



**HAL**  
open science

## Generalized DEL-sequents

Guillaume Aucher, Bastien Maubert, François Schwarzentruber

► **To cite this version:**

Guillaume Aucher, Bastien Maubert, François Schwarzentruber. Generalized DEL-sequents. [Research Report] RR-8012, INRIA. 2012, pp.23. hal-00716074

**HAL Id: hal-00716074**

**<https://inria.hal.science/hal-00716074v1>**

Submitted on 9 Jul 2012

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



# Generalized DEL-sequents

Guillaume Aucher, Bastien Maubert, François Schwarzentruber

**RESEARCH  
REPORT**

**N° 8012**

July 2012

Project-Team Distribcom and S4





## Generalized DEL-sequents

Guillaume Aucher<sup>\*</sup>, Bastien Maubert<sup>†</sup>, François  
Schwarzentruber<sup>‡</sup>

Project-Team Distribcom and S4

Research Report n° 8012 — July 2012 — 20 pages

**Abstract:** Let us consider a sequence of formulas providing partial information about an initial situation, about a set of events occurring sequentially in this situation, and about the resulting situation after the occurrence of each event. From this whole sequence, we want to infer more information, either about the initial situation, or about one of the events, or about the resulting situation after one of the events. Within the framework of Dynamic Epistemic Logic (DEL), we show that these different kinds of problems are all reducible to the problem of inferring what holds in the final situation after the occurrence of all the events. We then provide a tableau method deciding whether this kind of inference is valid. We implement it in LotrecScheme and show that these inference problems are NEXPTIME-complete. We extend our results to the cases where the accessibility relation is serial and reflexive and illustrate them with the coordinated attack problem.

**Key-words:** Dynamic epistemic logic, tableau method, computational complexity

---

<sup>\*</sup> Université de Rennes 1 - INRIA, France

<sup>†</sup> Université de Rennes 1, France

<sup>‡</sup> ENS Cachan, France

**RESEARCH CENTRE  
RENNES – BRETAGNE ATLANTIQUE**

Campus universitaire de Beaulieu  
35042 Rennes Cedex

## DEL-séquents généralisés

**Résumé :** Considérons une séquence de formules fournissant une information partielle à propos d'une situation initiale, d'un ensemble d'évènements se produisant séquentiellement dans cette situation, et de la situation résultante après l'occurrence de chaque évènement. De cette séquence, nous voulons déduire plus d'information, soit sur la situation initiale, soit sur l'un des évènements, soit sur la situation résultante après l'un des évènements. Dans le cadre de la logique épistémique dynamique (DEL), nous montrons que ces différentes sortes de problèmes sont tous réductibles au problème de déduire des propriétés de la situation finale après l'occurrence de tous les évènements. Nous fournissons ensuite une méthode tableau qui décide la validité de ce type d'inférence. Nous l'implémentons en LotrecScheme et prouvons que ces problèmes d'inférence sont NEXPTIME-complets. Nous étendons nos résultats aux cas où la relation d'accessibilité est sériale ou réflexive et les illustrons par le problème de l'attaque coordonnée.

**Mots-clés :** Logique épistémique dynamique, méthode tableau, complexité calculatoire

## 1 Introduction

Assume that a sequence of  $n$  events has occurred in a situation. We have some information about each event and about the resulting situation after the occurrence of each event, in the form of a sequence of formulas  $\varphi'_i$  and  $\varphi_i$  respectively:

$$\varphi_0, \varphi'_1, \varphi_1, \dots, \varphi'_i, \varphi_i, \dots, \varphi'_n, \varphi_n$$

Our aim is to infer some more information about one of the events or about one of the resulting situations from the rest of information provided by this sequence. This defines respectively two different kinds of inference problems.

In the first kind of inference problem, we want to infer more information about the  $i^{th}$  resulting situation. That is, given a formula  $\psi$  describing (incompletely) a situation, we want to verify whether or not we can infer that this property  $\psi$  necessarily held at the  $i^{th}$  resulting situation:

$$\varphi_0, \varphi'_1, \varphi_1, \dots, \varphi'_i, \varphi_i, \dots, \varphi'_n, \varphi_n \stackrel{!}{\vdash}_i \psi ?$$

In the second kind of inference problem, we want to infer more information about the  $i^{th}$  event. That is, given a formula  $\psi'$  describing (incompletely) this event, we want to verify whether or not we can infer that this property  $\psi'$  necessarily held during the occurrence of the  $i^{th}$  event:

$$\varphi_0, \varphi'_1, \varphi_1, \dots, \varphi'_i, \varphi_i, \dots, \varphi'_n, \varphi_n \stackrel{!}{\vdash}_i \psi' ?$$

Solving these problems is relevant for *dynamic diagnosis* for instance (see [10] for an early survey of dynamic diagnosis). In this field, one is interested in looking for and verifying that plausible diagnoses of a faulty system ‘fit’ a given history that contains some abnormal behaviours. A diagnosis is a series of faults together with the time when they occur. Within our setting, this verification problem amounts to deciding about the satisfiability of a sequence of formulas.

We moreover assume that our situations involve several agents and we are interested in reasoning about the beliefs and knowledge of these agents. Our formulas  $\varphi'_i$  and  $\varphi_i$  will therefore express beliefs of several agents about events and about the resulting situations. For these reasons, we address our two problems above within the framework of Dynamic Epistemic Logic (DEL for short), since this logical framework is very well suited to express and reason about the beliefs of several agents in a dynamic setting.

The paper is organised as follows. In Section 2, we recall the core of dynamic epistemic logic and define our two kinds of inference problems. We show that these two kinds of inference problems are actually both reducible to the problem of inferring what holds in the final situation after the occurrence of all the events from the rest of the sequence. In Section 3, we provide a terminating, sound and complete tableau method that decides whether this kind of inference is valid. In Section 4, we show that our tableau method is optimal, first by proving that it is in NEXPTIME, and then by proving that our inference problem is NEXPTIME-hard. We also provide in this section a link to an implementation of our tableau method in LotrecScheme together with some details about its implementation. In Section 5, we extend our results to richer semantics where the accessibility relations are reflexive and serial. In Section 6, we apply our generalized DEL-sequents to the coordinated attack problem of the distributed system literature. Finally, in Section 7, we discuss some related work, and then conclude.

## 2 Dynamic Epistemic Logic: DEL-sequents

Following the methodology of DEL, we split the exposition of the logical framework into three subsections. We then define our generalized DEL-sequents in Section 2.4.

## 2.1 Representation of the initial situation: $\mathcal{L}$ -model

In the rest of this paper,  $\Phi$  is a countably infinite set of propositional letters called *atomic facts* which describe static situations, and  $Agt$  is a finite set of agents.

An  $\mathcal{L}$ -model is a tuple  $\mathcal{M} = (W, R, V)$  where:

- $W$  is a non-empty set of possible worlds,
- $R : Agt \rightarrow 2^{W \times W}$  is a function assigning to each agent  $a \in Agt$  an accessibility relation on  $W$ ,
- $V : \Phi \rightarrow 2^W$  is a function assigning to each propositional letter of  $\Phi$  a subset of  $W$ . The function  $V$  is called a valuation.

We write  $w \in \mathcal{M}$  for  $w \in W$ , and  $(\mathcal{M}, w)$  is called a pointed  $\mathcal{L}$ -model ( $w$  often represents the actual world). If  $w, v \in W$ , we write  $wR_a v$  for  $R(a)(w, v)$  and  $R_a(w) = \{v \in W \mid wR_a v\}$ . Intuitively,  $wR_a v$  means that in world  $w$  agent  $a$  considers that world  $v$  might be the actual world.

Then, we define the following epistemic language  $\mathcal{L}$  that can be used to describe and state properties of  $\mathcal{L}$ -models:

$$\mathcal{L} : \varphi ::= p \mid \neg\varphi \mid \varphi \wedge \psi \mid B_a\varphi$$

where  $p$  ranges over  $\Phi$  and  $a$  over  $Agt$ . We define  $\varphi \vee \psi \stackrel{\text{def}}{=} \neg(\neg\varphi \wedge \neg\psi)$  and  $\langle B_a \rangle \varphi \stackrel{\text{def}}{=} \neg B_a \neg\varphi$ . The symbol  $\top$  is an abbreviation for  $p \vee \neg p$  for a chosen  $p \in \Phi$ . Let  $\mathcal{M}$  be an  $\mathcal{L}$ -model,  $w \in \mathcal{M}$  and  $\varphi \in \mathcal{L}$ .  $\mathcal{M}, w \models \varphi$  is defined inductively as follows:

$$\begin{aligned} \mathcal{M}, w \models p & \quad \text{iff } w \in V(p) \\ \mathcal{M}, w \models \neg\varphi & \quad \text{iff not } \mathcal{M}, w \models \varphi \\ \mathcal{M}, w \models \varphi \wedge \psi & \quad \text{iff } \mathcal{M}, w \models \varphi \text{ and } \mathcal{M}, w \models \psi \\ \mathcal{M}, w \models B_a\varphi & \quad \text{iff for all } v \in R_a(w), \mathcal{M}, v \models \varphi \end{aligned}$$

The formula  $B_a\varphi$  reads as ‘‘agent  $a$  believes  $\varphi$ ’’. Its truth conditions are defined in such a way that agent  $a$  believes  $\varphi$  holds in a possible world when  $\varphi$  holds in all the worlds agent  $a$  considers possible. Dually, the formula  $\langle B_a \rangle \varphi$  reads as ‘‘agent  $a$  considers  $\varphi$  is plausible’’. Agent  $a$  considers that  $\varphi$  is plausible in a possible world when  $\varphi$  holds in at least one of the worlds agent  $a$  considers possible.

## 2.2 Representation of the event: $\mathcal{L}'$ -model

The propositional letters  $p'$  describing events are called *atomic events* and range over an infinite set  $\Phi'$ . To each atomic event  $p'$ , we assign a formula of the language  $\mathcal{L}$ , which is called the *precondition* of  $p'$ . This precondition corresponds to the property that should be true in any world  $w$  of an  $\mathcal{L}$ -model so that the atomic event  $p'$  can ‘physically’ occur in this world  $w$ . To do so we define a surjection  $\text{Pre} : \Phi' \rightarrow \mathcal{L}$  that is called the *precondition function*. We take it surjective so that we have an atomic event for every possible precondition.

An  $\mathcal{L}'$ -model is a tuple  $\mathcal{M}' = (W', R', V')$  where:

- $W'$  is a non-empty set of possible events,
- $R' : Agt \rightarrow 2^{W' \times W'}$  is a function assigning to each agent  $a \in Agt$  an accessibility relation on  $W'$ ,

- $V' : \Phi' \rightarrow 2^{W'}$  is a function assigning to each propositional letter of  $\Phi'$  a subset of  $W'$  such that for all  $w' \in W'$ , there is at most one  $p'$  such that  $w' \in V'(p')$  (Exclusivity).

We write  $w' \in \mathcal{M}'$  for  $w' \in W'$ , and  $(\mathcal{M}', w')$  is called a *pointed  $\mathcal{L}'$ -model* and  $w'$  represents the actual event of  $(\mathcal{M}', w')$ . If  $w', u' \in W'$ , we write  $w' R'_a u'$  for  $R'(a)(w', u')$  and  $R'_a(w') = \{u' \in W' \mid w' R'_a u'\}$ . Intuitively,  $u' \in R'_a(w')$  means that while the possible event represented by  $w'$  is occurring, agent  $a$  considers possible that the possible event represented by  $u'$  is actually occurring. Our definition of an  $\mathcal{L}'$ -model is equivalent to the definition of an action signature in the logical framework of [6].<sup>1</sup>

Just as we defined a language  $\mathcal{L}$  for  $\mathcal{L}$ -models, we also define a language  $\mathcal{L}'$  for  $\mathcal{L}'$ -models ( $\mathcal{L}'$  was already introduced in [8]):

$$\mathcal{L}' : \varphi' ::= p' \mid \neg\varphi' \mid \varphi' \wedge \psi' \mid B_a\varphi'$$

where  $p'$  ranges over  $\Phi'$  and  $a$  over  $\text{Agt}$ . In the sequel, formulas of  $\mathcal{L}'$  are always indexed by the quotation mark ', unlike formulas of  $\mathcal{L}$ . The truth conditions of the language  $\mathcal{L}'$  are identical to the ones of the language  $\mathcal{L}$ . Let  $\mathcal{M}'$  be an  $\mathcal{L}'$ -model,  $w' \in \mathcal{M}'$  and  $\varphi' \in \mathcal{L}'$ .  $\mathcal{M}', w' \models \varphi'$  is defined inductively as follows:

$$\begin{aligned} \mathcal{M}', w' \models p' & \quad \text{iff } w' \in V'(p') \\ \mathcal{M}', w' \models \neg\varphi' & \quad \text{iff not } \mathcal{M}', w' \models \varphi' \\ \mathcal{M}', w' \models \varphi' \wedge \psi' & \quad \text{iff } \mathcal{M}', w' \models \varphi' \text{ and } \mathcal{M}', w' \models \psi' \\ \mathcal{M}', w' \models B_a\varphi' & \quad \text{iff for all } u' \in R'_a(w'), \mathcal{M}', u' \models \varphi' \end{aligned}$$

### 2.3 Update of the initial situation by the event: product update

The precondition function  $\text{Pre}$  of the previous section induces a precondition function for  $\mathcal{L}'$ -models, which assigns to each possible event  $w'$  of an  $\mathcal{L}'$ -model a formula  $\text{Pre}(w')$  of  $\mathcal{L}$ . The precondition function induced by the  $\mathcal{L}'$ -model  $\mathcal{M}' = (W', R', V')$  is defined as follows:  $\text{Pre}(w') = \text{Pre}(p')$  if there is  $p'$  such that  $\mathcal{M}', w' \models p'$ ;  $\text{Pre}(w') = \top$  otherwise.

We then redefine equivalently in our setting the BMS product update of [7] as follows. Let  $(\mathcal{M}, w) = (W, R, V, w)$  be a pointed  $\mathcal{L}$ -model and let  $(\mathcal{M}', w') = (W', R', V', w')$  be a pointed  $\mathcal{L}'$ -model such that  $\mathcal{M}, w \models \text{Pre}(w')$ . The *product update of  $(\mathcal{M}, w)$  and  $(\mathcal{M}', w')$*  is the pointed  $\mathcal{L}$ -model  $(\mathcal{M}, w) \otimes (\mathcal{M}', w') = (W^\otimes, R^\otimes, V^\otimes, (w, w'))$  defined as follows:

$$\begin{aligned} W^\otimes &= \{(u, u') \in W \times W' \mid \mathcal{M}, u \models \text{Pre}(u')\}, \\ R_a^\otimes(u, u') &= \{(v, v') \in W^\otimes \mid v \in R_a(u) \text{ and } v' \in R'_a(u')\}, \\ V^\otimes(p) &= \{(u, u') \in W^\otimes \mid \mathcal{M}, u \models p\}. \end{aligned}$$

This product update yields a new  $\mathcal{L}$ -model  $(\mathcal{M}, w) \otimes (\mathcal{M}', w')$  representing how the new situation which was previously represented by  $(\mathcal{M}, w)$  is perceived by the agents after the occurrence of the event represented by  $(\mathcal{M}', w')$ .

### 2.4 Generalized DEL-sequents

In this section we define two different inference relations on formulas of  $\mathcal{L}$  and  $\mathcal{L}'$  representing an initial situation, a series of events and resulting epistemic situations. One relation enables to infer information about one of the epistemic situations, the other about one of the events:

<sup>1</sup>Let  $\Sigma = (W', R', (w'_1, \dots, w'_n))$  be an action signature and let  $\varphi_1, \dots, \varphi_n \in \mathcal{L}$ . The  $\mathcal{L}'$ -model associated to  $(\Sigma, \varphi_1, \dots, \varphi_n)$  is the tuple  $M' = (W', R', V')$  where the valuation  $V'$  is defined as follows. We pick  $q' \in \Phi'$  such that  $\text{Pre}(q') = \top$ , and for all  $i \in \{1, \dots, n\}$ , we pick  $p'_i \in \Phi'$  such that  $\text{Pre}(p'_i) = \varphi_i$ . Then, for all  $i \in \{1, \dots, n\}$  we set  $V'(p'_i) = \{w'_i\}$ , we also set  $V'(q') = W' - \{w'_1, \dots, w'_n\}$ , and for all  $p' \in \Phi' - \{q', p'_1, \dots, p'_n\}$  we set  $V'(p') = \emptyset$ .



**Definition 1** Let  $\varphi_0, \varphi_1, \dots, \varphi_n, \psi \in \mathcal{L}$ ,  $\varphi'_1, \dots, \varphi'_n, \psi' \in \mathcal{L}'$ . We define the logical consequence relations  $\varphi_0, \varphi'_1, \varphi_1, \dots, \varphi'_n, \varphi_n \stackrel{1}{\vdash}_k \psi$  where  $0 \leq k \leq n$ , and  $\varphi_0, \varphi'_1, \varphi_1, \dots, \varphi'_n, \varphi_n \stackrel{2}{\vdash}_k \psi'$  where  $1 \leq k \leq n$ . In the following  $i$  ranges over  $\{0, \dots, n\}$  and  $j$  ranges over  $\{1, \dots, n\}$ .

$$\varphi_0, \varphi'_1, \varphi_1, \dots, \varphi'_n, \varphi_n \stackrel{1}{\vdash}_k \psi \text{ if}$$

for all pointed  $\mathcal{L}$ -models  $(\mathcal{M}_0, w_0)$  and  $\mathcal{L}'$ -models  $(\mathcal{M}'_j, w'_j)$ , if we have for all  $i$  that  $(\mathcal{M}_i, w_i) = (\mathcal{M}_0, w_0) \otimes (\mathcal{M}'_1, w'_1) \otimes \dots \otimes (\mathcal{M}'_i, w'_i)$  is defined<sup>2</sup> and  $\mathcal{M}_i, w_i \models \varphi_i$ , and for all  $j$  that  $\mathcal{M}'_j, w'_j \models \varphi'_j$ , then  $\mathcal{M}_k, w_k \models \psi$ .

$$\varphi_0, \varphi'_1, \varphi_1, \dots, \varphi'_n, \varphi_n \stackrel{2}{\vdash}_k \psi' \text{ if}$$

for all pointed  $\mathcal{L}$ -models  $(\mathcal{M}_0, w_0)$  and  $\mathcal{L}'$ -models  $(\mathcal{M}'_j, w'_j)$ , if we have for all  $i$  that  $(\mathcal{M}_i, w_i) = (\mathcal{M}_0, w_0) \otimes (\mathcal{M}'_1, w'_1) \otimes \dots \otimes (\mathcal{M}'_i, w'_i)$  is defined and  $\mathcal{M}_i, w_i \models \varphi_i$ , and for all  $j$  that  $\mathcal{M}'_j, w'_j \models \varphi'_j$ , then  $\mathcal{M}'_k, w'_k \models \psi'$ .

In fact, those two DEL-sequent relations are interdefinable:

**Proposition 1** For all  $\varphi_0, \dots, \varphi_n \in \mathcal{L}$ ,  $\varphi'_1, \dots, \varphi'_n, \psi' \in \mathcal{L}'$  and  $k \in \{1, \dots, n\}$ ,

$$\varphi_0, \varphi'_1, \varphi_1, \dots, \varphi'_n, \varphi_n \stackrel{2}{\vdash}_k \psi' \text{ iff } \varphi_0, \dots, \varphi'_k \wedge \neg\psi', \top, \varphi'_{k+1}, \dots, \varphi_n \stackrel{1}{\vdash}_k \neg\varphi$$

**Proof**  $i$  ranges over  $\{0, \dots, n\}$  and  $j$  ranges over  $\{1, \dots, n\}$ .

Suppose that  $\varphi_0, \varphi'_1, \varphi_1, \dots, \varphi'_n, \varphi_n \stackrel{2}{\vdash}_k \psi'$  holds and let  $(\mathcal{M}_0, w_0)$  be a pointed  $\mathcal{L}$ -model and  $(\mathcal{M}'_j, w'_j)$  be  $n$  pointed  $\mathcal{L}'$ -models. Assume that for all  $i$ ,  $\mathcal{M}_i, w_i$  is defined and for all  $i \neq k$ ,  $\mathcal{M}_i, w_i \models \varphi_i$ , for all  $j \neq k$ ,  $\mathcal{M}'_j, w'_j \models \varphi'_j$ , and  $\mathcal{M}'_k, w'_k \models \varphi'_k \wedge \neg\psi'$ . Assume towards a contradiction that  $\mathcal{M}_k, w_k \models \varphi_k$ . Then, since  $\varphi_0, \varphi'_1, \varphi_1, \dots, \varphi'_n, \varphi_n \stackrel{2}{\vdash}_k \psi'$  holds we have that  $\mathcal{M}'_k, w'_k \models \psi'$ . This is a contradiction, so  $\mathcal{M}_k, w_k \models \varphi_k$ .

Now suppose that  $\varphi_0, \dots, \varphi'_k \wedge \neg\psi', \top, \varphi'_{k+1}, \dots, \varphi_n \stackrel{1}{\vdash}_k \neg\varphi_k$  holds and let  $(\mathcal{M}_0, w_0)$  be a pointed  $\mathcal{L}$ -model and  $(\mathcal{M}'_j, w'_j)$  be  $n$  pointed  $\mathcal{L}'$ -models. Assume that for all  $i$ ,  $\mathcal{M}_i, w_i$  is defined and  $\mathcal{M}_i, w_i \models \varphi_i$ , and assume that for all  $j$ ,  $\mathcal{M}'_j, w'_j \models \varphi'_j$ . Assume towards a contradiction that  $\mathcal{M}'_k, w'_k \not\models \psi'$ . Then  $\mathcal{M}'_k, w'_k \models \neg\psi'$ , and because  $\varphi_0, \dots, \varphi'_k \wedge \neg\psi', \top, \varphi'_{k+1}, \dots, \varphi_n \stackrel{1}{\vdash}_k \neg\varphi_k$  holds we have that  $\mathcal{M}_k, w_k \models \neg\varphi_k$ , which is a contradiction.  $\square$

Moreover, the first relation can always be reduced to the case where information is inferred about the last situation:

**Proposition 2** For all  $\varphi_0, \dots, \varphi_n \in \mathcal{L}$ ,  $\varphi'_1, \dots, \varphi'_n \in \mathcal{L}'$ ,  $k < n$  and  $\psi \in \mathcal{L}$ ,

$$\varphi_0, \varphi'_1, \varphi_1, \dots, \varphi'_n, \varphi_n \stackrel{1}{\vdash}_k \psi \text{ iff } \varphi_0, \dots, \varphi_k \wedge \neg\psi, \dots, \varphi'_n, \top \stackrel{1}{\vdash}_n \neg\varphi_n$$

Considering Propositions 1 and 2, in the rest of the paper we will only provide a tableau method for the DEL-sequent  $\varphi_0, \varphi'_1, \varphi_1, \dots, \varphi'_n, \varphi_n \stackrel{1}{\vdash}_n \psi$ .

In [3], we defined three kinds of logical consequence relations dealing with a single event, which are special cases of the general relations defined here. Let  $\varphi_0, \varphi_1 \in \mathcal{L}$  and  $\varphi' \in \mathcal{L}'$ . It holds that:

$$\begin{aligned} \varphi_0, \varphi' \stackrel{1}{\vdash} \varphi_1 &\text{ iff } \varphi_0, \varphi', \top \stackrel{1}{\vdash}_1 \varphi_1 & \varphi', \varphi_1 \stackrel{3}{\vdash} \varphi_0 &\text{ iff } \top, \varphi', \varphi_1 \stackrel{1}{\vdash}_0 \varphi_0 \\ \varphi_0, \varphi_1 \stackrel{2}{\vdash} \varphi' &\text{ iff } \varphi_0, \top, \varphi_1 \stackrel{2}{\vdash} \varphi' \end{aligned}$$

<sup>2</sup>When  $i = 0$  we let  $(\mathcal{M}_0, w_0) \otimes (\mathcal{M}'_1, w'_1) \otimes \dots \otimes (\mathcal{M}'_i, w'_i) = (\mathcal{M}_0, w_0)$ .

### 3 Tableau method

We consider  $2n + 2$  formulas,  $\varphi_0, \dots, \varphi_n, \psi \in \mathcal{L}$  and  $\varphi'_1, \dots, \varphi'_n \in \mathcal{L}'$ , and by  $\mathcal{P} \subset \Phi'$  we denote the finite set of all atomic events appearing in one of the event formulas  $\varphi'_1, \dots, \varphi'_n$ .  $i$  ranges over  $\{0, \dots, n\}$  and  $j$  ranges over  $\{1, \dots, n\}$ . We want to address the problem of deciding whether  $\varphi_0, \varphi'_1, \varphi_1, \dots, \varphi'_n, \varphi_n \stackrel{\perp}{=} \psi$  holds. To do so we equivalently decide whether this does not hold, *i.e.* whether there exist a pointed  $\mathcal{L}$ -model  $(\mathcal{M}_0, w_0)$  and  $n$  pointed  $\mathcal{L}'$ -models  $(\mathcal{M}'_j, w'_j)$  such that for all  $i$ ,  $(\mathcal{M}_i, w_i) = (\mathcal{M}_0, w_0) \otimes (\mathcal{M}'_1, w'_1) \otimes \dots \otimes (\mathcal{M}'_i, w'_i)$  is defined and  $\mathcal{M}_i, w_i \models \varphi_i$ , for all  $j$   $\mathcal{M}'_j, w'_j \models \varphi'_j$ , and  $\mathcal{M}_n, w_n \models \neg\psi$ . In other terms, deciding whether  $\varphi_0, \varphi'_1, \varphi_1, \dots, \varphi'_n, \varphi_n \stackrel{\perp}{=} \psi$  holds reduces to the following problem called the *satisfiability problem*:

- Input:  $\varphi_0, \dots, \varphi_n, \psi \in \mathcal{L}$ ,  $\varphi'_1, \dots, \varphi'_n \in \mathcal{L}'$  and  $\text{Pre}|_{\mathcal{P}}$ , the restriction of  $\text{Pre}$  to the domain  $\mathcal{P}$ .
- Output: yes iff there exist a pointed  $\mathcal{L}$ -model  $(\mathcal{M}_0, w_0)$  and  $n$  pointed  $\mathcal{L}'$ -models  $(\mathcal{M}'_j, w'_j)$  such that for all  $j$ ,  $\mathcal{M}'_j, w'_j \models \varphi'_j$ , for all  $i$ ,  $(\mathcal{M}_i, w_i) = (\mathcal{M}_0, w_0) \otimes (\mathcal{M}'_1, w'_1) \otimes \dots \otimes (\mathcal{M}'_i, w'_i)$  is defined and  $\mathcal{M}_i, w_i \models \varphi_i$ , and  $\mathcal{M}_n, w_n \models \psi$ .

In the rest of the paper, when the initial pointed model  $(\mathcal{M}_0, w_0)$  and the pointed event models  $(\mathcal{M}'_i, w'_i)$  are clear from the context, we shall write  $\mathcal{M}_i$  for  $\mathcal{M}_0 \otimes \mathcal{M}'_1 \otimes \dots \otimes \mathcal{M}'_i$  and  $w_i$  for  $(w_0, w'_1, \dots, w'_i)$ .

#### 3.1 Tableau method description

Let  $\mathfrak{Lab}$  be a countable set of labels designed to represent worlds of the initial epistemic model. For all integers  $i$ , let  $\mathfrak{Lab}_i$  be a countable set of labels designed to represent events of the  $i^{\text{th}}$  event model. We suppose that  $\mathfrak{Lab}$  and  $\mathfrak{Lab}_i$  for all  $i$  are disjoint.

Our tableau method manipulates terms that we call *tableau terms* and they are of the following kind:

- $(\sigma \varphi)$  means that  $\mathcal{M}_0, w_0 \models \varphi$ , where  $w_0$  is the world of the initial epistemic model  $\mathcal{M}_0$  represented by  $\sigma$  and  $\varphi$  is a formula of  $\mathcal{L}$ ;
- $(\sigma_i \varphi')$  means that  $\mathcal{M}'_i, w'_i \models \varphi'$ , where  $w'_i$  is the event of the  $i^{\text{th}}$  event model  $\mathcal{M}'_i$  represented by  $\sigma_i$ , and  $\varphi'$  is a formula of  $\mathcal{L}'$ ;
- $(\sigma \sigma_1 \dots \sigma_i \varphi)$  means that  $\mathcal{M}_i, (w_0, w'_1, \dots, w'_i) \models \varphi$ , where  $w_0$  is the world of  $\mathcal{M}_0$  represented by  $\sigma$ ,  $w'_k \in \mathcal{M}'_k$  is the event represented by  $\sigma_k$ , and  $\varphi$  is a formula of  $\mathcal{L}$ ;
- $(\sigma \sigma_1 \dots \sigma_i 0)$  means that  $(w_0, w'_1, \dots, w'_i)$  is not in  $\mathcal{M}_i$ , where  $w_0$  is the world of  $\mathcal{M}_0$  represented by  $\sigma$  and  $w'_k \in \mathcal{M}'_k$  is the event represented by  $\sigma_k$ ;
- $(R_a \sigma \sigma')$  means that the worlds  $w$  and  $u$  of  $\mathcal{M}_0$  represented respectively by  $\sigma$  and  $\sigma'$  are such that  $wR_a u$ ;
- $(R_a^i \sigma_i \sigma'_i)$  means that in  $\mathcal{M}'_i$ , the events  $w'$  and  $u'$  represented by  $\sigma_i$  and  $\sigma'_i$  are such that  $w'R_a^i u'$ ;
- $\perp$  denotes an inconsistency.

We also use *generic labels*  $\Sigma$  to simplify notations:  $\Sigma$  can be either  $\sigma$ ,  $\sigma_i$  or  $\sigma \sigma_1 \dots \sigma_i$ .  $(R_a \Sigma \Sigma')$  is interpreted in the following way: if  $\Sigma = \sigma$ ,  $\Sigma' = \sigma'$  then  $(R_a \Sigma \Sigma')$  denotes  $(R_a \sigma \sigma')$ , if  $\Sigma = \sigma_i$ ,  $\Sigma' = \sigma'_i$  then  $(R_a \Sigma \Sigma')$  denotes  $(R_a^i \sigma_i \sigma'_i)$ , and if  $\Sigma = \sigma \sigma_1 \dots \sigma_i$ ,  $\Sigma' = \sigma' \sigma'_1 \dots \sigma'_i$  then  $(R_a \Sigma \Sigma')$  denotes  $(R_a \sigma \sigma')(R_a^1 \sigma_1 \sigma_1) \dots (R_a^i \sigma_i \sigma_i)$ .

A *tableau rule* is represented by a *numerator*  $\mathcal{N}$  above a line and a finite list of *denominators*  $\mathcal{D}_1, \dots, \mathcal{D}_k$  below this line, separated by vertical bars:

$$\frac{\mathcal{N}}{\mathcal{D}_1 \mid \dots \mid \mathcal{D}_k}$$

The numerator and the denominators are finite sets of tableau terms.

A *tableau* for a tuple  $(\varphi_0, \dots, \varphi_n, \varphi'_1, \dots, \varphi'_n)$  of formulas is a finite tree with a set of tableau terms at each node. The root is  $\Gamma^0 = \{(\sigma \varphi_0), (\sigma \sigma_1 \varphi_1), \dots, (\sigma \sigma_1 \dots \sigma_n \varphi_n), (\sigma_1 \varphi'_1), \dots, (\sigma_n \varphi'_n)\} \cup \{(\sigma_1 p_1^{'+}), \dots, (\sigma_n p_n^{'+})\}$ , where  $p_1^{'+}, \dots, p_n^{'+}$  are fresh new atomic events. Indeed, in our tableau method, when a new event label  $\sigma_i$  is added to the set  $\Gamma$  of a node, it is assigned a fresh new atomic event  $p^{'+}$ . By fresh we mean that they do not appear in any event formula  $\varphi'$  in  $\Gamma$ , and their precondition is also a fresh new atomic proposition that appears neither in any formula nor in any precondition of any atomic event  $p'$  of  $\Gamma$ . We denote it by  $\text{Pre}(p^{'+}) = p^+$ . We abuse notations by writing  $p' \in \Gamma$  whenever  $p'$  occurs in a formula appearing in a tableau term in  $\Gamma$ . Those additional fresh atomic propositions and events are used in the tableau method to avoid the problematic case of events being built with trivial precondition.

A rule with numerator  $\mathcal{N}$  is *applicable* to a node carrying a set  $\Gamma$  if  $\Gamma$  contains an instance of  $\mathcal{N}$ . If no rule is applicable,  $\Gamma$  is said to be *saturated*. We call a node  $n$  an *end node* if the set of formulas  $\Gamma$  it carries is saturated, or if  $\perp \in \Gamma$ . The tableau is extended as follows:

1. Choose a leaf node  $n$  carrying  $\Gamma$  where  $n$  is not an end node, and choose a rule  $\rho$  applicable to  $n$ .
2. (a) If  $\rho$  has only one denominator, add the appropriate instantiation to  $\Gamma$ .  
 (b) If  $\rho$  has  $k$  denominators with  $k > 1$ , create  $k$  successor nodes for  $n$ , where each successor  $i$  carries the union of  $\Gamma$  with an appropriate instantiation of denominator  $\mathcal{D}_i$ .

A branch in a tableau is a path from the root to an end node. A branch is *closed* if its end node contains  $\perp$ , otherwise it is *open*. A tableau is *closed* if all its branches are closed, otherwise it is *open*. We write  $\varphi_0, \varphi'_1, \varphi_1, \dots, \varphi'_n, \varphi_n \vdash \psi$  if there is a closed tableau for  $(\varphi_0, \dots, \varphi_{n-1}, \varphi_n \wedge \neg\psi, \varphi'_1, \dots, \varphi'_n)$ .

### 3.2 Tableau rules

- Common rules for epistemic formulas and event formulas:

$$\frac{(\Sigma \varphi \wedge \psi)}{(\Sigma \varphi) (\Sigma \psi)} \wedge \quad \frac{(\Sigma \neg(\varphi \wedge \psi))}{(\Sigma \neg\varphi) \mid (\Sigma \neg\psi)} \neg\wedge \quad \frac{(\Sigma \neg\neg\varphi)}{(\Sigma \varphi)} \neg \quad \frac{(\Sigma p)(\Sigma \neg p)}{\perp} \perp$$

$$\frac{(\Sigma \langle B_a \rangle \varphi)}{(R_a \Sigma \Sigma^+) (\Sigma^+ \varphi) (\sigma_1^+ p_1^{'+}) \dots (\sigma_i^+ p_i^{'+})} \langle B_a \rangle \quad \frac{(\Sigma B_a \varphi) (R_a \Sigma \Sigma')}{(\Sigma' \varphi) \mid (\Sigma' 0)} B_a$$

where  $p$  is in  $\Phi \cup \Phi'$ ,  $\sigma_1^+ \dots \sigma_i^+$  are the  $\sigma_k^+$  in the fresh generic label  $\Sigma^+$  (none if  $\Sigma^+ = \sigma^+$ ) and  $p_k^{'+}$  are fresh new atomic events with  $\text{Pre}(p_i^{'+}) = p_i^+$ .

- Specific rule for event formulas:

$$\frac{(\sigma_i p')(\sigma_i p'')}{\perp} \text{Excl}$$

where  $p', p''$  are not fresh ( $p', p'' \in \Gamma^0$ ) and  $p' \neq p''$ .

- Specific rules for epistemic formulas:

$$\frac{(\sigma \sigma_1 \dots \sigma_i p)}{(\sigma p)} \leftarrow_1 \quad \frac{(\sigma \sigma_1 \dots \sigma_i \neg p)}{(\sigma \neg p)} \leftarrow_2$$

$$\frac{(\sigma 0)}{\perp} \text{Pre}_0 \quad \frac{(\sigma \sigma_1 \dots \sigma_i \varphi)(\sigma_i p')}{(\sigma \sigma_1 \dots \sigma_{i-1} \text{Pre}(p'))} \text{Pre}_1 \quad \varphi \neq 0, i > 0$$

$$\frac{(\sigma \sigma_1 \dots \sigma_i 0)(\sigma_i p')}{(\sigma \sigma_1 \dots \sigma_{i-1} \neg \text{Pre}(p')) \mid (\sigma \sigma_1 \dots \sigma_{i-1} 0)} \text{Pre}_2 \quad i > 0$$

We explain informally the meaning of these rules. The boolean rules, rule  $\langle B_a \rangle$  and rule  $B_a$  are classic. Rule Excl enforces the Exclusivity of event models. It does not apply to the fresh atomic events that we add at the creation of each event label, because a “meaningful” atomic event may be added to such a label; however these fresh events are removed from the final constructed models unless they are the only atomic event in their possible event. Rules  $\leftarrow_1$  and  $\leftarrow_2$  reflect the fact that for a world  $(w_0, w'_1, \dots, w'_i)$  in  $\mathcal{M}_i$ , by definition of the update product,  $\mathcal{M}_i, (w_0, w'_1, \dots, w'_i) \models p$  if, and only if,  $\mathcal{M}_0, w_0 \models p$ . Rule Pre<sub>1</sub> says that if  $(w_0, w'_1, \dots, w'_i)$  is in  $\mathcal{M}_i$  and  $\mathcal{M}'_i, w'_i \models p'$ , then  $(w_0, w'_1, \dots, w'_{i-1})$  is in  $\mathcal{M}_{i-1}$  and  $\mathcal{M}_{i-1}, (w_0, w'_1, \dots, w'_{i-1})$  verifies Pre( $p'$ ). Rule Pre<sub>2</sub> says that if  $(w_0, w'_1, \dots, w'_i)$  is *not* in  $\mathcal{M}_i$  and  $\mathcal{M}'_i, w'_i \models p'$ , either  $(w_0, w'_1, \dots, w'_{i-1})$  is not in  $\mathcal{M}_{i-1}$ , or it is in  $\mathcal{M}_{i-1}$  but  $\mathcal{M}_{i-1}, (w_0, w'_1, \dots, w'_{i-1}) \not\models \text{Pre}(p')$ . Finally, Rule Pre<sub>0</sub> is used to forbid the rightmost choice in Rule Pre<sub>2</sub> when  $i = 1$ . Indeed it would make no sense to say that the world associated to a label  $\sigma$  must not be in the initial model  $\mathcal{M}_0$  when it has been created to be in  $\mathcal{M}_0$ .

**Proposition 3 (Tableau method soundness)** *For all  $\varphi_0, \dots, \varphi_n, \psi \in \mathcal{L}$ , for all  $\varphi'_1, \dots, \varphi'_n \in \mathcal{L}'$ ,  $\varphi_0, \varphi'_1, \varphi_1, \dots, \varphi'_n, \varphi_n \stackrel{\perp}{\vdash} \psi$  implies  $\varphi_0, \varphi'_1, \varphi_1, \dots, \varphi'_n, \varphi_n \stackrel{\perp}{\vDash} \psi$ .*

**Proof** We prove it by contrapositive. Suppose that there exist a pointed  $\mathcal{L}$ -model  $(\mathcal{M}_0, w_0)$ , and  $n$   $\mathcal{L}'$ -models  $(\mathcal{M}'_j, w'_j)$  such that for all  $i$ ,  $\mathcal{M}_i, w_i$  is defined and  $\mathcal{M}_i, w_i \models \varphi_i$ , for all  $j$ ,  $\mathcal{M}'_j, w'_j \models \varphi'_j$  and  $\mathcal{M}_n, w_n \not\models \psi$ . We must prove that every tableau for  $(\varphi_0, \dots, \varphi_{n-1}, \varphi_n \wedge \neg \psi, \varphi'_1, \dots, \varphi'_n)$  has an open branch (the proof of termination is in the proof of Lemma 8).

We say that a set  $\Sigma$  of tableau terms is *interpretable* with respect to a subset of relevant atomic events  $S'$  if there exists an  $\mathcal{L}$ -model  $\mathcal{M}_0$ ,  $n$   $\mathcal{L}'$ -models  $\mathcal{M}'_i$ , partial functions  $\nu : \mathcal{L}ab \rightarrow W$  and  $\nu_i : \mathcal{L}ab_i \rightarrow W'_i$  (where  $i$  ranges over  $\{1 \dots, n\}$ ) defined exactly on labels that appear in  $\Sigma$  such that  $(\mathcal{M}, (\mathcal{M}'_i)_{1 \leq i \leq n}, \nu, (\nu_i)_{1 \leq i \leq n})$  makes all the tableau terms in  $\Sigma$  true for the following semantics  $\models_T$ :

$$\begin{aligned} (\mathcal{M}_0, (\mathcal{M}'_i)_{1 \leq i \leq n}, \nu, (\nu_i)_{1 \leq i \leq n}) \models_T (\sigma \varphi) & \quad \text{iff } \mathcal{M}, \nu(\sigma) \models \varphi \\ (\mathcal{M}_0, (\mathcal{M}'_i)_{1 \leq i \leq n}, \nu, (\nu_i)_{1 \leq i \leq n}) \models_T (\sigma_i \varphi') & \quad \text{iff } \varphi' = p' \text{ for some } p' \notin S' \text{ or} \\ & \quad \mathcal{M}'_i, \nu_i(\sigma_i) \models \varphi' \\ (\mathcal{M}_0, (\mathcal{M}'_i)_{1 \leq i \leq n}, \nu, (\nu_i)_{1 \leq i \leq n}) \models_T (R_a \sigma \sigma') & \quad \text{iff } \nu(\sigma) R_a \nu(\sigma') \\ (\mathcal{M}_0, (\mathcal{M}'_i)_{1 \leq i \leq n}, \nu, (\nu_i)_{1 \leq i \leq n}) \models_T (R'_a \sigma_i \sigma'_i) & \quad \text{iff } \nu_i(\sigma_i) R'_a \nu_i(\sigma'_i) \end{aligned}$$

$$\begin{aligned}
(\mathcal{M}_0, (\mathcal{M}'_i)_{1 \leq i \leq n}, \nu, (\nu_i)_{1 \leq i \leq n}) \models_T (\sigma \sigma_1 \dots \sigma_i 0) & \text{ iff } (\nu(\sigma), \nu_1(\sigma_1), \dots, \nu_i(\sigma_i)) \notin \mathcal{M}_i \\
(\mathcal{M}_0, (\mathcal{M}'_i)_{1 \leq i \leq n}, \nu, (\nu_i)_{1 \leq i \leq n}) \models_T (\sigma \sigma_1 \dots \sigma_i \varphi_i) & \text{ iff} \\
& w_i \in \mathcal{M}_i, \text{ where } w_i = (\nu(\sigma), \nu_1(\sigma_1), \dots, \nu_i(\sigma_i)), \text{ and } \mathcal{M}_i, w_i \models \varphi_i \\
(\mathcal{M}_0, (\mathcal{M}'_i)_{1 \leq i \leq n}, \nu, (\nu_i)_{1 \leq i \leq n}) \models_T \perp & \text{ iff false}
\end{aligned}$$

Since  $\varphi_0, \varphi'_1, \varphi_1, \dots, \varphi'_n, \varphi_n \stackrel{\perp}{\vdash} \psi$ , the set  $\Gamma^0 = \{(\sigma \varphi_0), (\sigma \sigma_1 \varphi_1), \dots, (\sigma \sigma_1 \dots \sigma_{n-1} \varphi_{n-1})\} \cup \{(\sigma \sigma_1 \dots \sigma_n \varphi_n \wedge \neg \psi), (\sigma_1 \varphi'_1), \dots, (\sigma_n \varphi'_n), (\sigma_1 p_1^{'+}), \dots, (\sigma_n p_n^{'+})\}$  is interpretable. Moreover, if a set of formulas is interpretable, it does not contain  $\perp$ . So if we prove that when the numerator of a rule is interpretable, one of the denominators also is, then we have that every tableau for  $(\varphi_0, \dots, \varphi_{n-1}, \neg \varphi_n, \varphi'_1, \dots, \varphi'_n)$  has an open branch. We only prove it for Excl,  $\langle B_a \rangle$  and  $B_a$  when  $\Sigma = \sigma \sigma_1 \dots \sigma_i$ , and the specific rules for epistemic formulas, the proof for the other rules being standard. In the following, when  $\nu$  is a function, we let  $\nu(x \mapsto a)$  be the function that maps  $x$  to  $a$  and  $y$  to  $\nu(y)$  if  $y \neq x$ .

Rule  $\langle B_a \rangle$ : If  $(\mathcal{M}_0, (\mathcal{M}'_i)_{1 \leq i \leq n}, \nu, (\nu_i)_{1 \leq i \leq n}) \models_T (\sigma \sigma_1 \dots \sigma_i \langle B_a \rangle \varphi)$  then, if  $w_i = (\nu(\sigma), \nu_1(\sigma_1), \dots, \nu_i(\sigma_i))$ , then  $\mathcal{M}_i, w_i \models \langle B_a \rangle \varphi$ . So there exists  $u_i = (u, u'_1, \dots, u'_i) \in W_i$  such that  $w_i R_a^i u_i$  and  $\mathcal{M}_i, u_i \models \varphi$ . Since  $w R_a^i u$  we have that  $\nu(\sigma) R_a u$ ,  $\nu_j(\sigma_j) R_a^j u'_j$  for all  $j \leq i$ . So by letting  $\nu' := \nu(\sigma^+ \mapsto u)$ ,  $\nu'_j := \nu_j(\sigma_j^+ \mapsto u'_j)$ , and  $\mathcal{M}'_j$  being the same as  $\mathcal{M}'_j$  except that its valuation maps  $p_j^{'+}$  to  $\{u'_j\}$ , we have that  $(\mathcal{M}_0, (\mathcal{M}''_i)_{1 \leq i \leq n}, \nu', (\nu'_i)_{1 \leq i \leq n}) \models_T \{(R_a \sigma \sigma^+)(R_a^1 \sigma_1 \sigma_1^+) \dots (R_a^i \sigma_i \sigma_i^+)(\sigma^+ \sigma_1^+ \dots \sigma_i^+ \varphi)(\sigma_1^+ p_1^{'+}) \dots (\sigma_i^+ p_i^{'+})\}$ .

Rule  $B_a$ :

If  $(\mathcal{M}_0, (\mathcal{M}'_i)_{1 \leq i \leq n}, \nu, (\nu_i)_{1 \leq i \leq n}) \models_T \{(\sigma \sigma_1 \dots \sigma_i B_a \varphi)(R_a \sigma \sigma')(R_a^1 \sigma_i \sigma'_i) \dots (R_a^i \sigma_i \sigma'_i)\}$ , then, noting  $w_i = (\nu(\sigma), \nu_1(\sigma_1), \dots, \nu_i(\sigma_i))$ ,  $w_i \in \mathcal{M}_i$ ,  $(\mathcal{M}_i, w_i) \models B_a \varphi$ ,  $\nu(\sigma) R_a \nu(\sigma')$  and  $\nu_i(\sigma_j) R_a^j \nu_j(\sigma'_j)$  for all  $j \leq i$ . Note  $u_i = (\nu(\sigma'), \nu_1(\sigma'_1), \dots, \nu_i(\sigma'_i))$ . Either  $u_i \notin \mathcal{M}_i$ , and  $(\mathcal{M}_0, (\mathcal{M}'_i)_{1 \leq i \leq n}, \nu, (\nu_i)_{1 \leq i \leq n}) \models_T (\sigma' \sigma'_1 \dots \sigma'_i 0)$ , or  $u_i \in \mathcal{M}_i$ , and by definition of  $R_a^i$  we have that  $w_i R_a^i u_i$ , hence  $\mathcal{M}_i, u_i \models \varphi$  and we conclude that  $(\mathcal{M}_0, (\mathcal{M}'_i)_{1 \leq i \leq n}, \nu, (\nu_i)_{1 \leq i \leq n}) \models_T (\sigma' \sigma'_1 \dots \sigma'_i \varphi)$ .

Rules  $\leftarrow_1$  and  $\leftarrow_2$ : If  $(\mathcal{M}_0, (\mathcal{M}'_i)_{1 \leq i \leq n}, \nu, (\nu_i)_{1 \leq i \leq n}) \models_T (\sigma \sigma_1 \dots \sigma_i p)$  then, noting  $w_i = (\nu(\sigma), \nu_1(\sigma_1), \dots, \nu_i(\sigma_i))$ ,  $w_i \in \mathcal{M}_i$  and  $(\mathcal{M}_i, w_i) \models p$ . Since  $V(\nu(\sigma)) = V^i(w_i)$  by definition of the product,  $\mathcal{M}_0, \nu(\sigma) \models p$  hence  $(\mathcal{M}_0, (\mathcal{M}'_i)_{1 \leq i \leq n}, \nu, (\nu_i)_{1 \leq i \leq n}) \models_T (\sigma p)$ . The case of rule  $\leftarrow_2$  is proved similarly.

Rules Pre<sub>1</sub> and Pre<sub>2</sub>:

Suppose  $(\mathcal{M}_0, (\mathcal{M}'_i)_{1 \leq i \leq n}, \nu, (\nu_i)_{1 \leq i \leq n}) \models_T \{(\sigma \sigma_1 \dots \sigma_i \varphi_i)(\sigma_i p')\}$  for some  $\varphi \neq 0$  and note  $w_i = (\nu(\sigma), \nu_1(\sigma_1), \dots, \nu_i(\sigma_i))$  and  $w_{i-1} = (\nu(\sigma), \nu_1(\sigma_1), \dots, \nu_{i-1}(\sigma_{i-1}))$ . Then  $w_i \in \mathcal{M}_i$  and  $\mathcal{M}'_i, \nu_i(\sigma_i) \models p'$ . Since  $w_i \in \mathcal{M}_i$ , we have that  $\mathcal{M}_{i-1}, w_{i-1} \models \text{Pre}(p')$ . So  $(\mathcal{M}_0, (\mathcal{M}'_i)_{1 \leq i \leq n}, \nu, (\nu_i)_{1 \leq i \leq n}) \models_T (\sigma \sigma_1 \dots \sigma_{i-1} \text{Pre}(p'))$ . Concerning Pre<sub>2</sub>, If  $(\mathcal{M}_0, (\mathcal{M}'_i)_{1 \leq i \leq n}, \nu, (\nu_i)_{1 \leq i \leq n}) \models_T \{(\sigma \sigma_1 \dots \sigma_i 0)(\sigma_i p')\}$ , then  $(\nu(\sigma), \nu_1(\sigma_1), \dots, \nu_i(\sigma_i))$  is not in  $\mathcal{M}_i$  and  $\mathcal{M}'_i, \nu_i(\sigma_i) \models p'$ . By definition,  $\text{Pre}(\nu_i(\sigma_i)) = \text{Pre}(p')$ . There are two cases: either  $w_{i-1} = (\nu(\sigma), \nu_1(\sigma_1), \dots, \nu_{i-1}(\sigma_{i-1}))$  is not in  $\mathcal{M}_{i-1}$ , in which case  $(\mathcal{M}_0, (\mathcal{M}'_i)_{1 \leq i \leq n}, \nu, (\nu_i)_{1 \leq i \leq n}) \models_T (\sigma \sigma_1 \dots \sigma_{i-1} 0)$ , or  $w_{i-1}$  is in  $\mathcal{M}_{i-1}$ , and  $\mathcal{M}_{i-1}, w_{i-1} \not\models \text{Pre}(\nu_i(\sigma_i))$  hence  $(\mathcal{M}_0, (\mathcal{M}'_i)_{1 \leq i \leq n}, \nu, (\nu_i)_{1 \leq i \leq n}) \models_T (\sigma \sigma_1 \dots \sigma_{i-1} \neg \text{Pre}(p'))$ .

Rules Pre<sub>0</sub> and Excl are sound since their numerator is never interpretable.  $\square$

**Proposition 4 (Tableau method completeness)** For all  $\varphi_0, \dots, \varphi_n, \psi \in \mathcal{L}$ , for all  $\varphi'_1, \dots, \varphi'_n \in \mathcal{L}'$ ,  $\varphi_0, \varphi'_1, \varphi_1, \dots, \varphi'_n, \varphi_n \stackrel{\perp}{\vdash} \psi$  implies  $\varphi_0, \varphi'_1, \varphi_1, \dots, \varphi'_n, \varphi_n \stackrel{\perp}{\vdash} \psi$ .

We prove the contrapositive. Suppose there is a tableau for  $(\varphi_0, \dots, \varphi_{n-1}, \varphi_n \wedge \neg\psi, \varphi'_1, \dots, \varphi'_n)$  that has an open branch. We prove that there exist a pointed  $\mathcal{L}$ -model  $(\mathcal{M}_0, w_0)$  and  $n$  pointed  $\mathcal{L}'$ -models  $(\mathcal{M}'_j, w'_j)$  such that for all  $j$ ,  $\mathcal{M}'_j, w'_j \models \varphi'_j$ , for all  $i$ ,  $w_i \in \mathcal{M}_i$  and  $\mathcal{M}_i, w_i \models \varphi_i$ , and  $\mathcal{M}_n, w_n \models \neg\psi$ .

Let  $\Gamma_f$  be the set of tableau terms carried by the end node of the open branch. We define  $\mathcal{M}_0, \mathcal{M}'_1, \dots, \mathcal{M}'_n$  as follows.

- Let  $\mathcal{M}_0 = (W, R, V)$  with  $W = \{w_\sigma \mid (\sigma \psi) \in \Gamma_f\}$ ,  $R(a) = \{(w_\sigma, w_{\sigma'}) \mid (R_a \sigma \sigma') \in \Gamma_f\}$  and  $V(p) = \{w_\sigma \mid (\sigma p) \in \Gamma_f\}$ .
- Let  $\mathcal{M}'_i = (W'_i, R'_i, V'_i)$  with  $W'_i = \{w'_{\sigma_i} \mid (\sigma_i \varphi') \in \Gamma_f\}$ ,  $R'_i(a) = \{(w'_{\sigma_i}, w'_{\sigma'_i}) \mid (R'_a \sigma_i \sigma'_i) \in \Gamma_f\}$  and
 
$$V'_i(p') = \begin{cases} \{w'_{\sigma_i} \mid (\sigma_i p') \in \Gamma_f\} & \text{if } p' \in \Gamma_0 \\ \{w'_{\sigma_i} \mid (\sigma_i p') \in \Gamma_f \text{ and } (\sigma_i p'') \notin \Gamma_f \text{ for all } p'' \neq p'\} & \text{otherwise.} \end{cases}$$

Notice that each event model  $\mathcal{M}'_i$  verifies the Exclusivity condition: by saturation of rule Excl, there can not be  $p'$  and  $p''$  in  $\Gamma_0$  in the same event otherwise the branch would be closed, and the fresh  $p'^+$  assigned to an event label  $\sigma_i$  at its creation is kept in  $w'_{\sigma_i}$  only if there is no other atomic event in this event. So for every event  $w'_{\sigma_i}$ , there is exactly one atomic event  $p'$  such that  $w'_{\sigma_i} \in V'_i(p')$ .

Finally, for  $i \in \{1, \dots, n\}$  we define  $\mathcal{M}_i$  as  $\mathcal{M}_0 \otimes \mathcal{M}'_1 \otimes \dots \otimes \mathcal{M}'_i$ . Lemmas 5 and 6 below establish the completeness of our tableau method.

**Lemma 5** *Let  $\varphi \in \mathcal{L}$  and  $\varphi' \in \mathcal{L}'$  such that  $\varphi'$  not a fresh atomic event. If  $(\sigma \varphi) \in \Gamma^f$  then  $\mathcal{M}_0, w_\sigma \models \varphi$ , and if  $(\sigma_i \varphi') \in \Gamma^f$ , then  $\mathcal{M}'_i, w'_{\sigma_i} \models \varphi'$ .*

**Proof** We only prove it for  $\mathcal{M}_0$ , it is similar for  $\mathcal{M}'_i$ . The proof is done by induction on  $\varphi$ .

$p, \neg p$ : by definition of  $\mathcal{M}_0$ ,  $w_\sigma \in V(p)$  if and only if  $(\sigma p) \in \Gamma^f$ . So if  $(\sigma p) \in \Gamma^f$ ,  $\mathcal{M}_0, w_\sigma \models p$ , and if  $(\sigma \neg p) \in \Gamma^f$ , we have that  $(\sigma p) \notin \Gamma^f$  (otherwise the branch would be closed by saturation of rule  $\perp$ ), hence  $\mathcal{M}_0, w_\sigma \models \neg p$ . As for the case  $\varphi \wedge \psi$ , by saturation of rule  $\wedge$ ,  $\Gamma^f$  also contains  $(\sigma \varphi)$  and  $(\sigma \psi)$ . By induction hypothesis we have that  $\mathcal{M}_0, w_\sigma \models \varphi$  and  $\mathcal{M}_0, w_\sigma \models \psi$ , so  $\mathcal{M}_0, w_\sigma \models \varphi \wedge \psi$ . The cases  $\neg(\varphi \wedge \psi)$  and  $\neg\neg\varphi$  are proved similarly.

$\langle B_a \rangle \varphi$ : By saturation of rule  $\langle B_a \rangle$  (with  $\Sigma$  of the form  $\sigma$ ), there exists  $\sigma'$  such that  $(R_a \sigma \sigma') \in \Gamma^f$  and  $(\sigma' \varphi) \in \Gamma^f$ . By induction hypothesis  $\mathcal{M}_0, w_{\sigma'} \models \varphi$ , and  $w_\sigma R_a w_{\sigma'}$  holds by construction of  $\mathcal{M}_0$ , so  $\mathcal{M}_0, w_\sigma \models \langle B_a \rangle \varphi$ .

$B_a \varphi$ : Take some  $w_{\sigma'}$  in  $W_0$  such that  $w_\sigma R_a w_{\sigma'}$  holds, we prove that  $\mathcal{M}_0, w_{\sigma'} \models \varphi$  and conclude that  $\mathcal{M}_0, w_\sigma \models B_a \varphi$ . Since  $w_\sigma R_a w_{\sigma'}$  holds we know by construction of  $\mathcal{M}$  that  $(R_a \sigma \sigma')$  is in  $\Gamma^f$ . So by saturation of rule  $B_a$ ,  $(\sigma' \varphi)$  also belongs to  $\Gamma^f$ , and by induction hypothesis  $\mathcal{M}_0, w_{\sigma'} \models \varphi$ . □

**Lemma 6** *Let  $\varphi \in \mathcal{L}$ . For all  $i$  such that  $0 \leq i \leq n$ , if  $(\sigma \sigma_1 \dots \sigma_i \varphi) \in \Gamma^f$ , then  $w_i = (w_\sigma, w'_{\sigma_1}, \dots, w'_{\sigma_i}) \in \mathcal{M}_i$  and  $\mathcal{M}_i, w_i \models \varphi$ . If  $(\sigma \sigma_1 \dots \sigma_i 0) \in \Gamma^f$ , then  $w_i \notin \mathcal{M}_i$ .*

**Proof** We prove it by double induction on  $i$  and  $\varphi'$ .

For  $i = 0$ : if  $\varphi \in \mathcal{L}$ , the result holds by Lemma 5. If  $\varphi = 0$ ,  $(\sigma 0) \in \Gamma^f$ , hence by saturation of rule Pre $_0$ , the branch is closed. Contradiction.

Now assume that the result holds for  $i < n$ , we prove it for  $i + 1$ .

We write  $w_i = (w_\sigma, w'_{\sigma_1}, \dots, w'_{\sigma_i})$ ,  $w_{i+1} = (w_\sigma, w'_{\sigma_1}, \dots, w'_{\sigma_{i+1}})$  and we start with the case where  $(\sigma \ \sigma_1 \dots \sigma_{i+1} \ 0) \in \Gamma^f$ . Let  $p'$  be the only atomic event true in event  $w'_{\sigma_{i+1}}$ . We have  $(\sigma_{i+1} \ p') \in \Gamma^f$ , and by saturation of Rule Pre<sub>2</sub>, we obtain two cases. If  $(\sigma \ \sigma_1 \dots \sigma_i \ \neg \text{Pre}(p')) \in \Gamma^f$ , then by induction hypothesis  $\mathcal{M}_i, w_i \not\models \text{Pre}(w'_{\sigma_{i+1}})$ , hence  $w_{i+1}$  is not in  $\mathcal{M}_{i+1}$ . If  $(\sigma \ \sigma_1 \dots \sigma_i \ 0) \in \Gamma^f$ , by induction hypothesis,  $w_i$  is not in  $\mathcal{M}_i$ , hence  $\mathcal{M}_{i+1}, w_{i+1}$  is not defined.

Now the case where  $(\sigma \ \sigma_1 \dots \sigma_{i+1} \ \varphi) \in \Gamma^f$  with  $\varphi \neq 0$ . We first prove that  $w_{i+1} \in \mathcal{M}_{i+1}$ : by construction of  $\mathcal{M}'_i$  there is exactly one  $p' \in \Phi'$  such that  $w'_{\sigma_{i+1}} \in V'_{i+1}(p')$ , and  $(\sigma_{i+1} \ p') \in \Gamma^f$ . By saturation of Rule Pre<sub>1</sub>,  $(\sigma \ \sigma_1 \dots \sigma_i \ \text{Pre}(p'))$  is in  $\Gamma^f$ , and by induction hypothesis,  $w_i \in \mathcal{M}_i$  and  $\mathcal{M}_i, w_i \models \text{Pre}(p')$ . Since  $\text{Pre}(w'_{\sigma_{i+1}}) = \text{Pre}(p')$ , we have that  $\mathcal{M}_i, w_i \models \text{Pre}(w'_{\sigma_{i+1}})$ , hence  $w_{i+1}$  is in  $\mathcal{M}_{i+1}$ .

We prove that  $\mathcal{M}_{i+1}, w_{i+1} \models \varphi$ .

$p, \neg p$ : By saturation of Rule  $\leftarrow_1$ ,  $(\sigma \ p) \in \Gamma^f$ , and so  $\mathcal{M}_0, w_\sigma \models p$  by Lemma 5. Since  $w_{i+1} \in \mathcal{M}_{i+1}$ ,  $\mathcal{M}_{i+1}, w_{i+1} \models p$  by definition of the product update. The proof for  $\neg p$  is similar to the case of  $p$ . The proof of the other boolean cases  $\varphi \wedge \psi$ ,  $\neg(\varphi \wedge \psi)$  and  $\neg\neg\varphi$  is obtained by applying straightforwardly the Induction Hypothesis.

$\langle B_a \rangle \varphi$ : If  $(\sigma \ \sigma_1 \dots \sigma_{i+1} \ \langle B_a \rangle \varphi) \in \Gamma^f$ , then by saturation of Rule  $\langle B_a \rangle$ ,  $(R_a \ \sigma \ \sigma^+)$ ,  $(R_a^1 \ \sigma_1 \ \sigma_1^+)$ ,  $\dots$ ,  $(R_a^i \ \sigma_i \ \sigma_i^+)$ ,  $(\sigma^+ \ \sigma_1^+ \dots \sigma_{i+1}^+ \ \varphi)$  and  $(\sigma_1^+ \ p_1^+)$ ,  $\dots$ ,  $(\sigma_{i+1}^+ \ p_{i+1}^+)$  are in  $\Gamma^f$ . Let  $u_{i+1} = (w_{\sigma^+}, w'_{\sigma_1^+}, \dots, w'_{\sigma_{i+1}^+})$ . Since  $(\sigma^+ \ \sigma_1^+ \dots \sigma_{i+1}^+ \ \varphi)$  is in  $\Gamma^f$ , by induction hypothesis,  $u_{i+1}$  is in  $\mathcal{M}_{i+1}$  and  $\mathcal{M}_{i+1}, u_{i+1} \models \varphi$ . By construction of  $\mathcal{M}_0$  and  $\mathcal{M}'_j$ , for  $1 \leq j \leq i+1$ , we have  $w_\sigma R_a w_{\sigma^+}$ ,  $w'_{\sigma_1} R_a^1 w'_{\sigma_1^+}$ ,  $\dots$ ,  $w'_{\sigma_{i+1}} R_a^{i+1} w'_{\sigma_{i+1}^+}$ , so  $w_{i+1} R_a^{i+1} u_{i+1}$ , hence the result.

$B_a \varphi$ : If  $(\sigma \ \sigma_1 \dots \sigma_{i+1} \ B_a \varphi) \in \Gamma^f$ , take  $u_{i+1} = (w_{\sigma'}, w'_{\sigma'_1}, \dots, w'_{\sigma'_{i+1}})$  in  $\mathcal{M}_{i+1}$  such that  $w_{i+1} R_a^{i+1} u_{i+1}$ , we prove that  $\mathcal{M}_{i+1}, u_{i+1} \models \varphi$ . By definition of the product and of the models  $\mathcal{M}_0$  and  $\mathcal{M}'_1, \dots, \mathcal{M}'_{i+1}$ , we have that  $(R_a \ \sigma \ \sigma')$ ,  $(R_a^1 \ \sigma_1 \ \sigma'_1)$ ,  $\dots$ ,  $(R_a^i \ \sigma_i \ \sigma'_i)$  are in  $\Gamma^f$ . By saturation of Rule  $B_a$ , whether  $(\sigma' \ \sigma'_1 \dots \sigma'_{i+1} \ \varphi)$  is in  $\Gamma^f$  in which case we conclude by induction hypothesis, or  $(\sigma' \ \sigma'_1 \dots \sigma'_{i+1} \ 0)$  is in  $\Gamma^f$  and by induction hypothesis  $u_{i+1}$  is not in  $\mathcal{M}_{i+1}$ , which is a contradiction. □

## 4 NEXPTIME-completeness and implementation

The NEXPTIME-completeness of our simple DEL-sequents [3] extends to generalized DEL-sequents.

**Proposition 7** *The satisfiability problem is NEXPTIME-complete*

**Lemma 8** *The satisfiability problem is in NEXPTIME.*

**Proof** The tableau rules presented in Subsection 3.2 give rise to a non-deterministic algorithm running in exponential time. The algorithm starts with the following set of tableau formulas

$$\Gamma^0 = \{(\sigma \ \varphi_0), (\sigma \ \sigma_1^0 \ \varphi_1), \dots, (\sigma \ \sigma_1^0 \dots \sigma_n^0 \ \varphi_n), (\sigma_1^0 \ \varphi'_1), \dots, (\sigma_n^0 \ \varphi'_n)\} \cup \{(\sigma_1^0 \ p_1^+), \dots, (\sigma_n^0 \ p_n^+)\}$$

where the  $p_1^+, \dots, p_n^+$  are fresh new atomic events.

Let  $\delta(\cdot)$  be the function that gives the modal depth of a given formula. Let  $max_{pre}$  be the maximum of  $\delta(\text{Pre}(p'))$  where  $p'$  ranges over the propositions appearing in the formulas  $\varphi'_1, \dots, \varphi'_n$ .

Let  $(\Delta W_i)_{i \in \{0, \dots, n\}}$  be defined as follows:

- $\Delta W_n = \delta(\varphi_n)$
- $\Delta W_i = \max(\delta(\varphi'_i) + max_{pre}, \delta(\varphi_i))$  if  $0 < i < n$
- $\Delta W_0 = \max(\delta(\varphi'_i) + max_{pre}, \delta(\varphi_i), \Delta W_1, \dots, \Delta W_n)$ .

The algorithm runs as follows. Let  $N = \Delta W_0$ . For  $t = 0$  to  $N$ , we execute:

1.  $\Gamma^{t'}$  := the saturation of  $\Gamma^t$  by rules  $\wedge, \neg\wedge, \neg, \perp, Excl, \leftarrow_1, \leftarrow_2, Pre_1, Pre_2$ ;
2. If  $\perp \in \Gamma^{t'}$ , we stop the current execution;
3.  $\Gamma^{t+1}$  := the set of tableau terms obtained by applying  $\langle B_a \rangle, B_a$  on  $\Gamma^{t'}$ .

Step 1 is non-deterministic and corresponds to a Boolean saturation of labels of depth  $i$ . It non-deterministically runs in linear size of  $\Gamma^i$ . Step 2 consists in checking if rule  $\perp$  has been executed. In this case, the current execution halts.

Remark that the term of the modal depth of  $\varphi'$  in terms of the form  $(\sigma_i \varphi')$  is strictly decreasing when we apply rule  $\langle B_a \rangle$  and  $B_a$ . So when  $t > \delta(\varphi'_i)$ , the set  $\Gamma^t$  will not contain any terms of the form  $(\sigma_i \varphi')$ .

Let us prove by induction that for all  $i$  we have the following property  $P(i)$ : for all  $t > \Delta W_i$ , the set  $\Gamma^t$  will not contain any term of the form  $(\sigma \sigma_1 \dots \sigma_i \psi)$ .

Let us begin with the basic case:  $i = n$ . Note that the maximal depth of formulas  $\psi$  in tableau terms corresponding to the last product model, that is, the modal depth of  $\psi$  in terms of the form  $(\sigma \sigma_1 \dots \sigma_n \psi)$  in  $\Gamma^t$  is strictly decreasing with  $t$  (see rule  $\langle B_a \rangle$  and  $B_a$ ). So when  $t > \Delta W_n$ , there is no more tableau formula of the form  $(\sigma \sigma_1 \dots \sigma_n \psi)$  in  $\Gamma^t$  where  $\psi$  is a formula or 0. So when  $t > \delta(\varphi)$ , the rules  $Pre_2$ ,  $\langle B_a \rangle$  and  $B_a$  will no more be applied to terms of the form  $(\sigma \sigma_1 \dots \sigma_n \psi)$ . So  $P(n)$  is true.

Now suppose  $P(i+1)$  and let us prove  $P(i)$ . Suppose there exists  $t > \Delta W_i$  such that  $\Gamma^t$  contains a term of the form  $(\sigma \sigma_1 \dots \sigma_i \psi)$ . As the modal depth decreases at each step, if  $\psi$  is a subformula of  $\varphi_i$ , we have  $t \leq \delta(\varphi_i)$ . Also by definition of  $\Delta W_i$ , we have  $\delta(\varphi_i) \leq \Delta W_i$ . This leads to a contradiction with  $t > \Delta W_i$ . So the formula  $\psi$  is either a subformula of a (possibility negated) precondition added from rule  $Pre_1$  or  $Pre_2$  (1) or is a literal from  $\leftarrow_1$  or  $\leftarrow_2$  (2).

In the case (1), it implies that  $t \leq \Delta W_i + 1 + max_{pre}$ . There is a contradiction. In the case (2), it implies that  $t \leq \Delta W_i$  for all  $i > 0$ . There is a contradiction.

That is why we have  $\Gamma^{N+1} = \emptyset$  and the algorithm has applied rules until saturation, that is, the set of tableau formulas  $\bigcup_{t=0}^N \Gamma^t$  is saturated. Now let us have a look at the time required to execute the algorithm. Let  $x$  be the size of the input, that is the sum of the sizes of  $\varphi_0, \dots, \varphi_n, \varphi'_1, \dots, \varphi'_n, \varphi$  and the function  $Pre$ . Step 1 saturates the nodes  $\sigma, \sigma_i$  and  $(\sigma \sigma_1 \dots \sigma_i)$  appearing in the tableau formulas in  $\Gamma^t$ . For each of those worlds, the saturation is linear in  $x$ . Step 3 creates new tableau terms for each  $\langle B_a \rangle$ -formula appearing in  $\Gamma^{t'}$ : more precisely each  $\langle B_a \rangle$ -formula creates at most  $n$ -new labels  $\sigma^+, \sigma_1^+, \dots, \sigma_n^+$ . So for each world in  $\Gamma^t$  it produces at most  $x$  worlds in the initial model,  $x$  worlds in each  $i$ -th event model,  $x$  worlds in the  $i$ -th product model, that is to say it produces at most  $(2x)^n$  new worlds. If we note  $y_t$  the maximal number of worlds in  $\Gamma^t$ , we have that  $y_{t+1} = (2x)^n y_t$ . So  $y_t = (2x)^{nt}$ . Now, remark



that  $N = \max(\max(\delta(\varphi'_1), \dots, \delta(\varphi'_n)) + \max_{pre}, \delta(\varphi_0), \dots, \delta(\varphi_n)) \leq x$ . The total number of created worlds is bounded by  $\sum_{t=0}^N y_t \leq \sum_{t=0}^x (2x)^{nt} \leq (2x)^{n(x+1)}$  and this construction takes an exponential amount of time.  $\square$

**Lemma 9** *The satisfiability problem is NEXPTIME-hard.*

**Proof** The satisfiability problem is NEXPTIME-hard because we can encode in it the satisfiability problem of a syntactic fragment. This syntactic fragment is studied in the proof of the Proposition 4.2 in [3] and we summarize here the result in the following fact.

**Claim 1** *The following problem*

- *input:  $\varphi_0, \varphi_1 \in \mathcal{L}$  and  $\varphi'_1 \in \mathcal{L}$  such that:*
  - *there is only one agent  $a$  occurring in  $\varphi_0, \varphi_1, \varphi'_1$ ;*
  - *$\varphi_1$  is of the form  $\psi_1 \wedge \bigwedge_{i=1}^{i \leq \delta(\varphi'_1)} B_a^i p'$  where  $\psi_1$  does not contain any atomic event;*
  - *$Pre(p') = \top$ .*
- *output: yes iff  $\varphi_0, \varphi'_1, \top \stackrel{\text{H}}{\vDash} \varphi_1$  does not hold.*

*is NEXPTIME-hard.*  $\square$

An implementation of the tableau method can be found at the following webpage:

<http://www.irisa.fr/prive/fschwarz/lotrecscheme/>.

The tableau rules are written in LotrecScheme [16] which is a term rewriting system designed for implementing tableau methods. The corresponding rules can be found directly in the software by clicking ‘open’ and ‘generalized DEL-sequents’.

The implemented rules are similar to those presented in Subsection 3.2. There are two main differences. The first one is that we tag worlds with  $ok$  and  $\neg ok$  to say respectively that the node belongs to the model or not. We use those two tags for a reason of efficiency. The second difference concerns the pattern-matching of a condition of a rule: as LotrecScheme does not enable an arbitrary number of terms to match in a condition, we adapt the method so that the numbers of terms in all rules are fixed. For instance, let us consider the rule  $B_a$ .  $(R_a \Sigma \Sigma')$  is a macro to denote an arbitrary number of terms whereas in the implementation  $(R_a \Sigma \Sigma')$  is not a macro but a term and we have to simulate the way relations are defined in the product update:

$$\frac{(R_a \Sigma \Sigma')(R_a a b)(\Sigma :: a ok)}{(R_a \Sigma :: a \Sigma' :: b)} \quad \text{and} \quad \frac{(R_a \Sigma :: a \Sigma' :: b)}{(R_a \Sigma \Sigma')(R_a a b)}$$

where  $a, b, \Sigma$  and  $\Sigma'$  are terms, and where  $\Sigma :: a$  is the concatenation of  $\Sigma$  and  $a$ , and  $\Sigma' :: b$  is the concatenation of  $\Sigma'$  and  $b$ .  $\Sigma$  and  $\Sigma'$  are sequences of labels, and  $a$  and  $b$  are labels representing worlds of an event model.

## 5 Extension to other semantics

In this section, we investigate the complexity of the satisfiability problem with semantics where the accessibility relations are also assumed to be reflexive or serial (an accessibility relation  $R_a$  is *reflexive* when for all  $w \in W$ ,  $w \in R_a(w)$ , and *serial* when for all  $w \in W$ , there is  $u \in W$  such that  $u \in R_a(w)$ ).

For reflexivity, we add the following rule to the tableau method:

$$\frac{(\Sigma \varphi)}{(R_a \Sigma \Sigma)} T$$

where  $\Sigma$  can be  $\sigma$  or  $\sigma_i$  (reflexivity of product models stems from initial and event models being reflexive). This rule is sound and complete with respect to reflexive models.

For seriality, we add the following rule to the tableau method:

$$\frac{(\Sigma B_a \varphi)}{(R_a \Sigma \Sigma^+)(\Sigma^+ \varphi)} D$$

where  $\Sigma$  can be  $\sigma$ ,  $\sigma_i$  or  $\sigma \sigma_1 \dots \sigma_i$ . This rule is sound and complete with respect to serial models. There is no problem of termination because we add a successor to a node if, and only if there is a modal formula in it.

**Proposition 10** *The satisfiability problem when the relations are reflexive or serial is NEXPTIME-complete.*

For the membership, it suffices to notice that adding the rule for reflexivity of seriality to the tableau method does not modify the proof of NEXPTIME membership 8.

For the lower-bound, we reduce the satisfiability problem given in the proof of Lemma 9 to the satisfiability problems when the relations are reflexive and when the relations are serial. Given new fresh atomic propositions  $p_0, p_1, \dots$ , let us consider the two following translations:

$$\begin{array}{ll} tr_k(p) & = p & tr'_k(p') & = p'_k \\ tr_k(\perp) & = \perp & tr'_k(\perp) & = \perp \\ tr_k(B_j \psi) & = B_j(p_{k+1} \rightarrow tr_{k+1}(\psi)) & tr'_k(B_j \psi) & = B_j(p'_{k+1} \rightarrow tr_{k+1}(\psi)) \end{array}$$

where  $\text{Pre}(p'_k) = p_k$  for all  $k$ .

We define  $tr(\varphi) = p_0 \wedge tr_0(\varphi)$  and  $tr'(\varphi') = p'_0 \wedge tr'_0(\varphi')$ . The three following statements are equivalent:

1.  $\varphi_0, \varphi', \varphi_1$  is satisfiable
2.  $tr(\varphi_0), tr'(\varphi'), tr(\varphi_1)$  is satisfiable in reflexive models
3.  $tr(\varphi_0), tr'(\varphi'), tr(\varphi_1)$  is satisfiable in serial models.

(i)  $\rightarrow$  (ii): If  $\varphi_0, \varphi', \varphi_1$  is satisfiable, then  $\varphi_0, \varphi', \varphi_1$  is satisfiable in models that are trees (indeed, the tableau method is sound and complete and the corresponding models are trees). So there exists a tree-model  $\mathcal{M} = (W, R, V)$ , a world  $w \in W$ , and a tree event model  $\mathcal{M}' = (W', R', V')$  and an event  $w' \in W'$  such that  $\mathcal{M}, w \models \varphi_0$ ,  $\mathcal{M}', w' \models \varphi'$  and  $\mathcal{M} \otimes \mathcal{M}', (w, w') \models \varphi_1$ . We add reflexive edges at each worlds in  $\mathcal{M}$  and  $\mathcal{M}'$ . We also extend the valuations so that  $p_k$  is true in a world  $u$  of  $\mathcal{M}$  iff the depth of  $u$  in the tree-model  $\mathcal{M}$  is equal to  $k$ . Let  $\mathcal{M}^* = (W^*, R^*, V^*)$  and  $\mathcal{M}'^* = (W'^*, R'^*, V'^*)$  be the resulting models. We have:

- $R_a = R_a^* \cap \{(u, v) \mid \text{for all } k, \text{ if } u \in V(p_k) \text{ then } u \in V(p_{k+1})\}$
- and  $R' = R'^* \cap \{(u', v') \mid \text{for all } k, \text{ if } u' \in V'(p'_k) \text{ then } v' \in V'(p'_{k+1})\}$ .

We have  $\mathcal{M}^*, w \models tr(\varphi_0)$ ,  $\mathcal{M}'^*, w' \models tr'(\varphi')$  and  $\mathcal{M}^* \otimes \mathcal{M}'^*, (w, w') \models tr(\varphi_1)$ .

(ii)  $\rightarrow$  (iii): This direction is trivial.

(iii)  $\rightarrow$  (i): Suppose that there exist a serial epistemic model  $\mathcal{M}^* = (W, R^*, V)$ ,  $w \in W$ , a serial event model  $\mathcal{M}'^* = (W', R'^*, V')$  and  $w' \in W'$  such that  $\mathcal{M}^*, w \models tr(\varphi_0)$ ,  $\mathcal{M}'^*, w' \models tr'(\varphi')$ ,  $\mathcal{M}^* \otimes \mathcal{M}'^*, (w, w') \models tr(\varphi_1)$  and  $\mathcal{M}^* \otimes \mathcal{M}'^*$  is serial. We define the following new model  $\mathcal{M} = (W, R, V)$  in the following way:

- $R_a := R_a^* \cap \{(u, v) \mid \text{for all } k, \text{ if } u \in V(p_k) \text{ then } v \in V(p_{k+1})\}$ .

and  $\mathcal{M}' = (W', R', V')$  in the following way:

- $R' := R'^* \cap \{(u', v') \mid \text{for all } k, \text{ if } u' \in V'(p'_k) \text{ then } v' \in V'(p'_{k+1})\}$ .

We have  $\mathcal{M}, w \models \varphi_0$ ,  $\mathcal{M}', w' \models \varphi'$  and  $\mathcal{M} \otimes \mathcal{M}', (w, w') \models \varphi_1$ .

## 6 Example: coordinated attack problem

In this section, we assume that there are only two agents  $a$  and  $b$ . We extend our epistemic language  $\mathcal{L}$  with the common knowledge operator  $C_{a,b}$ . The truth conditions of the common knowledge operator are defined as follows:

$$\mathcal{M}, w \models C_{a,b}\varphi \quad \text{iff for all } v \in (R_a \cup R_b)^+(w), \mathcal{M}, v \models \varphi,$$

where  $(R_a \cup R_b)^+$  is the transitive closure of  $(R_a \cup R_b)$ .

Intuitively,  $C_{a,b}\varphi$  is an abbreviation of an infinite conjunction (see [13] for more details):  $C_{a,b}\varphi = E_{a,b}\varphi \wedge E_{a,b}^2\varphi \wedge E_{a,b}^3\varphi \wedge \dots$ , where  $E_{a,b}^k\varphi$  is defined inductively as follows:  $E_{a,b}^1\varphi = B_a\varphi \wedge B_b\varphi$  and  $E_{a,b}^{k+1}\varphi = B_aE^k\varphi \wedge B_bE^k\varphi$ . The definitions of DEL-sequents of Definition 1 can easily be adapted to this extended language with common knowledge.

The coordinated attack problem from the distributed systems folklore can be described informally as follows. Two generals need to attack their common enemy simultaneously if they want to win. However, their only way to communicate is by means of a messenger, and this messenger may be captured at any time between the two camps. If we assume that the messenger is really lucky and never gets caught on that particular night, how long will it take for the generals to coordinate an attack?

We can model this problem within our framework. Assume that general  $a$  has decided to attack at dawn. General  $a$  then sends a messenger to general  $b$  to inform him of his decision. The content of the first message sent by general  $a$  to general  $b$  is represented by the propositional letter *attack* standing for ‘general  $a$  has decided to attack at dawn’. This message eventually reaches general  $b$ , but general  $a$  does not know it yet. This event is represented by an atomic event  $p'_1$  standing for ‘general  $b$  receives the decision of general  $a$  to attack at dawn’. Its precondition is  $\text{Pre}(p'_1) = \textit{attack}$ . The only information we have about this event  $p'_1$  is that general  $b$  knows about its occurrence:  $B_b p'_1$ . As a result of this event, general  $b$  now knows that general  $a$  has decided to attack:  $B_b \textit{attack}$ . However, general  $a$  does not know it, so they still cannot coordinate a simultaneous attack. Therefore, general  $b$  sends an acknowledgement to general  $a$ . This message eventually reaches general  $a$ . This event is represented by the atomic event  $p'_2$  standing for ‘general  $a$  receives the first acknowledgement of general  $b$ ’. The precondition of atomic event  $p'_2$  is  $\text{Pre}(p'_2) = B_b \textit{attack}$ . This time, general  $b$  does not know that his message has been delivered. Therefore, the only information we have about this event  $p'_2$  is that general  $a$  knows about its occurrence:  $B_a p'_2$ . As a result of this event, general  $a$  now knows that general  $b$  knows that general  $a$  has decided to attack:  $B_a B_b \textit{attack}$ . However, there is still no common knowledge that general  $a$  has decided to attack. This informal reasoning could go on indefinitely. It shows that common knowledge that general  $a$  has decided to attack cannot be attained.

The above informal reasoning can be formalized within our framework in a natural way. To achieve this aim, we define for all  $k \in \mathbb{N}^*$  the atomic event  $p'_k$  whose precondition is  $\text{Pre}(p'_k) = B_a B_b \dots B_a B_b \textit{attack}$  if  $k$  is odd and  $\text{Pre}(p'_k) = B_b B_a B_b \dots B_a B_b \textit{attack}$  if  $k$  is even ( $k - 1$  knowledge operators are nested alternatively in  $\text{Pre}(p'_k)$ ). The statement that after any

finite number of messages exchanged, it is impossible to infer that there is common knowledge that general  $a$  has decided to attack is formalized by the fact that, for all  $k \in \mathbb{N}^*$ :

$$\begin{aligned} B_a attack, B_b p'_1, \top, B_a p'_2, \dots, B_a p'_{k-1}, \top, B_b p'_k, \top &\stackrel{\perp}{\Vdash} C_{a,b} attack && \text{if } k \text{ is odd} \\ B_a attack, B_b p'_1, \top, B_a p'_2, \dots, B_b p'_{k-1}, \top, B_a p'_k, \top &\stackrel{\perp}{\Vdash} C_{a,b} attack && \text{if } k \text{ is even} \end{aligned}$$

This result is itself due to the following fact:

**Proposition 11** For all  $k \in \mathbb{N}^*$ ,

$$\begin{aligned} B_a attack, B_b p'_1, \top, B_a p'_2, \dots, B_b p'_k, \top &\stackrel{\perp}{\Vdash} E_{a,b}^{k+1} attack && \text{if } k \text{ is odd} \\ B_a attack, B_b p'_1, \top, B_a p'_2, \dots, B_a p'_k, \top &\stackrel{\perp}{\Vdash} E_{a,b}^{k+1} attack && \text{if } k \text{ is even} \end{aligned}$$

**Proof** To prove the result of the coordinated attack problem, we define the following epistemic model  $\mathcal{M} = (W, R, V)$  by:

- $W = \{w, u\}$ ;
- $R_a = \{(w, w), (u, u)\}$ ;  $R_b = \{(w, w), (w, u), (u, u)\}$ ;
- $V(attack) = \{w\}$ .

We define the following event models  $\mathcal{M}^k = (W^k, R^k, V^k)$  for all  $k \in \mathbb{N}$ . In the DEL literature, these event models represent private announcements. If  $k$  is even,  $(\mathcal{M}^k, w^k)$  represents the private announcement to general  $a$  that  $B_b B_a B_b \dots B_a B_b attack$  holds, and if  $k$  is odd,  $(\mathcal{M}^k, w^k)$  represents the private announcement to general  $b$  that  $B_a B_b \dots B_a B_b attack$  holds.

- $W^k = \{w^k, u^k\}$ ;
- $R_a^k = \{(w^k, w^k), (u^k, u^k)\}$ ;  $R_b^k = \{(w^k, w^k), (w^k, u^k), (u^k, u^k)\}$  if  $k$  is even;
- $R_b^k = \{(w^k, w^k), (u^k, u^k)\}$ ;  $R_a^k = \{(w^k, w^k), (w^k, u^k), (u^k, u^k)\}$  if  $k$  is odd;
- $V^k(p'_k) = \{w^k\}$ .
- $V^k(p') = \{u^k\}$  where  $\text{Pre}(p') = \top$ .

We then have that  $\mathcal{M}, w \models B_a attack$ ,  $\mathcal{M}^1, w^1 \models B_b p'_1$ ,  $\mathcal{M}^2, w^2 \models B_a p'_2$ ,  $\dots$ ,  $\mathcal{M}^k, w^k \models B_b p'_k$  and  $(\mathcal{M}, w) \otimes (\mathcal{M}^1, w^1) \otimes \dots \otimes (\mathcal{M}^k, w^k) \models \langle B_b \rangle \dots \langle B_a \rangle \langle B_b \rangle \neg attack$ .  $\square$

We illustrate our tableau method by proving this fact for  $k = 1$  in Figure 1. Remark that  $E_{a,b}^2 attack = B_a (B_a attack \wedge B_b attack) \wedge B_b (B_a attack \wedge B_b attack)$ . Also, we only show one branch of the full tableau, the one which is open.

## 7 Conclusion

### 7.1 Related work

In [12], Dupin de Saint-Cyr and Lang define an operator of “extrapolation”. This operator takes as input a temporal formula, which corresponds to a sequence of observations under the form of propositional formulas indexed by time stamps, and yields as output another temporal formula, which corresponds semantically to the *preferred* sequences of states which satisfy the input temporal formula. The authors follow an internal approach. In [9], Booth and Nittka

$$\begin{aligned}
\Gamma_0 &= \{(\sigma \ B_a \text{attack}), (\sigma_1 \ B_b p'_1), (\sigma_1 \ p_1^+), (\sigma \ \sigma_1 \ \neg E_{a,b}^2)\} \\
&\quad \left( \text{Pre}_1 \text{ and } \neg \wedge \right. \\
\Gamma_1 &= \Gamma_0 \cup \{(\sigma \ p_1^+)\} \cup \{(\sigma \ \sigma_1 \ \langle B_a \rangle \neg (B_a \text{attack} \wedge B_b \text{attack}))\} \\
&\quad \left( \langle B_a \rangle \right. \\
\Gamma_2 &= \Gamma_1 \cup \{(R_a \ \sigma \ \sigma'), (R_a^1 \ \sigma_1 \ \sigma'_1), (\sigma' \ \sigma'_1 \ \neg (B_a \text{attack} \wedge B_b \text{attack})), (\sigma'_1 \ p_2^+)\} \\
&\quad \left( \text{Pre}_1, B_a \text{ and } \neg \wedge \right. \\
\Gamma_3 &= \Gamma_2 \cup \{(\sigma' \ p_2^+)\} \cup \{(\sigma' \ \text{attack})\} \cup \{(\sigma' \ \sigma'_1 \ \langle B_b \rangle \neg \text{attack})\} \\
&\quad \left( \langle B_b \rangle \right. \\
\Gamma_4 &= \Gamma_3 \cup \{(R_b \ \sigma' \ \sigma''), (R_b^1 \ \sigma'_1 \ \sigma''_1), (\sigma'' \ \sigma''_1 \ \neg \text{attack}), (\sigma''_1 \ p_3^+)\} \\
&\quad \left( \text{Pre}_1 \text{ and } \leftarrow_1 \right. \\
\Gamma_5 &= \Gamma_4 \cup \{(\sigma'' \ p_3^+)\} \cup \{(\sigma \ \neg \text{attack})\}
\end{aligned}$$

Figure 1: Open branch of the tableau proving that  $B_a \text{attack}, B_b p'_1, \top \not\vdash E_{a,b}^2 \text{attack}$

address a similar problem but with an imperfect external approach (see [1] for more details on the different modeling approaches). They are interested in inferring what the agent believed (or will believe) at a given moment, based on a sequence of observations consisting of responses that the agent has given to a sequence of belief revision inputs. Both papers deal with situations involving a single agent. Our approach is different from both approaches, because we do not strive to “extrapolate” new information from existing observations by resorting to an argument of minimal change. Instead, we are only interested in inferring some *necessary* property that follows from these existing observations.

Some tableau methods have been proposed for DEL, but only for public announcement logic [4, 11] and hybrid public announcement logic [14]. A terminating tableau method has also been proposed for the full BMS framework in [14] by encoding the reduction axioms as tableau rules. However, none of these tableau methods can somehow address the two problems raised in the introduction, because the BMS language of [6] does not allow for partial and incomplete descriptions of events: an event model or a formula announced publicly specifies *completely* how all the agents perceive the occurrence of the corresponding event. In particular, it is impossible to model the coordinated attack problem.

In [5], a sequent calculus has been developed, yet in an algebraic setting, making a systematic comparison difficult. Their sequents  $m_1, \dots, q_1, \dots, A_1, \dots, m_k, \dots, q_l, \dots, A_n \vdash \delta$  are arbitrarily long and consist of different types of formulas which can contain propositions  $m_1, \dots, m_k$ , events  $q_1, \dots, q_l$  and agents  $A_1, \dots, A_n$ , and which resolve into a single proposition *or* event  $\delta$ . A sequent calculus has also been proposed for public announcement logic [15] which, alike our tableau terms, refers explicitly to possible worlds and relations.

## 7.2 Concluding remarks

As we said in the previous subsection, our tableau method can infer necessary information which was somehow already encoded in the sequence. In fact, our method allows us to *verify* in a *complexity-optimal* way that a piece of information about a sequence of events does follow from this sequence. It also allows us to check that a given epistemic plan (viewed as a sequence of event properties) yields an epistemic goal. Even if it does not provide an algorithm to synthesize

it ([2] deals more with synthesis), it can still be instrumental in finding out some of its necessary properties. It can also be used during the design of the epistemic plan itself to check that we are designing it in the ‘right’ direction.

As the title of this paper suggests, we have generalized our previous work [3] to arbitrary long sequences of events, and extended it to the case of reflexive or serial models. Also we illustrate our method by encoding the coordinated attack problem. However, other generalizations still need to be done, such as the addition of a common knowledge operator in the language (as illustrated in Section 6) and the integration of ontic events.

## References

- [1] Guillaume Aucher. An internal version of epistemic logic. *Studia Logica*, 94(1):1–22, 2010.
- [2] Guillaume Aucher. DEL-sequents for regression and epistemic planning. *Journal of Applied Non-Classical Logics*, 2012. to appear.
- [3] Guillaume Aucher, Bastien Maubert, and François Schwarzentruber. Tableau method and NEXPTIME-completeness of DEL-sequents. *Electronic Notes in Theoretical Computer Science*, 278:17–30, 2011.
- [4] Philippe Balbiani, Hans van Ditmarsch, Andreas Herzig, and Tiago de Lima. Tableaux for public announcement logic. *Journal of Logic and Computation*, 20(1):55–76, 2010.
- [5] Alexandru Baltag, Bob Coecke, and Mehrnoosh Sadrzadeh. Algebra and sequent calculus for epistemic actions. In *Proceedings of Workshop on Logic and Communication in Multi-Agent Systems (LCMAS’04)*, pages 60–78, 2004.
- [6] Alexandru Baltag and Larry Moss. Logic for epistemic programs. *Synthese*, 139(2):165–224, 2004.
- [7] Alexandru Baltag, Larry Moss, and Slawomir Solecki. The logic of common knowledge, public announcement, and private suspicions. In I. Gilboa, editor, *Proceedings of the 7th conference on theoretical aspects of rationality and knowledge (TARK98)*, pages 43–56, 1998.
- [8] Alexandru Baltag, Larry Moss, and Slawomir Solecki. The logic of public announcements, common knowledge and private suspicions. Technical report, Indiana University, 1999.
- [9] Richard Booth and Alexander Nittka. Reconstructing an agent’s epistemic state from observations about its beliefs and non-beliefs. *J. Log. Comput.*, 18(5):755–782, 2008.
- [10] Vittorio Brusoni, Luca Console, Paolo Terenziani, and Daniele Theseider Dupré. A spectrum of definitions for temporal model-based diagnosis. *Artificial Intelligence*, 102(1):39–79, 1998.
- [11] Mathijs de Boer. KE tableaux for public announcement logic. In *Proceedings of Formal Approaches to Multi-Agent Systems Workshop (FAMAS 07)*, Durham UK, 2007.
- [12] Florence Dupin de Saint-Cyr and Jérôme Lang. Belief extrapolation (or how to reason about observations and unpredicted change). *Artif. Intell.*, 175(2):760–790, 2011.
- [13] Ronald Fagin, Joseph Halpern, Yoram Moses, and Moshe Vardi. *Reasoning about knowledge*. MIT Press, 1995.
- [14] Jens Ulrik Hansen. Terminating tableaux for dynamic epistemic logic. *Electronic Notes in Theoretical Computer Science*, 262:141–156, 2010.

- [15] Paolo Maffezioli and Sara Negri. A gentzen-style analysis of public announcement logic. In *Proceedings of the International Workshop on Logic and Philosophy of Knowledge, Communication and Action*, pages 293–313, 2010.
- [16] François Schwarzentruber. Lotrecscheme. *Electronic Notes in Theoretical Computer Science*, 278:187–199, 2011.



**RESEARCH CENTRE  
RENNES – BRETAGNE ATLANTIQUE**

Campus universitaire de Beaulieu  
35042 Rennes Cedex

Publisher  
Inria  
Domaine de Voluceau - Rocquencourt  
BP 105 - 78153 Le Chesnay Cedex  
[inria.fr](http://inria.fr)

ISSN 0249-6399