Generalized DEL-sequents

Guillaume Aucher[1], Bastien Maubert[2], and François Schwarzentruber[3]

[1] University of Rennes 1 - INRIA, France,
guillaume.aucher@irisa.fr
[2] University of Rennes 1, France,
bastien.maubert@irisa.fr
[3] ENS Cachan, France,
francois.schwarzentruber@bretagne.ens-cachan.fr

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.

1 Introduction

Assume that a sequence of n events has occured 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 φ'_i and φ_i respectively:

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

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 ψ describing (incompletely) a situation, we want to verify whether or not we can infer that this property ψ necessarily held at the i^{th} resulting situation:

$$\varphi_0, \varphi'_1, \varphi_1, \ldots, \varphi'_n, \varphi_n \models^1_i \psi \ ?$$

In the second kind of inference problem, we want to infer more information about the i^{th} event. That is, given a formula ψ' describing (incompletely) this

event, we want to verify whether or not we can infer that this property ψ' necessarily held during the occurrence of the i^{th} event:

$$\varphi_0, \varphi'_1, \varphi_1, \ldots, \varphi'_n, \varphi_n \stackrel{2}{\models_i} \psi' \ ?$$

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 φ'_i and φ_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.[4]

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, Φ is a countably infinite set of propositional letters called *atomic facts* which describe static situations, and Agt is a finite set of agents.

[4] Note that all the proofs of this paper are available in a Technical Report at the following address: `http://hal.inria.fr/hal-00716074`.

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

- W is a non-empty set of possible worlds,
- $R : Agt \to 2^{W \times W}$ is a function assigning to each agent $a \in Agt$ an accessibility relation on W,
- $V : \Phi \to 2^W$ is a function assigning to each propositional letter of Φ 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_av for $R(a)(w, v)$ and $R_a(w) = \{v \in W \mid wR_av\}$. Intuitively, wR_av 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:

$$\mathcal{L} : \varphi \quad ::= \quad p \quad | \quad \neg\varphi \quad | \quad \varphi \wedge \varphi \quad | \quad B_a\varphi$$

where p ranges over Φ 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:

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

The formula $B_a\varphi$ reads as "agent a believes φ". Its truth conditions are defined in such a way that agent a believes φ holds in a possible world when φ holds in all the worlds agent a considers possible. Dually, the formula $\langle B_a\rangle\varphi$ reads as "agent a considers φ is plausible". Agent a considers that φ is plausible in a possible world when φ 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 Φ'. 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 $\text{Pre} : \Phi' \to \mathcal{L}$ 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,
- $R' : Agt \to 2^{W' \times W'}$ is a function assigning to each agent $a \in Agt$ an accessibility relation on W',

– $V' : \Phi' \to 2^{W'}$ is a function assigning to each propositional letter of Φ' 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].[5]

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]):

$$\mathcal{L}' : \varphi' \quad ::= \quad p' \quad | \quad \neg\varphi' \quad | \quad \varphi' \wedge \varphi' \quad | \quad B_a\varphi'$$

where p' ranges over Φ' and a over 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:

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

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

The precondition function 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 $\mathrm{Pre}(w')$ of $\mathcal{L}$. The precondition function induced by the $\mathcal{L}'$-model $\mathcal{M}' = (W', R', V')$ is defined as follows: $\mathrm{Pre}(w') = \mathrm{Pre}(p')$ if there is p' such that $\mathcal{M}', w' \models p'$; $\mathrm{Pre}(w') = \top$ 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 $\mathcal{M}, w \models \mathrm{Pre}(w')$. 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:

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

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

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:

Definition 1. *Let $\varphi_0, \varphi_1, \ldots, \varphi_n, \psi \in \mathcal{L}$, $\varphi'_1, \ldots, \varphi'_n, \psi' \in \mathcal{L}'$. We define the logical consequence relations $\varphi_0, \varphi'_1, \varphi_1, \ldots, \varphi'_n, \varphi_n \mid\!\!\overset{1}{=}_k \psi$ where $0 \leq k \leq n$, and $\varphi_0, \varphi'_1, \varphi_1, \ldots, \varphi'_n, \varphi_n \mid\!\!\overset{2}{=}_k \psi'$ where $1 \leq k \leq n$, as follows:*

$$\varphi_0, \varphi'_1, \varphi_1, \ldots, \varphi'_n, \varphi_n \mid\!\!\overset{1}{=}_k \psi \text{ if}$$

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 $i \in \{0, \ldots, n\}$ that $(\mathcal{M}_i, w_i) = (\mathcal{M}_0, w_0) \otimes (\mathcal{M}'_1, w'_1) \otimes \ldots \otimes (\mathcal{M}'_i, w'_i)$ is defined[6] *and $\mathcal{M}_i, w_i \models \varphi_i$, and for all $j \in \{1, \ldots, n\}$ that $\mathcal{M}'_j, w'_j \models \varphi'_j$, then $\mathcal{M}_k, w_k \models \psi$.*

$$\varphi_0, \varphi'_1, \varphi_1, \ldots, \varphi'_n, \varphi_n \mid\!\!\overset{2}{=}_k \psi' \text{ if}$$

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 $i \in \{0, \ldots, n\}$ that $(\mathcal{M}_i, w_i) = (\mathcal{M}_0, w_0) \otimes (\mathcal{M}'_1, w'_1) \otimes \ldots \otimes (\mathcal{M}'_i, w'_i)$ is defined and $\mathcal{M}_i, w_i \models \varphi_i$, and for all $j \in \{1, \ldots, n\}$ 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 $\varphi_0, \ldots, \varphi_n \in \mathcal{L}$, $\varphi'_1, \ldots, \varphi'_n, \psi' \in \mathcal{L}'$ and $k \in \{1, \ldots, n\}$,*

$$\varphi_0, \varphi'_1, \varphi_1, \ldots, \varphi'_n, \varphi_n \mid\!\!\overset{2}{=}_k \psi' \text{ iff } \varphi_0, \ldots, \varphi'_k \wedge \neg\psi', \top, \varphi'_{k+1}, \ldots, \varphi_n \mid\!\!\overset{1}{=}_k \neg\varphi_k$$

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

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

$$\varphi_0, \varphi'_1, \varphi_1, \ldots, \varphi'_n, \varphi_n \mid\!\!\overset{1}{=}_k \psi \text{ iff } \varphi_0, \ldots, \varphi_k \wedge \neg\psi, \ldots, \varphi'_n, \top \mid\!\!\overset{1}{=}_n \neg\varphi_n$$

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

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:

$$\varphi_0, \varphi' \models \varphi_1 \text{ iff } \varphi_0, \varphi', \top \mid\!\!\overset{1}{=}_1 \varphi_1 \qquad \varphi', \varphi_1 \mid\!\!\overset{3}{=} \varphi_0 \text{ iff } \top, \varphi', \varphi_1 \mid\!\!\overset{1}{=}_0 \varphi_0$$
$$\varphi_0, \varphi_1 \mid\!\!\overset{2}{=} \varphi' \text{ iff } \varphi_0, \top, \varphi_1 \mid\!\!\overset{2}{=}_1 \varphi'$$

[6] When $i = 0$ we let $(\mathcal{M}_0, w_0) \otimes (\mathcal{M}'_1, w'_1) \otimes \ldots \otimes (\mathcal{M}'_i, w'_i) = (\mathcal{M}_0, w_0)$.

3 Tableau method

We consider $2n+2$ formulas, $\varphi_0, \ldots, \varphi_n, \psi \in \mathcal{L}$ and $\varphi'_1, \ldots, \varphi'_n \in \mathcal{L}'$, and by $\mathcal{P} \subset \Phi'$ we denote the finite set of all atomic events appearing in one of the event formulas $\varphi'_1, \ldots, \varphi'_n$. i ranges over $\{0, \ldots, n\}$ and j ranges over $\{1, \ldots, n\}$. We want to address the problem of deciding whether $\varphi_0, \varphi'_1, \varphi_1, \ldots, \varphi'_n, \varphi_n \models\!\!\!\models^{\frac{1}{n}} \psi$ 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, $(\mathcal{M}_i, w_i) = (\mathcal{M}_0, w_0) \otimes (\mathcal{M}'_1, w'_1) \otimes \ldots \otimes (\mathcal{M}'_i, w'_i)$ 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 $\varphi_0, \varphi'_1, \varphi_1, \ldots, \varphi'_n, \varphi_n \models\!\!\!\models^{\frac{1}{n}} \psi$ holds reduces to the following problem called the *satisfiability problem*:

- Input: $\varphi_0, \ldots, \varphi_n, \psi \in \mathcal{L}$, $\varphi'_1, \ldots, \varphi'_n \in \mathcal{L}'$ and $\mathrm{Pre}_{|\mathcal{P}}$, the restriction of 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, $(\mathcal{M}_i, w_i) = (\mathcal{M}_0, w_0) \otimes (\mathcal{M}'_1, w'_1) \otimes \ldots \otimes (\mathcal{M}'_i, w'_i)$ 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 $\mathcal{M}_0 \otimes \mathcal{M}'_1 \otimes \ldots \otimes \mathcal{M}'_i$ and w_i for $(w_0, w'_1, \ldots, w'_i)$.

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^{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 σ and φ 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^{th} event model $\mathcal{M}'_i$ represented by σ_i, and φ' is a formula of $\mathcal{L}'$;
- $(\sigma\ \sigma_1 \ldots\ \sigma_i\ \varphi)$ means that $\mathcal{M}_i, (w_0, w'_1, \ldots, w'_i) \models \varphi$, where w_0 is the world of $\mathcal{M}_0$ represented by σ, $w'_k \in \mathcal{M}'_k$ is the event represented by σ_k, and φ is a formula of $\mathcal{L}$;
- $(\sigma\ \sigma_1 \ldots\ \sigma_i\ 0)$ means that $(w_0, w'_1, \ldots, w'_i)$ is not in $\mathcal{M}_i$, where w_0 is the world of $\mathcal{M}_0$ represented by σ and $w'_k \in \mathcal{M}'_k$ is the event represented by σ_k;
- $(R_a\ \sigma\ \sigma')$ means that the worlds w and u of $\mathcal{M}_0$ represented respectively by σ and σ' are such that wR_au;
- $(R^i_a\ \sigma_i\ \sigma'_i)$ means that in $\mathcal{M}'_i$, the events w' and u' represented by σ_i and σ'_i are such that $w'R'^i_au'$;
- $\perp$ denotes an inconsistency.

We also use *generic labels* Σ to simplify notations: Σ can be either σ, σ_i or $\sigma\ \sigma_1\ \ldots\ \sigma_i$. $(R_a\ \Sigma\ \Sigma')$ 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 $(R^i_a\ \sigma_i\ \sigma'_i)$, and if $\Sigma = \sigma\ \sigma_1\ \ldots\ \sigma_i$, $\Sigma' = \sigma'\ \sigma'_1\ \ldots\ \sigma'_i$ then $(R_a\ \Sigma\ \Sigma')$ denotes $(R_a\ \sigma\ \sigma')(R^1_a\ \sigma_1\ \sigma_1)\ldots(R^i_a\ \sigma_i\ \sigma_i)$.

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

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

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

A *tableau for* a tuple $(\varphi_0, \ldots, \varphi_n, \varphi'_1, \ldots, \varphi'_n)$ of formulas is a finite tree with a set of tableau terms at each node. The root is $\Gamma^0 = \{(\sigma\ \ \varphi_0), (\sigma\ \ \sigma_1\ \ \varphi_1), \ldots, (\sigma\ \ \sigma_1\ \ \ldots\ \ \sigma_n\ \ \varphi_n), (\sigma_1\ \ \varphi'_1), \ldots, (\sigma_n\ \ \varphi'_n)\} \cup \{(\sigma_1\ \ p'^{+}_1), \ldots, (\sigma_n\ \ p'^{+}_n)\}$, where $p'^{+}_1, \ldots, p'^{+}_n$ are fresh new atomic events. Indeed, in our tableau method, when a new event label σ_i is added to the set Γ 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 φ' in Γ, 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 Γ. We denote it by $\mathrm{Pre}(p'^{+}) = p^{+}$. We abuse notations by writing $p' \in \Gamma$ whenever p' occurs in a formula appearing in a tableau term in Γ. 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 Γ if Γ contains an instance of $\mathcal{N}$. If no rule is applicable, Γ is said to be *saturated*. We call a node n an *end node* if the set of formulas Γ it carries is saturated, or if $\bot \in \Gamma$. The tableau is extended as follows:

1. Choose a leaf node n carrying Γ where n is not an end node, and choose a rule ρ applicable to n.
2. (a) If ρ has only one denominator, add the appropriate instanciation to Γ.
   (b) If ρ has k denominators with $k > 1$, create k successor nodes for n, where each successor i carries the union of Γ with an appropriate instanciation 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 $\bot$, otherwise it is *open*. A tableau is *closed* if all its branches are closed, otherwise it is *open*. We write $\varphi_0, \varphi'_1, \varphi_1, \ldots, \varphi'_n, \varphi_n \overset{1}{\vdash} \psi$ if there is a closed tableau for $(\varphi_0, \ldots, \varphi_{n-1}, \varphi_n \wedge \neg\psi, \varphi'_1, \ldots, \varphi'_n)$.

3.2 Tableau rules

– Common rules for epistemic formulas and event formulas:

$$\frac{(\Sigma\ \varphi \wedge \psi)}{(\Sigma\ \varphi)\ (\Sigma\ \psi)}\ \wedge \qquad \frac{(\Sigma\ \neg(\varphi \wedge \psi))}{(\Sigma\ \neg\varphi) \mid (\Sigma\ \neg\psi)}\ \neg\wedge \qquad \frac{(\Sigma\ \neg\neg\varphi)}{(\Sigma\ \varphi)}\ \neg \qquad \frac{(\Sigma\ p)(\Sigma\ \neg p)}{\bot}\ \bot$$

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

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

– Specific rule for event formulas:

$$\frac{(\sigma_i\ p')(\sigma_i\ p'')}{\bot}\ \mathrm{Excl}$$

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

– Specific rules for epistemic formulas:

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

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

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

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_1 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 $\mathrm{Pre}(p')$. Rule Pre_2 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 $\mathcal{M}_{i-1}, (w_0, w_1', \dots, w_{i-1}') \not\models \mathrm{Pre}(p')$. Finally, Rule Pre_0 is used to forbid the rightmost choice in Rule Pre_2 when $i = 1$. Indeed it would make no sense to say that the world associated to a label σ 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 and completeness). *For all* $\varphi_0, \dots, \varphi_n, \psi \in \mathcal{L}$, *for all* $\varphi_1', \dots, \varphi_n' \in \mathcal{L}'$, $\varphi_0, \varphi_1', \varphi_1, \dots, \varphi_n', \varphi_n \vdash^1 \psi$ *iff* $\varphi_0, \varphi_1', \varphi_1, \dots, \varphi_n', \varphi_n \models^1_n \psi$.

4 NEXPTIME-completeness and implementation

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

Proposition 4. *The satisfiability problem is NEXPTIME-complete*

An implementation of the tableau method can be found at:

`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:

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

where a, b, Σ and Σ' are terms, and where $\Sigma :: a$ is the concatenation of Σ and a, and $\Sigma' :: b$ is the concatenation of Σ' and b. Σ and Σ' 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 $u \in R_a(w)$).

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

For reflexivity, we add the rule T above to the tableau method, where Σ can be σ or σ_i (reflexivity of product models stems from initial and event models being reflexive). Rule T is sound and complete with respect to reflexive models.

For seriality, we add the rule D above to the tableau method, where Σ can be σ, σ_i or $\sigma\ \sigma_1\ \ldots\ \sigma_i$. Rule D 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 5. *The satisfiability problem when the relations are reflexive or serial is NEXPTIME-complete.*

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 $C_{a,b}\varphi$. The truth conditions of the common knowledge operator are defined as follows:

$\mathcal{M}, w \models C_{a,b}\varphi$ 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): $C_{a,b}\varphi = E_{a,b}\varphi \wedge E^2_{a,b}\varphi \wedge E^3_{a,b}\varphi \wedge \ldots$, where $E^k_{a,b}\varphi$ is defined inductively as follows: $E^1_{a,b}\varphi = B_a\varphi \wedge B_b\varphi$ and $E^{k+1}_{a,b}\varphi = B_aE^k\varphi \wedge B_bE^k\varphi$. 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 $\mathrm{Pre}(p'_1) = attack$. The only information we have about this event p'_1 is that general b knows about its occurrence: $B_bp'_1$. As a result of this event, general b now knows that general a has decided to attack: $B_battack$. 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 $\mathrm{Pre}(p'_2) = B_battack$. 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: $B_ap'_2$. As a result of this event, general a now knows that general b knows that general a has decided to attack: $B_aB_battack$. 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 $\mathrm{Pre}(p'_k) = B_a B_b \dots B_a B_b attack$ if k is odd and $\mathrm{Pre}(p'_k) = B_b B_a B_b \dots B_a B_b attack$ if k is even ($k-1$ knowledge operators are nested alternatively in $\mathrm{Pre}(p'_k)$). 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 $k \in \mathbb{N}^*$:

$$B_a attack, B_b p'_1, \top, B_a p'_2, \dots, B_a p'_{k-1}, \top, B_b p'_k, \top \not\vdash^{\downarrow}_{\mathcal{K}} C_{a,b} attack \quad \text{if } k \text{ is odd}$$
$$B_a attack, B_b p'_1, \top, B_a p'_2, \dots, B_b p'_{k-1}, \top, B_a p'_k, \top \not\vdash^{\downarrow}_{\mathcal{K}} C_{a,b} attack \quad \text{if } k \text{ is even}$$

This result is itself due to the fact that for all $k \in \mathbb{N}^*$, $B_a attack, B_b p'_1, \top, B_a p'_2, \dots, B_b p'_k, \top \not\vdash^{\downarrow}_{\mathcal{K}} E^{k+1}_{a,b} attack$ if k is odd, and $B_a attack, B_b p'_1, \top, B_a p'_2, \dots, B_a p'_k, \top \not\vdash^{\downarrow}_{\mathcal{K}} E^{k+1}_{a,b} attack$ if k is even.

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

$$\Gamma_0 = \{(\sigma\ B_a attack), (\sigma_1\ B_b p'_1), (\sigma_1\ p'^{+}_1), (\sigma\ \sigma_1\ \neg E^2_{a,b})\}$$
$$\Big\downarrow \quad \mathrm{Pre}_1 \text{ and } \neg\wedge$$
$$\Gamma_1 = \Gamma_0 \cup \{(\sigma\ p^+_1)\} \cup \{(\sigma\ \sigma_1\ \langle B_a\rangle \neg(B_a attack \wedge B_b attack))\}$$
$$\Big\downarrow \quad \langle B_a\rangle$$
$$\Gamma_2 = \Gamma_1 \cup \{(R_a\ \sigma\ \sigma'), (R^1_a\ \sigma_1\ \sigma'_1), (\sigma'\ \sigma'_1\ \neg(B_a attack \wedge B_b attack)), (\sigma'_1\ p'^{+}_2)\}$$
$$\Big\downarrow \quad \mathrm{Pre}_1,\ B_a \text{ and } \neg\wedge$$
$$\Gamma_3 = \Gamma_2 \cup \{(\sigma'\ p^+_2)\} \cup \{(\sigma'\ attack)\} \cup \{(\sigma'\ \sigma'_1\ \langle B_b\rangle \neg attack)\}$$
$$\Big\downarrow \quad \langle B_b\rangle$$
$$\Gamma_4 = \Gamma_3 \cup \{(R_b\ \sigma'\ \sigma''), (R^1_b\ \sigma'_1\ \sigma''_1), (\sigma''\ \sigma''_1\ \neg attack)), (\sigma''_1\ p'^{+}_3)\}$$
$$\Big\downarrow \quad \mathrm{Pre}_1 \text{ and } \leftarrow_1$$
$$\Gamma_5 = \Gamma_4 \cup \{(\sigma''\ p^+_3)\} \cup \{(\sigma\ \neg attack)\}$$

Fig. 1: Open branch of the tableau proving that $B_a attack, B_b p'_1, \top \not\vdash^{\downarrow}_{\mathcal{A}} E^2_{a,b} attack$

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 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 anouncement 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 $m_1, \ldots, q_1, \ldots, A_1, \ldots, m_k, \ldots, q_l, \ldots, A_n \vdash \delta$ are arbitrarily long and consist of different types of formulas which can contain propositions $m_1, \ldots, m_k$, events $q_1, \ldots, q_l$ and agents $A_1, \ldots, A_n$, and which resolve into a single proposition *or* event δ. A sequent calculus has also been proposed for public announcement logic [15] which, alike our tableau terms, refers explicitely 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.[7] Also we illustrate our method by encoding the coordinated

[7] A tableau rule is missing in [3]. This error is corrected in the paper available on the HAL archive at `http://hal.inria.fr/docs/00/64/64/81/PDF/M4M11.pdf`

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 anouncement 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.