition rule expresses that a and b commonly learn, and, in particular, a learns that b knows the message x.

The semantics for the language $\mathscr{L}_C$, provided in the next definition, follows the set-up of, e.g., Baltag et al. (1998), and Clark and Jacab (2000). Definition 4.2 is organized in three layers. First, there is the layer of the actions of the language. The defining clauses make use of an auxiliary operation $\triangleright_p$. This operation helps in the processing of the relevant transition rules. The set $Mod(M, w, p)$ collects all the transition rules that will change the model. The next layer of the definition concerns the body of a transition rule. The last part of Definition 4.2 concerns the validity of the formulas of $\mathscr{L}_C$.

DEFINITION 4.2. Let $\mathscr{C}$ be a finite set of transition rules. For $\sigma \in \mathscr{L}_C$ the relation $[\![\sigma]\!]$ on models for $\mathscr{A}$ over $\mathscr{P}$ is given by

$$(M, w)[Priv(i \rightarrow j, p)](M', w')$$

$$\Leftrightarrow (M, w) \models B_i p \Rightarrow (\text{UPDATE}_{p,j}(M, w) \triangleright_p (M', w'))$$

$$(M, w)[Pub(i, p)](M', w')$$

$$\Leftrightarrow (M, w) \models B_i p \Rightarrow (\text{UPDATE}_{p,\mathscr{A}}(M, w) \triangleright_p (M', w'))$$

$$(M, w)[\sigma; \sigma'](M', w')$$

$$\Leftrightarrow (M, w)[\sigma](M'', w'')[\sigma'](M', w')$$

for some model (M'', w'')

$$
\begin{aligned}
&(M,w) \triangleright_p (M',w') \\
&\quad \Leftrightarrow \texttt{if}(x \Rightarrow \beta) \in Mod(M,w,p) \\
&\qquad \texttt{then } (M,w)\langle\beta\rangle(M'',w'') \triangleright_p (M',w') \\
&\qquad\quad \text{for some } (M'',w'') \\
&\qquad \texttt{else } (M,w) = (M',w') \texttt{ end}
\end{aligned}
$$

$$
\begin{aligned}
&(M,w)\langle\rangle(M',w') \\
&\quad \Leftrightarrow (M,w) = (M',w') \\
&(M,w)\langle L_{\mathscr{B}}p;\beta\rangle(M',w') \\
&\quad \Leftrightarrow \text{UPDATE}_{(p,\mathscr{B})}(M,w)\langle\beta\rangle(M',w') \\
&(M,w)\langle S_{\mathscr{B},\mathscr{C}}p;\beta\rangle(M',w') \\
&\quad \Leftrightarrow \text{SIDE-EFFECT}_{(p,\mathscr{B},\mathscr{C})}(M,w)\langle\beta\rangle(M',w') \\
&(M,w)\langle S^0_{\mathscr{B},\mathscr{C},\mathscr{D}}p;\beta\rangle(M',w') \\
&\quad \Leftrightarrow \text{0-side-effect}_{(p,\mathscr{X})}(M,w)\langle\beta\rangle \\
&\quad \text{where } \mathscr{X} = \{\mathscr{B},\mathscr{C},\mathscr{D},\mathscr{A} \setminus (\mathscr{B} \cup \mathscr{C} \cup \mathscr{D})\}
\end{aligned}
$$

$$
\begin{aligned}
&(M,w) \models p \\
&\quad \Leftrightarrow \pi(w)(p) = \texttt{true} \\
&(M,w) \models \neg\phi \\
&\quad \Leftrightarrow (M,w)\phi \\
&(M,w) \models \phi \wedge \psi \\
&\quad \Leftrightarrow (M,w) \models \phi \text{ and } (M,w) \models \psi \\
&(M,w) \models B_i\phi \\
&\quad \Leftrightarrow (M,v) \models \phi \text{ for all } v \text{ such that } R_i(w,v) \\
&(M,w) \models [\sigma]\phi \\
&\quad \Leftrightarrow (M',w') \models \phi \text{ if } (M,w)\llbracket\sigma\rrbracket(M',w')
\end{aligned}
$$

where

$$
\begin{aligned}
Mod(M,w,p) = \{B_ip \Rightarrow \beta \in \mathscr{C} | (M,w) \models B_ip,\\
(M,w)\langle\beta\rangle(M',w'), \exists\phi : (M',w')\\
\models \phi \ \not\leftrightarrow (M,w) \models \phi\}
\end{aligned}
$$

The private and public communication of p can only be executed under the condition B_ip. The communication has the effect that the agent j, respectively all agents get informed about p. Next, the transition rules are invoked as a consequence of some parties learning p, as expressed by the operator $\rhd_p$. The 'modifiers', the transition rules in the set $Mod(M, w, p)$ are those rules that match the learning of p and, moreover, will transform (M, w) into a different model, i.e., some formula ϕ will have changed its truth value. As a consequence of the algebraic properties of the update and side-effect operators of Section 3, the order of in which the transition rules are processed does not matter and every 'modifying' rule gets applied at most once (by idempotency). So, no rules are applied over and over again. Apart from this, the above definition also works for general formulas instead of objective ones (cf. Balta et al., 1998; van Ditmansch 2000).

5. EXAMPLES

In this section we discuss how the machinery developed above works out for a concrete example. Preparatory for this, in order to keep the models within reasonable size, we employ two helpful tricks. The first one is the disregarding of propositions not known to any agent. Thus, if a proposition is not part of the model, then the interpretation is that no agent has any knowledge about it. What we then have to specify is how to add a fresh proposition to the model. We accomplish this by making two copies of the original states. One of them we assign 'positive' and the other 'negative'. In the positive states, the proposition will be `true`, and in the negative states, the proposition will be `false`.

DEFINITION 5.1. Given a model $(M, w) = \langle S, \pi, R_1, \ldots, R_m \rangle$ and a fresh proposition p, we define the operation $\textsc{addatom}_p$ such that $(M', w') = \textsc{addatom}_p(M, w) = \langle S', \pi', R'_1, \ldots, R'_m \rangle$ where

- $S' = pos(S) \cup neg(S)$
- $\pi'(pos(s))(q) =$ if $p = q$ then `true` else $\pi(s)(q)$
- $\pi'(neg(s))(q) =$ if $p = q$ then `false` else $\pi(s)(q)$
- $R'_i(s, t) \Leftrightarrow R_i(s, t)$ for any agent i
- $w' = pos(w)$

We suppress straightforward technicalities regarding the restriction of the domain of the valuation π or expansion of the set of proposition $\mathscr{P}$.

We have the following property.

LEMMA 5.1. Given a model (M, w) and a fresh proposition p such that $(M', w') = \textsc{addatom}_p(M, w)$ it holds that

(a) $(M', w') \models p$;
(b) $(M, w) \models \phi \Leftrightarrow (M', w') \models \phi$ for $p \notin \phi^*$ with ϕ^* the closure under subformulas of ϕ;
(c) $(M', w') \not\models B_i p$ for all agents i.

Proof. (a) Trivial, since we make the new point the positive copy of the old point.

(b) We prove the stronger property $(M, s) \models \phi, p \notin \phi^* \Leftrightarrow (M', s')$ where s' is a copy of s. Proof by induction on complexity of ϕ. Suppose ϕ is objective then it's trivial (since $p \notin \phi^*$). Now suppose $\phi = B_i\psi$, then each state that is reachable in the resulting model is reachable if and only if it was reachable from a copy in the original model. By induction, we have the property for ψ, so it follows that we have it for ϕ.

(c) It holds that $w' = pos(w)$ and we have some $w'' = neg(w)$, such that $R_a(w', w'')$. □

The second trick helps to short-cut the application of rules which helps keeping the model in a reasonable size.

LEMMA 5.2. Given a model (M, w), an agent $i \in \mathscr{A}$, a set of agents $\mathscr{B} \subseteq \mathscr{A}$ it holds that the model (M', w') such that

$$(M, w)\langle L_i p; S_{\mathscr{B},i}\, p\rangle x \langle L_{\mathscr{A}} p\rangle (M', w')$$

for some model x, and the model (M'', w'') such that

$$(M, w)\langle L_{\mathscr{A}} p\rangle (M'', w'')$$

are bisimilar.

Proof. The proof is a corollary of the following two properties:

1. $L_i p; L_{\mathscr{A}} p = L_{\mathscr{A}} p$
2. $S_{\mathscr{B},i} p; L_{\mathscr{A}} p = L_{\mathscr{A}} p$

The proof of these properties is similar to the ones we have seen before. Here we proof (1). Because in both operations the last operation is a $L_{\mathscr{A}}$ we can easily that only *new* states are reachable. Bisimulation is constructed by associating links between states that are copies of each other in the original model. Checking bisimulation now is trivial. For (2) it is similar. Intuitively for the first operation, again in the last step, only the *new* are reachable. The removal of a in the knowledge of $\mathscr{B}$ in the first step is redundant because of the removal that happens in any case in the second step. □

That is to say, if an agent i learns p and then all other agents learn about i that it has learned p, followed by the action where everyone learns p (commonly), then it is equivalent to say that they have just learned p commonly.

5.1. *The SRA Three Pass protocol*

Shamir, Rivest and Adleman have suggested the three-pass protocol (Clark and Jacob, 1997) for the transmission of a message under the assumption of a commutative cipher. It is known to be insecure and various attacks have been suggested. However, it serves an illustrative purpose here. The protocol has the following steps:

1. $a \rightarrow b : \{x\}_{k_a}$
2. $b \rightarrow a : \{\{x\}_{k_a}\}_{k_b}$
3. $a \rightarrow b : \{x\}_{k_b}$

Here, both agent a and b have their own symmetric and unshared encryption key, k_a and k_b, respectively. Agent a wants to send message x to agent b through an insecure channel and therefore wants to send x encrypted to b. It does this by sending x protected with its own key. Next, b will encrypt this message with b's key and sends this back. Since the encryption is assumed to be commutative, a can now decrypt this message and sends the result to b. Finally, b can decrypt the message it has just received and learn the value of x.

In our modeling, we consider three agents $\{a, b, c\}$. It is assumed that all agents can see the activity of the network. In particular, they see messages been sent out and received. We are interested in what agent c can learn during a run of this protocol between agents a and b. We use the notation $m_{\mathscr{K}}$, for a possibly empty set of agents $\mathscr{K}$, to denote the message m encrypted with the keys of all agents in the set $\mathscr{K}$. We have, e.g., $m_{\{a,b\}} = \{\{m\}_{k_a}\}_{k_b}$. Since the cipher is commutative, this is well defined. Also, we write $S_{*,a}\phi$ instead of $S_{\mathscr{A},\{a\}}\phi$, to express that all agents learn that agent a knows about formula ϕ.

Next, we define the transition rules. The first transition rule models the fact that agents can encrypt with their own key:

$$B_j m_{\mathscr{K}} \Rightarrow L_j m_{\mathscr{K}+j}; S_{*,j} m_{\mathscr{K}+j}.$$

For simplicity, it is assumed that $j \notin \mathscr{K}$. Thus, if an agent j happens to learn the value of a message encrypted with the keys of the agents in the set $\mathscr{K}$, then agent j can encrypt the message received with its own key added (provided it was not used already). Moreover, as the other agents have seen that agent j has received the message, the agents collectively learn that agent j knows about the result after adding its key, as expressed by $S_{*,j} m_{\mathscr{K}+j}$.

Complementary, we have the transition rule

$$B_j m_{\mathscr{K}} \Rightarrow L_j m_{\mathscr{K}-j}; S_{*,j} m_{\mathscr{K}-j}$$

for every agent $j \in \mathscr{A}$. Now, it is assumed that the agent j is among the agents in $\mathscr{K}$. By commutativity of the cipher, agent j can then (partially) decrypt the message $m_{\mathscr{K}}$ and learn $m_{\mathscr{K}-j}$ whereas all others know that agent j can do this.

In the modeling, we limit ourselves by defining the list of useful propositions. The propositions we want to consider here are $\mathscr{P} = \{m, m_a, m_b, m_{ab}\}$ where m_a abbreviates $\{x\}_{k_a} = [\![\{x\}_{k_a}]\!]$ and m_{ab} abbreviates $\{\{x\}_{k_a}\}_{k_b} = [\![\{\{x\}_{k_a}\}_{k_b}]\!]$. Recall, that $[\![y]\!]$ denotes the *real* value of y, i.e., the value of the expression y in the point of the model.

Next, we must represent the initial knowledge of the agents, i.e., their knowledge before the run of the protocol. We will assume that a is the only agent that knows m and m_a. Furthermore, we will assume that the other agents know this about a. The corresponding Kripke structure is in Figure 12. The SRA protocol can be captured by three public announcement $Pub(a, m_a)$, $Pub(b, m_{ab})$ and $Pub(a, m_b)$. We are curious whether at the end of the protocol

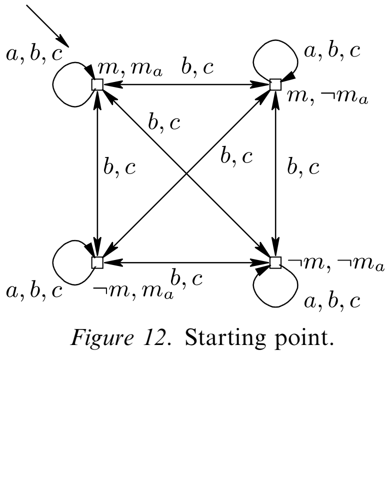

Figure 12. Starting point.

(i) agent b will know m;
(ii) agent a will know that agent b knows m;
(iii) agent c only knows that agents a and b know about m.

The first step is executed. That is, m_a is propagated on the network, so all agents will learn its value. So, we execute the action $Pub(a, m_a)$. If we discard the states that become unreachable, this results in the model of Figure 13. Note that in this model $B_b m_a$ holds. This is the condition of one of the transition rules, that is, it triggers $B_b m_a \Rightarrow L_b m_{ab}; S_{*,b} m_{ab}$ since its antecedent holds in the point now.

For processing the operation $L_b m_{ab}$, we notice that m_{ab} is not modeled yet, so this is the first thing to do. We will not repeat m_a in the figure since this holds in any state of the model. The operation ADDATOM$_{m_{ab}}$ results in the model of Figure 14. Instead of applying the body of the transition rule $L_b m_{ab}; S_{*,b} m_{ab}$ to this model, we observe that in the next step of the protocol $L_{\mathscr{A}} m_{ab}$ is executed, as the result of the action Pub(b, m_{ab}), since the message is being transmitted to all agents on the network. So, with an appeal to Lemma 5.2, it is justified to skip the operation that are required by the transition rules and perform $L_{\mathscr{A}} m_{ab}$ only. We arrive at the model in Figure 15.

In turn, this results in the triggering of the transition rule: $B_a m_{ab} \Rightarrow L_a m_b; S_{*,a} m_b$. This is in fact a completely similar case to the previous step of the protocol. Since the next action of the protocol is

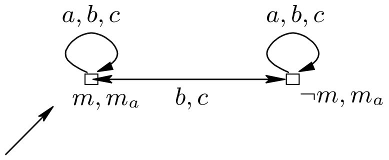

Figure 13. After $Pub(a, m_a)$.

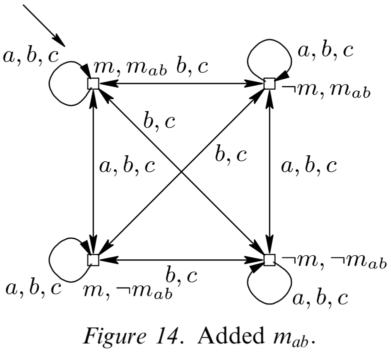

Figure 14. Added m_{ab}.

Pub(a, m_b) the value of m_b will be learned by all agents, anyway. Again, we dismiss the $\mathrm{m_{ab}}$ proposition since every agent has learned this.

We introduce the proposition m_b and execute Pub(a, m_b). So, we end up with the model in Figure 16. The last transition rule that is triggered is $B_b m_b \Rightarrow L_b m; S_{*,b} m$. Again, we discard the proposition that holds in every state, which is m_b, and focus on the most interesting proposition m. First b learns m, as dictated by the operation $L_b m$, which results in the model in Figure 17.

Next, the second action for the transition rule is that all learn $B_b m \vee B_b \neg m$. If we execute this, we get the model (M', w') which is depicted in Figure 18. Recall that this model (M', w') is obtained from the initial model (M, w) by application of the actions $Pub(a, m_a); Pub(b, m_{ab}); Pub(a, m_b)$ and associated transition rules.

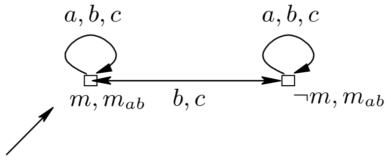

Figure 15. After $Pub(b, m_{ab})$.

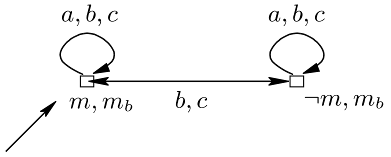

Figure 16. After $Pub(a, m_b)$.

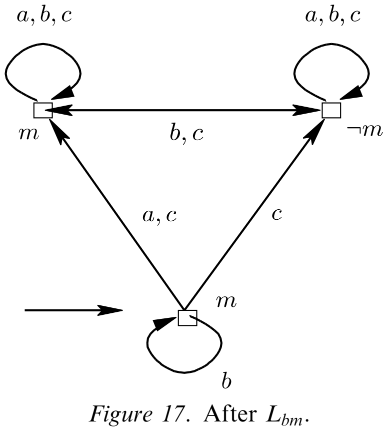

Figure 17. After L_{bm}.

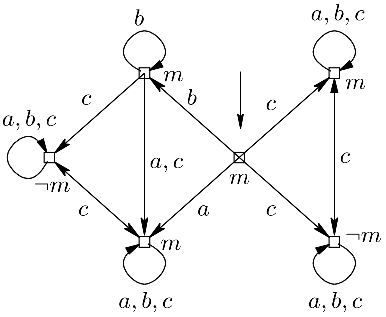

Figure 18. After $S_{*,b}m$.

Moreover, in the resulting model (M', w') it holds that (i) $B_b m$, (ii) $C_{ab} m$, and $\neg(B_c m \vee B_c \neg m)$.

5.2. *The Wide-Mouthed Frog protocol*

The next example we address to illustrate the update machinery developed above, is the well-known Wide-Mouthed Frog protocol (see, e.g., Borrows 1990; Abadi and Gordon 1999). The protocol exchanges a session key k from the agent a to another agent b via a server s. Then, agent a sends agent b a message protected with the session key k. It is assumed, that the agents a and b share each a symmetric key, k_{as} and k_{bs} say, with the server. The protocol can be described by

1. $a \to s : \{k\}_{k_{as}}$
2. $s \to b : \{k\}_{k_{bs}}$
3. $a \to b : \{m\}_k$

The keys k_{as} and k_{bs} are shared among a and s, and among b and s, respectively. The key k is fresh and initially only known to agent a, as is message m.

In the analysis we want to focus on the session key k and the message m it protects. Therefore the protocol is represented by the sequence of actions

$$Priv(a \to s, k); Priv(s \to b, k); Pub(a, \{m\}_k).$$

Thus, the security of the channel, based on the server keys k_{as} and k_{bs} is expressed by private rather than public communication. We assume that the 'ports' of the channel from a to b can be observed, but the ones for the communication with the server are not visible to other parties.

The initial knowledge is depicted in Figure 19. As transition rule we adopt

$$B_b\{m\}_k \Rightarrow L_b m; S_{*,b} m; S_{*,b} k$$

i.e., after b has received the encrypted message $\{m\}_k$ it can learn its content m and everybody learns that b knows about it. Moreover, if agent b is known to know about the message m, then it must know about the session key k as well. Note that, there are no transition rules dealing with the communication with the server.

Execution of the first action Priv$(a \to s, k)$ leads to an update of the knowledge of s. This is represented by a model with six (reachable) states in Figure 20. Similarly, but more complicated, the execution of the second action Priv$(s \to b, k)$ induces the model in Figure 21. The model gets more involved because, by assumption, the

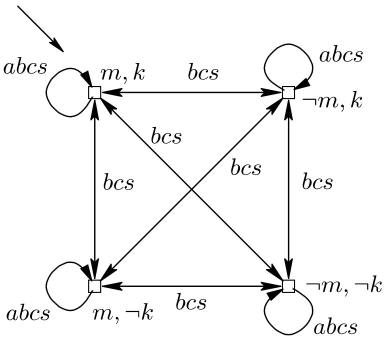

Figure 19. Starting point.

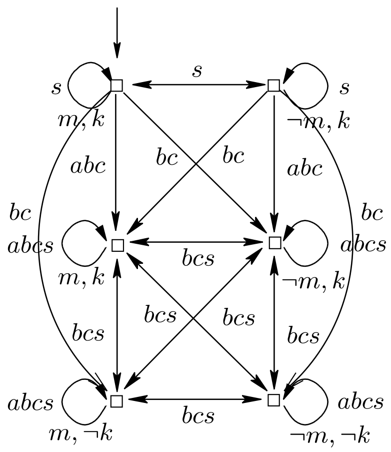

Figure 20. After *Priv*$(a \to s, k)$.

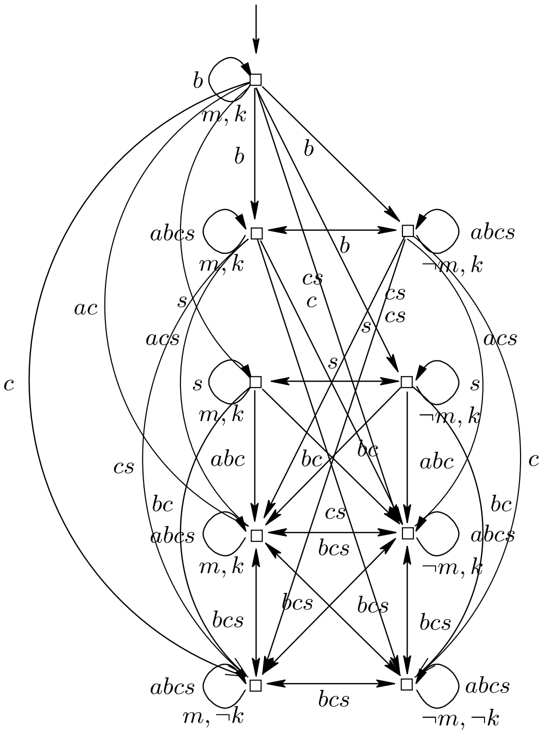

Figure 21. After *Priv*$(s \to b, k)$.

learning of messages exchanged with the server is private. For example, agent a is not aware of agent b learning about the key k. So far, no transition rules have been triggered.

Next, we execute the last step of the protocol, *viz*. the public communication $Pub(a, \{m\}_k)$. For this we need to add the atom m_k abbreviating $\{m\}_k = [\![\{m\}_k]\!]$. This doubles the number of states of the models. However, since m_k will be known to all agents, its negative part can be discarded. Now, the transition rules gets activated. So, agent b learns the content m and the other agents learn that b knows about m and k, resulting in the final model in Figure 22. Since agent b is learning twice in a row, the difference between Figures 21 and 22 are the absence of b-arrows to $\neg m$-states and between states with different values for about m and k.

Typical properties of this model include $B_b(m \wedge k)$, agent b knows the values of the message m and session key k, $\neg C_{ab}m$, m is not commonly known by agents a and b, and, $\neg B_c B_s k$ agent c does not know that the server knows the session key k. That agents a and b do not share

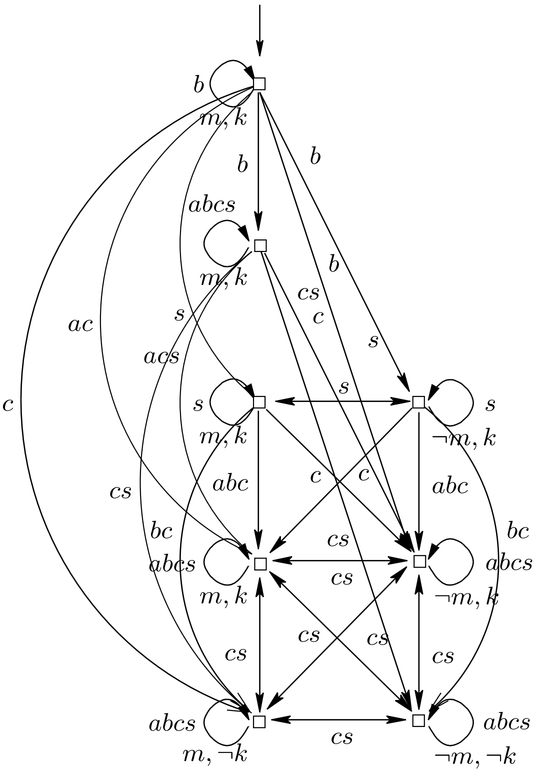

Figure 22. After *Pub*$(a, \{m\}_k)$.

the knowledge about the session key, is debatable. One way out is to modify the transition rules and have the operation $L_{ab}m$ instead of L_b.

6. CONCLUSION

Inspired by recent work on dynamic epistemic logics, we have proposed a logical language for describing (properties of) runs of security protocols. The language contains constructs for the three basic types of epistemic actions that happen during such runs. The semantics of the language is based on traditional Kripke models representing the epistemic state of the agents. Changes in the epistemic state of the agent system as a result of the execution of a protocol are described by means of transition rules that precisely indicate what belief updates happen under certain preconditions. These belief updates give rise to modifications of the models representing the agents' epistemic state in a way that is precisely given by semantic operations on these models. We have illustrated our approach for two well-known security protocols, *viz*. the SRA Three Pass protocol and the Wide-Mouthed Frog protocol.

The semantic updates we used, operate on traditional Kripke models as opposed to updates in the approaches of Gerbrandy and Baltag et al. We believe that this will make it less troublesome to integrate these updates into existing model checkers, which hopefully will lead to better and new tools for verifying properties of security protocols. However, for the development of the theory, it is important to establish the precise connection of the explicit approach followed here and the approach based on action models as advocated in Baltag et al. (1998), van Ditmarsch (2000), Baltag (2002), and Baltag and Moss (2004) A first step into this direction has been presented here, but many others will have to follow. Nevertheless, it points to a promising opportunity to establish a firm relationship between logical theory and security protocol analysis, to the benefit of the latter.

Although future research will have to justify this, we are confident that our method, preferably with some form of computer assistance, can be employed for a broad class of verification problems concerning security protocols because of the flexibility of our approach using transition rules for epistemic updates.

ACKNOWLEDGEMENTS

We are grateful to the anonymous reviewers for their constructive comments and to Simona Orzan for various discussions on the subject. We are also indebted to Peter Lucas for his support to this research.

REFERENCES

Abadi, M. and A. Gordon: 1999, ‘A calculus for cryptographic protocols: The spi calculus’, *Information and Computation* **148**, 1–70.

Abadi, M. and M. Tuttle: 1991, ‘A semantics for a logic of authentication’, in *Proc. PODC’91*, ACM, pp. 201–216.

Agray, N., W. van der Hoek, and E. P. de Vink: 2001, ‘On BAN logics for industrial security protocols’, in B. Dunin-Keplicz and E. Nawarecki, (eds.), *From Theory to Practice in Multi-Agent Systems*, LNAI 2296, pp. 29–38.

Anderson, R. J.: 2001, *Security Engineering: A Guide to Building Dependable Distributed Systems*, Wiley.

Baltag, A.: 2002, 'A logic for suspicous players: Epistemic actions and belief-updates in games', *Bulletin of Economic Research* **54**, 1–46.

Baltag, A.: and L. S. Moss: 2004, 'Logics for epistemic programs', *Synthese: Knowledge, Rationality and Action* **139**, 165–224.

Baltag, A., L. S. Moss, and S. Solecki: 1998, 'The logic of public announcements, common knowledge and private suspicions', in Itzhak Gilboa, (ed.), *Proc. TARK'98*, pp. 43–56.

Bleeker, A. and L. Meertens: 1997, 'A semantics for BAN logic', in *Proceedings DIMACS Workshop on Design and Formal Verification of Protocols*, DIMACS, Rutgers University, http://dimacs.rutgers.edu/Workshops/Security.

Burrows, M., M. Abadi, and R. M. Needham: 1990, 'A logic of authentication', *ACM Transactions on Computer Systems* **8**, 16–36.

Clark, J. A. and J. L Jacob: 1997, 'A survey of authentication protocols 1.0', *Technical Report*, University of York.

Dolev, D. and A. C. Yao: 1983, 'On the security of public-key protocols', *IEEE Transaction on Information Theory* **29**, 198–208.

Gerbrandy, J.: 1997, 'Dynamic epistemic logic', *Technical Report LP- 97–04*, ILLC.

Gerbrandy, J.: 1999, 'Bisimulations on Planet Kripke', PhD thesis, ILLC Dissertation Series 1999–01, University of Amsterdam.

Hommersom, A. J.: 2003, 'Reasoning about security', Master's thesis, Universiteit Utrecht.

Kessler, V. and H. Neumann: 1998, 'A sound logic for analyzing electronic commerce protocols', in J. -J. Quisquater, Y. Deswarte, C. Meadows, and D. Gollman (eds.), *Proc. ESORICS'98*, LNCS 1485, pp. 345–360.

Kooi, B.: 2003, 'Knowledge, Chance, and Change', PhD thesis, ILLC Dissertation Series 2003–01, University of Groningen.

Lowe, G.: 1996, 'Breaking and fixing the Needham-Schroeder public-key protocol using FDR', *Software - Concepts and Tools* **17**, 93–102.

Roorda, J.-W., W. van der Hoek, and J.-J. Ch Meyer: 2002, 'Iterated belief change in multi-agent systems', in *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems: Part 2.*

Schneier, B.: 2000, *Secrets and Lies: Digital Security in a Networked World*, Wiley.

Stubblebine, S. G. and R. N. Wright: 2002, 'An authentication logic with formal semantics supporting synchronization, revocation and recency', *IEEE Transactions on Software Engineering* **28**, 256–285.

van Ditmarsch, H. P.: 2000, 'Knowledge games', PhD thesis, ILLC Dissertation Series 2000–06, University of Groningen.

van Ditmarsch, H. P.: 2001, 'The semantics of concurrent knowledge actions', in M. Pauly and G. Sandu, (eds.), *Proc. ESSLLI Workshop on Logic and Games*, Helsinki.

Wedel, G. and V. Kessler: 1996, 'Formal semantics for authentication logics', in E. Bertino, H. Kurth, G. Martello, and E. Montolivo, (eds.), *Proc. ESORICS'96,* LNCS 1146, pp. 219–241.

Arjen Hommersom
Nijmegen Institute for Computing and Information Sciences
University of Nijmegen
Nijmegen, The Netherlands
E-mail: arjenh@cs.kun.nl

John-Jules Meyer
Institute of Information and Computing Sciences
University of Utrecht
Utrecht, The Netherlands

Erik de Vink
Department of Mathematics and Computer Science
Technische Universiteit Eindhoven
Leiden Institute of Advanced Computer Science
Leiden University
Leiden, The Netherlands