

Tableau method and NEXPTIME-Completeness of DEL-Sequents

Guillaume Aucher, Bastien Maubert, François Schwarzenruber

► **To cite this version:**

Guillaume Aucher, Bastien Maubert, François Schwarzenruber. Tableau method and NEXPTIME-Completeness of DEL-Sequents. Methods for Modalities (M4M), Nov 2011, Sevilla, Spain. inria-00627642v2

HAL Id: inria-00627642

<https://hal.inria.fr/inria-00627642v2>

Submitted on 30 Nov 2011

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.

Tableau Method and NEXPTIME-Completeness of DEL-Sequents¹

Guillaume Aucher^{2,3}

*IRISA
INRIA - University of Rennes 1
Rennes, France*

Bastien Maubert⁴

*IRISA
ENS Cachan
Rennes, France*

François Schwarzentruber⁵

*IRISA
ENS Cachan
Rennes, France*

Abstract

Dynamic Epistemic Logic (DEL) deals with the representation of situations in a multi-agent and dynamic setting. It can express in a uniform way statements about:

- (i) what is true about an initial situation
- (ii) what is true about an event occurring in this situation
- (iii) what is true about the resulting situation after the event has occurred.

After proving that what we can infer about (ii) given (i) and (iii) and what we can infer about (i) given (ii) and (iii) are both reducible to what we can infer about (iii) given (i) and (ii), we provide a tableau method deciding whether such an inference is valid. We implement it in LOTRECscheme and show that this decision problem is NEXPTIME-complete. This contributes to the proof theory and the study of the computational complexity of DEL which have rather been neglected so far.

Keywords: Dynamic epistemic logic, tableau method, computational complexity

¹ This paper corrects the paper published under the same name and with the same authors in the proceedings of M4M 2011. The rule \perp'' of the tableau method was missing.

² We thank Sophie Pinchinat for helpful discussions and three reviewers for comments.

³ Email: guillaume.aucher@irisa.fr

⁴ Email: bastien.maubert@irisa.fr

⁵ Email: francois.schwarzentruber@bretagne.ens-cachan.fr

1 Introduction

Dynamic Epistemic Logic (DEL) deals with the logical study in a multi-agent setting of knowledge and belief change, and more generally of information change [van Ditmarsch et al., 2007]. To account for these logical dynamics, the core idea of DEL is to split the task of representing the agents' beliefs into three parts: first, one represents their beliefs about an initial situation; second, one represents their beliefs about an event taking place in this situation; third, one represents the way the agents update their beliefs about the situation after (or during) the occurrence of the event. Consequently, one can express uniformly within the logical framework of DEL epistemic statements about:

- (i) what is true about an initial situation,
- (ii) what is true about an event occurring in this situation,
- (iii) what is true about the resulting situation after the event has occurred.

From a logical point of view, this trichotomy begs the following three questions. Given (i) and (ii), what can we infer about (iii)? Given (i) and (iii), what can we infer about (ii)? Given (ii) and (iii), what can we infer about (i)? Providing formal tools that can be used to answer these questions is certainly of interest for human or artificial agents. Indeed, they could not only use them to plan their actions to achieve a given epistemic goal (the first and second questions actually correspond respectively to the problems of *deductive* and *abductive* planning in the situation calculus), but they could also use them to explain and determine a posteriori the causes that lead to a given situation. Nevertheless, to be applicable, these formal tools should lead to implementable decision procedures. To this aim, we provide a tableau method giving an answer to the first question. This is sufficient since we prove that the two other questions are in fact both reducible formally to the first one.

The paper is organized as follows. In Section 2, we define our three DEL-sequents corresponding to our three questions above, and we show that these DEL-sequents are interdefinable. In Section 3, we provide two terminating, sound and complete tableau methods. This leads us to define in Section 4 an algorithm in NEXPTIME, which we prove to be optimal by reducing a tiling problem known to be NEXPTIME-complete to our decision problem. A link to an implementation of our tableau method in LOTRECScheme is provided in Section 5. Finally, we conclude in Section 6 by a discussion of related works.

2 Dynamic Epistemic Logic: DEL-sequents

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

In the rest of this paper, Φ is a set of propositional letters called *atomic facts* which describe static situations, and Agt is a finite set of agents. A \mathcal{L} -model

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

- W is a non-empty set of possible worlds,
- $R : \text{Agt} \rightarrow 2^{W \times W}$ is a function assigning to each agent $j \in \text{Agt}$ an accessibility relation on W ,
- $V : \Phi \rightarrow 2^W$ 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_jv for $R(j)(w, v)$ and $R_j(w) = \{v \in W \mid wR_jv\}$. Intuitively, wR_jv means that in world w agent j considers that world v might correspond to 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:

$$\mathcal{L} : \phi ::= p \mid \neg\phi \mid \phi \wedge \phi \mid B_j\phi$$

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

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

We write $\mathcal{M} \models \phi$ when $\mathcal{M}, w \models \phi$ for all $w \in \mathcal{M}$, and $\models \phi$ when for all \mathcal{L} -model \mathcal{M} , $\mathcal{M} \models \phi$. An \mathcal{L} -formula ϕ is said to be *valid* if $\models \phi$.

The formula $B_j\phi$ reads as “agent j believes ϕ ”. Its truth conditions are defined in such a way that agent j believes ϕ holds in a possible world when ϕ holds in all the worlds agent j considers possible.

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

The propositional letters p'_ψ describing events are called *atomic events* and range over $\Phi' = \{p'_\psi \mid \psi \text{ ranges over } \mathcal{L}\}$. The reading of p'_ψ is “an event of precondition ψ is occurring”. A \mathcal{L}' -model is a tuple $\mathcal{M}' = (W', R', V')$ where:

- W' is a non-empty set of possible events,
- $R' : \text{Agt} \rightarrow 2^{W' \times W'}$ is a function assigning to each agent $j \in \text{Agt}$ an accessibility relation on W' ,
- $V' : \Phi' \rightarrow 2^{W'}$ 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'_ψ such that $w' \in V(p'_\psi)$ (Exclusivity).

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 event). If $w', v' \in W'$, we write $w'R'_jv'$ for $R'(j)(w', v')$ and $R'_j(w') = \{v' \in W' \mid w'R'_jv'\}$. Intuitively, $v' \in R_j(w')$ means

that while the possible event represented by w' is occurring, agent j considers possible that the possible event represented by v' is actually occurring. Our definition of a \mathcal{L}' -model is equivalent to the definition of an action signature in the logical framework of [Baltag and Moss, 2004].⁶ Just as we defined a language \mathcal{L} for \mathcal{L} -models, we also define a language \mathcal{L}' for \mathcal{L}' -models:

$$\mathcal{L}' : \phi' ::= p'_{\psi} \mid \neg\phi' \mid \phi' \wedge \phi' \mid B_j\phi'$$

where p'_{ψ} ranges over $\Phi' = \{p'_{\psi} \mid \psi \in \mathcal{L}\}$ and j over Agt . In fact, \mathcal{L}' was already introduced in [Baltag et al., 1999]. 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 a \mathcal{L}' -model, $w' \in \mathcal{M}'$ and $\phi' \in \mathcal{L}'$. $\mathcal{M}', w' \models \phi'$ is defined inductively as follows:

$$\begin{aligned} \mathcal{M}', w' \models p'_{\psi} & \quad \text{iff } w' \in V'(p'_{\psi}) \\ \mathcal{M}', w' \models \neg\phi' & \quad \text{iff not } \mathcal{M}', w' \models \phi' \\ \mathcal{M}', w' \models \phi' \wedge \psi' & \quad \text{iff } \mathcal{M}', w' \models \phi' \text{ and } \mathcal{M}', w' \models \psi' \\ \mathcal{M}', w' \models B_j\phi' & \quad \text{iff for all } v' \in R_j(w'), \mathcal{M}', v' \models \phi' \end{aligned}$$

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

A \mathcal{L}' -model induces the definition of a precondition function. The precondition $\text{Pre}(w')$ of a possible event w' corresponds to the property that should be true at a world w of a \mathcal{L} -model so that the possible event w' can ‘physically’ occur in this world w . The *precondition function* $\text{Pre} : W' \rightarrow \mathcal{L}$ induced by the \mathcal{L}' -model $\mathcal{M}' = (W', R', V')$ is defined as follows: $\text{Pre}(w') = \psi$ if there is p'_{ψ} such that $\mathcal{M}', w' \models p'_{\psi}$; $\text{Pre}(w') = \top$ otherwise.

We then redefine equivalently in our setting the BMS product update of [Baltag et al., 1998] 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 $(\mathcal{M} \otimes \mathcal{M}', (w, w')) = (W^{\otimes}, R^{\otimes}, V^{\otimes}, (w, w'))$ defined as follows:

- $W^{\otimes} = \{(v, v') \in W \times W' \mid \mathcal{M}, v \models \text{Pre}(v')\}$,
- $R_j^{\otimes}(v, v') = \{(u, u') \in W^{\otimes} \mid u \in R_j(v) \text{ and } u' \in R'_j(v')\}$,
- $V^{\otimes}(p) = \{(v, v') \in W^{\otimes} \mid \mathcal{M}, v \models p\}$.

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') .

⁶ If $\Sigma = (W', R', (w'_1, \dots, w'_n))$ is an action signature and $\phi_1, \dots, \phi_n \in \mathcal{L}$, then the \mathcal{L}' -model associated to $(\Sigma, \phi_1, \dots, \phi_n)$ is the tuple $\mathcal{M}' = (W', R', V')$ where $V'(p'_{\psi}) = \{w'_i\}$ if $\psi = \phi_i$, $V'(p'_{\psi}) = W' - \{w'_1, \dots, w'_n\}$ if $\psi = \top$, and $V'(p'_{\psi}) = \emptyset$ otherwise.

2.4 Definitions of our DEL-sequents

Let $\phi, \phi'' \in \mathcal{L}$ and $\phi' \in \mathcal{L}'$. We define the logical consequence relations $\phi, \phi' \models \phi''$, $\phi, \phi'' \models^2 \phi'$ and $\phi', \phi'' \models^3 \phi$ as follows. The second and third relations can be used for epistemic planning and goal regression respectively.

$\phi, \phi' \models \phi''$ iff for all pointed \mathcal{L} -model (\mathcal{M}, w) , and \mathcal{L}' -model (\mathcal{M}', w') such that $\mathcal{M}, w \models \text{Pre}(w')$, $\mathcal{M}, w \models \phi$ and $\mathcal{M}', w' \models \phi'$, it holds that $(\mathcal{M}, w) \otimes (\mathcal{M}', w') \models \phi''$

$\phi, \phi'' \models^2 \phi'$ iff for all pointed \mathcal{L} -models (\mathcal{M}, w) , and (\mathcal{M}'', w'') such that $\mathcal{M}, w \models \phi$ and $\mathcal{M}'', w'' \models \phi''$, if (\mathcal{M}', w') is a pointed \mathcal{L}' -model such that $\mathcal{M}, w \models \text{Pre}(w')$ and $(\mathcal{M}, w) \otimes (\mathcal{M}', w')$ is bisimilar to (\mathcal{M}'', w'') , then $\mathcal{M}', w' \models \phi'$

$\phi', \phi'' \models^3 \phi$ iff for all pointed \mathcal{L}' -model (\mathcal{M}', w') , and \mathcal{L} -model (\mathcal{M}'', w'') such that $\mathcal{M}', w' \models \phi'$ and $\mathcal{M}'', w'' \models \phi''$, if (\mathcal{M}, w) is a pointed \mathcal{L} -model such that $\mathcal{M}, w \models \text{Pre}(w')$ and $(\mathcal{M}, w) \otimes (\mathcal{M}', w')$ is bisimilar to (\mathcal{M}'', w'') , then $\mathcal{M}, w \models \phi$

In fact, as the following proposition shows, our three DEL-sequent are inter-definable. Therefore, in the rest of this paper, we will focus only on providing a tableau method for the DEL-sequent $\phi, \phi' \models \phi''$. Tableau methods and complexity results for the other DEL-sequents can easily be adapted from the ones provided for this DEL-sequent.

Proposition 2.1 *For all $\phi, \phi'' \in \mathcal{L}$ and $\phi' \in \mathcal{L}'$,*

$$\phi, \phi'' \models^2 \phi' \quad \text{iff} \quad \phi, \neg\phi' \models \neg\phi'' \qquad \phi', \phi'' \models^3 \phi \quad \text{iff} \quad \neg\phi, \phi' \models \neg\phi''$$

3 Tableau method

We consider three formulae, $\phi \in \mathcal{L}$, $\phi' \in \mathcal{L}'$ and $\phi'' \in \mathcal{L}$, and we want to address the problem of deciding whether $\phi, \phi' \models \phi''$ holds. To do so we equivalently decide whether there exist a pointed \mathcal{L} -model (\mathcal{M}, w) and a pointed \mathcal{L}' -model (\mathcal{M}', w') such that $\mathcal{M}, w \models \text{Pre}(w')$, $\mathcal{M}, w \models \phi$, $\mathcal{M}', w' \models \phi'$ and $\mathcal{M} \otimes \mathcal{M}', (w, w') \models \neg\phi''$. We call this dual problem the *satisfiability problem*.

3.1 Tableau method description

The formulas that appear in our tableau method and that we call *tableau formulas* are of the following kind:

- $(l \phi)$: l is a label l_w (resp. $l_{w'}$) that represents a world of the model \mathcal{M} (resp. \mathcal{M}') being constructed, and ϕ is a formula of \mathcal{L} (resp. \mathcal{L}') that should be true at \mathcal{M}, w (resp. \mathcal{M}', w').
- $(l_w l_{w'} \phi'')$: l_w represents a world w of \mathcal{M} , $l_{w'}$ a world w' of \mathcal{M}' , and ϕ'' is a formula of \mathcal{L} that should be true at $\mathcal{M} \otimes \mathcal{M}', (w, w')$. Moreover,

$(l_w l_{w'} 0)$ means that (w, w') is not in $\mathcal{M} \otimes \mathcal{M}'$.

- $(R \ l \ l')$ (resp. $(R' \ l \ l')$): R (resp. R') is some R_j (resp. R'_j), l and l' represent two worlds w and u (resp. w' and u') such that wR_ju (resp. $w'R_ju'$).
- \perp : Denotes an inconsistency.

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 formulas.

A *tableau for* a triple (ϕ, ϕ', ϕ'') of formulas is a finite tree with a set of tableau formulas at each node, and whose root is:

$$\Gamma_0 = \{(l_w \phi), (l_{w'} \phi'), (l_w l_{w'} \phi'')\}$$

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 $\perp \in \Gamma$. The tableau is extended the following way:

- (i) Choose a leaf node n carrying Γ where n is not an end node, and choose a rule ρ applicable to n .
- (ii) (a) If ρ has only one denominator, add the appropriate instantiation 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 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*. A triple (ϕ, ϕ', ϕ'') is said to be *consistent* if no tableau for (ϕ, ϕ', ϕ'') is closed, and a triple (ϕ, ϕ', ϕ'') is a *theorem*, which we write $\phi, \phi' \vdash \phi''$, if there is a closed tableau for $(\phi, \phi', \neg\phi'')$.

3.2 Tableau rules

Common rules for \mathcal{M} , \mathcal{M}' and \mathcal{M}'' (l is either l_w , $l_{w'}$ or $l_w l_{w'}$):

$$\frac{(l \phi \wedge \psi)}{(l \phi) (l \psi)} \wedge \qquad \frac{(l \neg(\phi \wedge \psi))}{(l \neg\phi) \mid (l \neg\psi)} \neg\wedge \qquad \frac{(l \neg\neg\phi)}{(l \phi)} \neg \qquad \frac{(l p)(l \neg p)}{\perp} \perp$$

where $p \in \Phi$

Specific rules for \mathcal{M} and \mathcal{M}' (l is either l_w or $l_{w'}$):

$$\frac{(l \langle B_j \rangle \phi)}{(R l l')(l' \phi)} \langle B_j \rangle \quad \frac{(l B_j \phi)(R l l')}{(l' \phi)} B_j \quad \frac{(l_{w'} p'_\phi)(l_{w'} p'_\psi)}{\perp} \text{Excl}$$

where $\phi \neq \psi$

Specific rules for \mathcal{M}'' :

$$\frac{(l_w l_{w'} \langle B_j \rangle \phi)}{(R l_w l_u)(R' l_{w'} l_{w'})(l_u l_{w'} \phi)} \langle B_j \rangle_{\otimes} \quad \frac{(l_w l_{w'} B_j \phi)(R l_w l_u)(R' l_{w'} l_{w'})}{(l_u l_{w'} \phi)|(l_u l_{w'} 0)} B_{j\otimes}$$

$$\frac{(l_w l_{w'} p)}{(l_w p)} \leftarrow_1 \quad \frac{(l_w l_{w'} \neg p)}{(l_w \neg p)} \leftarrow_2$$

$$\frac{(l_w l_{w'} 0)(l_{w'} p'_\psi)}{(l_w \neg \psi)} \text{Pre}_1 \quad \frac{(l_w l_{w'} \phi)(l_{w'} p'_\psi)}{(l_w \psi)} \text{Pre}_2 \quad \frac{(l_w l_{w'} \phi)(l_w l_{w'} 0)}{\perp} \perp''$$

where $\phi \neq 0$

where $\phi \neq 0$

Remark 3.1 Another sound and complete tableau method can be obtained by replacing the rules Pre_1 and \perp'' above by the following rule:

$$\frac{(l_w l_{w'} 0)}{(l_{w'} p'_{\psi_1}) (l_w \neg \psi_1) \mid \dots \mid (l_{w'} p'_{\psi_n}) (l_w \neg \psi_n)} \text{Pre}'_1$$

where $p'_{\psi_1}, \dots, p'_{\psi_n}$ is the set of propositional letters appearing in ϕ' at the root of the tableau. This second tableau method is more modular, in the sense that if we remove Rule (Excl), then the resulting tableau method is still sound and complete with respect to the semantics where we do not impose the (Exclusivity) condition on \mathcal{L}' -models. Note also that the \mathcal{L} -model and \mathcal{L}' -model obtained from an open branch with this tableau method do not need to be adapted to fulfill the satisfiability problem, as in the proof of Proposition 3.3 with the first tableau method.

3.3 Tableau method soundness and completeness

Proposition 3.2 (Tableau method soundness) *For all $\phi, \phi'' \in \mathcal{L}$, for all $\phi' \in \mathcal{L}'$, $\phi, \phi' \vdash \phi''$ implies $\phi, \phi' \models \phi''$*

Proof. Instead of proving that $\phi, \phi' \vdash \phi''$ implies $\phi, \phi' \models \phi''$, we equivalently prove that $\phi, \phi' \not\models \phi''$ implies $\phi, \phi' \not\vdash \phi''$. Suppose there exist a pointed \mathcal{L} -model (\mathcal{M}, w) , a \mathcal{L}' -model (\mathcal{M}', w') such that $\mathcal{M}, w \models \phi$, $\mathcal{M}', w' \models \phi'$, $\mathcal{M}, w \models \text{Pre}(w')$ and $\mathcal{M} \otimes \mathcal{M}', (w, w') \not\models \phi''$. We must prove that every tableau for $(\phi, \phi', \neg \phi'')$ has an open branch (the proof of termination is postponed to Section 4).

We say that a set Σ of tableau formulae is *interpretable* if there exist a \mathcal{L} -model \mathcal{M} , a \mathcal{L}' -model \mathcal{M}' , $f : \text{LABEL} \rightarrow W$ and $f' : \text{LABEL}' \rightarrow W'$ (where LABEL and LABEL' are the sets of labels for worlds appearing in Σ) such that $(\mathcal{M}, \mathcal{M}', f, f')$ makes all the tableau formulae in Σ true for the following

semantics \models_T :

$$\begin{aligned}
 (\mathcal{M}, \mathcal{M}', f, f') \models_T (l_w \phi) & \quad \text{iff } \mathcal{M}, f(l_w) \models \phi \\
 (\mathcal{M}, \mathcal{M}', f, f') \models_T (l_{w'} \phi') & \quad \text{iff } \mathcal{M}', f'(l_{w'}) \models \phi' \\
 (\mathcal{M}, \mathcal{M}', f, f') \models_T (R l_w l_u) & \quad \text{iff } f(l_w)Rf(l_u) \\
 (\mathcal{M}, \mathcal{M}', f, f') \models_T (R' l_{w'} l_{u'}) & \quad \text{iff } f'(l_{w'})R'f'(l_{u'}) \\
 (\mathcal{M}, \mathcal{M}', f, f') \models_T (l_w l_{w'} 0) & \quad \text{iff } \mathcal{M}, f(l_w) \not\models \text{Pre}(f'(l_{w'})) \\
 (\mathcal{M}, \mathcal{M}', f, f') \models_T (l_w l_{w'} \phi'') & \quad \text{iff } \mathcal{M}, f(l_w) \models \text{Pre}(f'(l_{w'})) \text{ and} \\
 & \quad \mathcal{M} \otimes \mathcal{M}', (f(l_w), f'(l_{w'})) \models \phi'' \\
 (\mathcal{M}, \mathcal{M}', f, f') \models_T \perp & \quad \text{iff false}
 \end{aligned}$$

Notice that since $\phi, \phi' \not\models \phi''$, the set $\Gamma_0 = \{(l_w \phi)(l_{w'} \phi')(l_w l_{w'} \neg\phi'')\}$ is interpretable. Furthermore, 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 $(\phi, \phi', \neg\phi'')$ has an open branch. We only prove it for the specific rules of \mathcal{M}'' , the proof for the other rules being standard. In the following, when f is a function, we let $f(x \mapsto a)$ be the function that maps x to a and y to $f(y)$ if $y \neq x$.

Rule $\langle B_j \rangle_{\otimes}$: If $\mathcal{M}, \mathcal{M}', f, f' \models_T (l_w l_{w'} \langle B_j \rangle \phi)$ then $\mathcal{M} \otimes \mathcal{M}', (f(l_w), f'(l_{w'})) \models \langle B_j \rangle \phi$. So there exists $(u, u') \in W''$ such that $(f(l_w), f'(l_{w'}))R''(u, u')$ and $\mathcal{M} \otimes \mathcal{M}', (u, u') \models \phi$. Since $(f(l_w), f'(l_{w'}))R''(u, u')$ we have that $f(l_w)Ru, f'(l_{w'})R'u'$ and $\mathcal{M}, u \models \text{Pre}(u')$. So by letting $g := f(l_u \mapsto u)$ and $g' := f'(l_{u'} \mapsto u')$ we have that $\mathcal{M}, \mathcal{M}', g, g' \models_T \{(R l_w l_u)(R' l_{w'} l_{u'})(l_u l_{u'} \phi)\}$.

Rule $B_{j\otimes}$: If $\mathcal{M}, \mathcal{M}', f, f' \models_T \{(l_w l_{w'} B_j \phi)(R l_w l_u)(R' l_{w'} l_{u'})\}$ then $\mathcal{M}, f(l_w) \models \text{Pre}(f'(l_{w'}))$, $\mathcal{M} \otimes \mathcal{M}', (f(l_w), f'(l_{w'})) \models B_j \phi$, $f(l_w)Rf(l_u)$ and $f'(l_{w'})R'f'(l_{u'})$. So, either $\mathcal{M}, f(l_u) \not\models \text{Pre}(l_{u'})$ or $\mathcal{M}, f(l_u) \models \text{Pre}(l_{u'})$. In the first case, $\mathcal{M}, \mathcal{M}', f, f' \models_T (l_u l_{u'} 0)$. In the second case, $(f(l_u), f'(l_{u'}))$ is a world of \mathcal{M}'' , and $(f(l_w), f'(l_{w'}))R''(f(l_u), f'(l_{u'}))$. Therefore we have $\mathcal{M} \otimes \mathcal{M}', (f(l_u), f'(l_{u'})) \models \phi$, hence $\mathcal{M}, \mathcal{M}', f, f' \models_T (l_u l_{u'} \phi)$.

Rules \leftarrow_1 and \leftarrow_2 : If $\mathcal{M}, \mathcal{M}', f, f' \models_T \{l_w l_{w'} p\}$ then $\mathcal{M}, f(l_w) \models \text{Pre}(f'(l_{w'}))$ and $\mathcal{M} \otimes \mathcal{M}', (f(l_w), f'(l_{w'})) \models p$. Since $V''(f(l_w), f'(l_{w'})) = V(f(l_w))$, we have that $\mathcal{M}, f(l_w) \models p$, hence $\mathcal{M}, \mathcal{M}', f, f' \models_T \{(l_w p)\}$. Rule \leftarrow_2 is proved similarly.

Rules Pre_1 and Pre_2 : If $\mathcal{M}, \mathcal{M}', f, f' \models_T \{(l_w l_{w'} \phi)(l_{w'} p'_\psi)\}$ for some $\phi \neq 0$, then $\mathcal{M}, f(l_w) \models \text{Pre}(f'(l_{w'}))$, and $f'(l_{w'}) \in V'(p'_\psi)$. So $\mathcal{M}, f(l_w) \models \text{Pre}(p'_\psi)$, and $\mathcal{M}, \mathcal{M}', f, f' \models_T (l_w \psi)$. As for Rule Pre_1 , if $\mathcal{M}, \mathcal{M}', f, f' \models_T \{(l_w l_{w'} 0)(l_{w'} p'_\psi)\}$, then, by definition of \models_T , $\mathcal{M}, f(l_w) \models \neg\text{Pre}(f'(l_{w'}))$ and $\mathcal{M}', f'(l_{w'}) \models p'_\psi$. Therefore, by the (Exclusivity) condition, $\text{Pre}(f'(l_{w'})) = \psi$, and so $\mathcal{M}, f(l_w) \models \neg\psi$, i.e. $(l_w \neg\psi)$.

Rule \perp'' : if $\phi \neq 0$, then the set $\{(l_w l_{w'} \phi)(l_w l_{w'} 0)\}$ is not interpretable

by definition. So the result trivially holds in this case. \square

Proposition 3.3 (Tableau method completeness) *For all $\phi, \phi'' \in \mathcal{L}$, for all $\phi' \in \mathcal{L}'$, $\phi, \phi' \models \phi''$ implies $\phi, \phi' \vdash \phi''$.*

Proof. In the proof of the proposition we make use of an unbounded number of atomic facts. So we start by showing that the case where Φ is finite reduces to the case where it is infinite.

We define the \mathcal{L}_Φ -models as the \mathcal{L} -models, where the valuation function has domain Φ . The epistemic language \mathcal{L}_Φ is defined like \mathcal{L} , where the atomic facts range over Φ , and we write $\models_\Phi \phi$ if $\mathcal{M} \models \phi$ for all \mathcal{L}_Φ -model \mathcal{M} . We similarly define \mathcal{L}'_Φ , and we write \mathcal{L}'_Φ for $\mathcal{L}'_{\{p'_\psi \mid \psi \in \mathcal{L}_\Phi\}}$. We finally define $\phi, \phi' \models_\Phi \phi''$ like $\phi, \phi' \models \phi''$, where models are \mathcal{L}_Φ -models and \mathcal{L}'_Φ -models.

We just need to prove that for all Φ , for all $\phi, \phi'' \in \mathcal{L}_\Phi$ and $\phi' \in \mathcal{L}'_\Phi$, $\phi, \phi' \models_\Phi \phi''$ implies that for all Φ^* such that $\Phi \subseteq \Phi^*$, it holds that $\phi, \phi' \models_{\Phi^*} \phi''$. Indeed, if this holds, then if $\phi, \phi' \models_\Phi \phi''$ with Φ finite, we take some infinite $\Phi \subseteq \Phi^*$, we have that $\phi, \phi' \models_{\Phi^*} \phi''$; hence if Proposition 3.3 holds for Φ infinite, it also holds for Φ finite.

To prove it we need the following result, in which, for $\phi \in \mathcal{L}_\Phi$ and $\phi' \in \mathcal{L}'_\Phi$, $\phi \otimes \phi' \in \mathcal{L}_\Phi$ is the *progression* of ϕ by ϕ' (see [Aucher, 2011] for definition).

Theorem 3.4 ([Aucher, 2011]) *For any Φ , for any $\phi, \phi'' \in \mathcal{L}_\Phi$ and $\phi' \in \mathcal{L}'_\Phi$, $\phi, \phi' \models_\Phi \phi''$ iff $\models_\Phi \phi \otimes \phi' \rightarrow \phi''$ (note that $\phi \otimes \phi' \rightarrow \phi''$ is an \mathcal{L}_Φ -formula).*

Let Φ be a set of propositional letters and let $\phi, \phi'' \in \mathcal{L}_\Phi$ and $\phi' \in \mathcal{L}'_\Phi$. Assume that $\phi, \phi' \models_\Phi \phi''$. Now let Φ^* such that $\Phi \subseteq \Phi^*$. By Theorem 3.4, $\models_\Phi \phi \otimes \phi' \rightarrow \phi''$. Besides, $\models_{\Phi^*} \phi \otimes \phi' \rightarrow \phi''$ also holds because $\Phi \subseteq \Phi^*$ and $\phi \otimes \phi' \rightarrow \phi'' \in \mathcal{L}_\Phi$. Then, by Theorem 3.4 again, $\phi, \phi' \models_{\Phi^*} \phi''$.

We now prove Proposition 3.3 in the case where Φ is infinite. We prove that $\phi, \phi' \not\models \phi''$ implies $\phi, \phi' \not\vdash \phi''$. Suppose there is a tableau for $(\phi, \phi', \neg\phi'')$ that has an open branch, we prove that there exist a pointed \mathcal{L} -model (\mathcal{M}, w) and a pointed \mathcal{L}' -model (\mathcal{M}', w') such that $\mathcal{M}, w \models \text{Pre}(w')$, $\mathcal{M}, w \models \phi$, $\mathcal{M}', w' \models \phi'$ and $\mathcal{M} \otimes \mathcal{M}', (w, w') \models \neg\phi''$.

Let Γ_f be the set of tableau formulas carried by the end node of the open branch. We define \mathcal{M} and \mathcal{M}' as follows. Each of them is built in two steps.

- Let $\mathcal{M} = (W, R, V)$ with $W = \{w \mid (l_w \psi) \in \Gamma_f\}$ and $R = \{(w, u) \mid (R l_w l_u) \in \Gamma_f\}$. V is defined in two steps : for all atoms p that appear in ϕ or ϕ'' , $V(p) = \{w \mid (l_w p) \in \Gamma_f\}$. Then, for each world w in W , we assign a fresh atomic variable p_w , and define $V(p_w) = \{w\}$. This is possible because Φ is infinite.
- Let $\mathcal{M}' = (W', R', V')$ with $W' = \{w' \mid (l_{w'} \psi) \in \Gamma_f\}$, $V(p'_\psi) = \{w' \mid (l_{w'} p'_\psi) \in \Gamma_f\}$, and $R' = \{(w', u') \mid (R' l_{w'} l_{u'}) \in \Gamma_f\}$. Moreover, for all $w' \in \mathcal{M}'$ such that there is no $(l_{w'} p'_\psi) \in \Gamma_f$, we set $w' \in V'(p'_{\psi_{w'}})$,

where $\psi_{w'}$ is defined as follows : let $S_{w'} = \{w \mid \exists \phi \neq 0, (l_w l_{w'} \phi) \in \Gamma_f\}$; then $\psi_{w'} = \bigvee_{w \in S_{w'}} p_w$. Note that by soundness of Rule (Excl), \mathcal{M}' satisfies the exclusivity condition.

Finally, we define \mathcal{M}'' as $\mathcal{M} \otimes \mathcal{M}'$ (we will prove later that $\mathcal{M}, w \models \text{Pre}(w')$). Lemmas 3.5 and 3.6 below establish the completeness of our tableau method. \square

Lemma 3.5 *If $(l_w \phi) \in \Gamma_f$ then $\mathcal{M}, w \models \phi$, and if $(l_{w'} \phi') \in \Gamma_f$ then $\mathcal{M}', w' \models \phi'$.*

Proof. We only prove it for \mathcal{M} , it is similar for \mathcal{M}' . The proof is done by induction on ϕ .

$p, \neg p$: by definition of V . As for the case $\phi \wedge \psi$, by saturation of rule \wedge , Γ_f also contains $(l_w \phi)$ and $(l_w \psi)$. By induction hypothesis we have that $\mathcal{M}, w \models \phi$ and $\mathcal{M}, w \models \psi$, so $\mathcal{M}, w \models \phi \wedge \psi$. The cases $\neg(\phi \wedge \psi)$ and $\neg\neg\phi$ are proved similarly.

$\langle B_j \rangle \phi$: By saturation of rule $\langle B_j \rangle$ there exists l_u such that $(R l_w l_u) \in \Gamma_f$ and $(l_u \phi) \in \Gamma_f$. By induction hypothesis $\mathcal{M}, u \models \phi$, and wRu holds by construction of \mathcal{M} , so $\mathcal{M}, w \models \langle B_j \rangle \phi$.

$B_j \phi$: Take some u in W such that wRu holds, we prove that $\mathcal{M}, u \models \phi$ and conclude that $\mathcal{M}, w \models B_j \phi$. Since wRu holds we know by construction of \mathcal{M} that $(R l_w l_u)$ is in Γ_f . So by saturation of rule B_j , $(l_u \phi)$ also belongs to Γ_f , and by induction hypothesis $\mathcal{M}, u \models \phi$. \square

Lemma 3.6 *If there is $\phi'' \neq 0$ such that $(l_w l_{w'} \phi'') \in \Gamma_f$, then $\mathcal{M}, w \models \text{Pre}(w')$ and $\mathcal{M} \otimes \mathcal{M}', (w, w') \models \phi''$.*

Proof. We first prove the following Fact:

Fact 3.7 *If $(l_w l_{w'} \phi) \in \Gamma_f$ with $\phi \neq 0$, then $\mathcal{M}, w \models \text{Pre}(w')$.*

Assume towards a contradiction that $\mathcal{M}, w \not\models \text{Pre}(w')$. There are then two cases: either there is $(l_{w'} p'_\psi) \in \Gamma_f$ or there is no $(l_{w'} p'_\psi) \in \Gamma_f$. In the first case, $\mathcal{M}, w \not\models \psi$ because $\psi = \text{Pre}(w')$ by the (Exclusivity) condition. However, by the rule Pre_2 , $(l_w \psi) \in \Gamma_f$. Then, by Lemma 3.5, $\mathcal{M}, w \models \psi$. This is impossible. In the second case, $\mathcal{M}', w' \models p'_{\psi_{w'}}$ by definition of V' , and therefore $\text{Pre}(w') = \psi_{w'}$. Besides, $\mathcal{M}, w \models p_w$ by definition of V , and since $(l_w l_{w'} \phi) \in \Gamma_f$ for some $\phi \neq 0$, $\models p_w \rightarrow \psi_{w'}$. So $\mathcal{M}, w \models \psi_{w'}$, and finally $\mathcal{M}, w \models \text{Pre}(w')$, which is impossible.

We can now prove Lemma 3.6. We prove it by induction on ϕ .

$p, \neg p$: By Rule \leftarrow_1 , $(l_w p) \in \Gamma_f$, and so $\mathcal{M}, w \models p$ by Lemma 3.5. Moreover, by Fact 3.7, $\mathcal{M}, w \models \text{Pre}(w')$. Therefore, $\mathcal{M} \otimes \mathcal{M}', (w, w') \models p$ 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 $\phi \wedge \psi$, $\neg(\phi \wedge \psi)$ and $\neg\neg\phi$ is obtained by

applying straightforwardly the Induction Hypothesis.

$\langle B_j \rangle \phi$: If $(l_w l_{w'} \langle B_j \rangle \phi) \in \Gamma_f$, then by saturation of Rule $\langle B_j \rangle_{\otimes}$, $(R l_w l_u)$, $(R' l_{w'} l_{u'})$ and $(l_u l_{u'} \phi)$ belong to Γ_f . By application of Fact 3.7, $\mathcal{M}, w \models \text{Pre}(w')$. Now, by definition of \mathcal{M} and \mathcal{M}' , $u \in R(w)$ and $u' \in R'(w')$. Moreover, $(l_u l_{u'} \phi) \in \Gamma_f$ and $\phi \neq 0$, so by Fact 3.7, $\mathcal{M}, u \models \text{Pre}(u')$, hence $(u, u') \in R(w, w')$. By application of the induction hypothesis, $\mathcal{M} \otimes \mathcal{M}', (u, u') \models \phi$. Therefore, $\mathcal{M} \otimes \mathcal{M}', (w, w') \models \langle B_j \rangle \phi$.

$B_j \phi$: If $(l_w l_{w'} B_j \phi) \in \Gamma_f$, then by application of Fact 3.7, $\mathcal{M}, w \models \text{Pre}(w')$. Let $(u, u') \in R(w, w')$. Then $u \in R(w)$ and $u' \in R'(w')$ by definition of the product update. Then, by definition of \mathcal{M} and \mathcal{M}' , $(R l_w l_u) \in \Gamma_f$ and $(R' l_{w'} l_{u'}) \in \Gamma_f$. By saturation of Rule $B_{j\otimes}$, either (i) $(l_u l_{u'} 0) \in \Gamma_f$ or (ii) $(l_u l_{u'} \phi) \in \Gamma_f$.

(i) In the first case, assume that there is $(l_{u'} p'_{\psi}) \in \Gamma_f$. Then, by saturation of Rule Pre_1 , $(l_u \neg\psi) \in \Gamma_f$. Therefore, $\mathcal{M}, u \models \neg\psi$ by Lemma 3.5. This is impossible because $\mathcal{M}', u' \models p'_{\psi}$, and so $\mathcal{M}, u \models \psi$ should also hold because $(u, u') \in \mathcal{M} \otimes \mathcal{M}'$. Therefore, there is no $(l_{u'} p'_{\psi}) \in \Gamma_f$.

(a) If $u \in S_{u'}$, then there is $\phi \neq 0$ such that $(l_u l_{u'} \phi) \in \Gamma_f$. Hence, by saturation of Rule \perp'' , and because $(l_u l_{u'} 0) \in \Gamma_f$, the branch should be closed, which is impossible.

(b) If $u \notin S_{u'}$, then $\mathcal{M}, u \not\models \psi_{u'}$, because $\psi_{u'}$ characterizes exactly the worlds in $S_{u'}$. Hence, $\mathcal{M}, u \not\models \text{Pre}(u')$ by definition of V' , because $\mathcal{M}', u' \models p'_{\psi_{u'}}$. However $(u, u') \in R(w, w')$, so $\mathcal{M}, u \models \text{Pre}(u')$. There is a contradiction, so this case is impossible.

(ii) In the second case, by Induction Hypothesis, $\mathcal{M}, u \models \text{Pre}(u')$ and $\mathcal{M} \otimes \mathcal{M}', (u, u') \models \phi$

So, in any case, $\mathcal{M} \otimes \mathcal{M}, (u, u') \models \phi$. Therefore, $\mathcal{M} \otimes \mathcal{M}', (w, w') \models B_j \phi$. \square

4 Complexity of the satisfiability problem

Proposition 4.1 *The satisfiability problem is in NEXPTIME.*

Proof. The tableau rules presented in Section 3.2 give rise to a non-deterministic algorithm running in exponential time. We say that a label l_u is of depth k if there is a sequence $w = u_1, \dots, u_k = u$ such that $(R l_{w_i} l_{w_{i+1}})$ for all $i < k$. Let $p'_{\psi_1}, \dots, p'_{\psi_n}$ be the set of atomic propositions appearing in ϕ' . Let $\delta(\cdot)$ be the function that gives the modal depth of a given formula.⁷ The algorithm starts with the following set of tableau formulas $\Gamma_0 = \{(l_w \phi), (l_{w'} \phi'), (l_w l_{w'} \phi'')\}$. Let $N = \max\{\delta(\phi), \delta(\phi'), \delta(\phi'') +$

⁷ $\delta(\phi)$ is defined inductively as follows: $\delta(p) = 0$, $\delta(\neg\phi) = \delta(\phi)$, $\delta(\phi \wedge \psi) = \max\{\delta(\phi), \delta(\psi)\}$ and $\delta(B_j \phi) = 1 + \delta(\phi)$.

$\max_{k \in \{1, \dots, n\}} \delta(\psi_k)$.

The algorithm runs as follows. For $i = 0$ to N , we execute:

- (i) $\Gamma'_i :=$ the saturation of Γ_i by rules $\wedge, \neg\wedge, \neg, \perp, Excl, \leftarrow_1, \leftarrow_2, Pre_1, Pre_2, \perp''$;
- (ii) If $\perp \in \Gamma'_i$, we stop the current execution;
- (iii) $\Gamma_{i+1} :=$ the set of tableau formulas obtained by applying $\langle B_j \rangle, B_j, \langle B_j \rangle_{\otimes}, B_{j_{\otimes}}$ on Γ'_i .

Step 1 is non-deterministic and corresponds to a Boolean saturation of labels of depth i . It non-deterministically runs in linear size of Γ_i . Step 2 consists in checking if rule \perp has been executed. In this case, the current execution halts. Step 3 produces tableau formulas where labels are of depth $i + 1$.

Note that the maximal depth of formulas ψ'' in tableau formulas of the form $(l_u l_{u'} \psi'')$ in Γ_i is strictly decreasing with i (see rule $\langle B_j \rangle_{\otimes}$ and $B_{j_{\otimes}}$). So when $i > \delta(\phi'')$, there is no more tableau formula of the form $(l_u l_{u'} \psi'')$ in Γ_i with $\psi'' \neq 0$. So when $i > \delta(\phi'')$, the rules $Pre_2, \langle B_j \rangle_{\otimes}$ and $B_{j_{\otimes}}$ will no more be applied.

Likewise, the maximal depth of formulas ψ (resp. ψ') in tableau formulas of the form $(l_u \psi)$ (resp. $(l_{u'} \psi')$) in Γ_i is strictly decreasing with i . Moreover the depth of the formulas ψ appearing in a tableau formula of the form $(l_u \psi)$ is less than $\max\{\delta(\phi), \max_{k \in \{1, \dots, n\}} \delta(\psi_k)\}$, and the depth of the formulas ψ' appearing in a tableau formula of the form $(l_{u'} \psi')$ is less than $\delta(\phi')$.

At the end, $\Gamma_{N+1} = \emptyset$ and the algorithm has applied rules until saturation, that is, the set of tableau formulas $\bigcup_{i=0}^N \Gamma_i$ 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 ϕ, ϕ', ϕ'' and $Pre(p'_{\psi_k})$. Step 1 saturates the worlds u, u' and (u, u') appearing in the tableau formulas in Γ_i . For each of those worlds, the saturation is linear in x . Step 3 creates new tableau formulas for each $\langle B_j \rangle$ -formula appearing in Γ'_i . So for each world in Γ_i it produces at most $2x$ new worlds. If we note y_i the maximal number of worlds in Γ_i , we have that $y_{i+1} = 2xy_i$. So $y_i = (2x)^i$. The number of created worlds is bounded by $(2x)^{x+1}$ and this construction takes an exponential amount of time. \square

To prove NEXPTIME-hardness of the satisfiability problem, we will reduce a NEXPTIME-complete tiling problem to it [Boas, 1997]. Let k be a natural number. A tile type t is a 4-tuple of colors $t = (left(t), right(t), up(t), down(t))$. The tiling problem we consider is defined as follows.

- Input: a finite set T of tile types, a $t_0 \in T$ and a natural number k written in its binary form.

- Output: yes iff we can tile a $k \times k$ grid with the tile types of T and t_0 being placed onto $(0, 0)$.

In other words, the problem is to decide whether there exists a function τ from $\{1, \dots, k\}^2$ to T satisfying the following constraints:

- (i) $\tau(0, 0) = t_0$;
- (ii) $up(\tau(x, y)) = down(\tau(x, y + 1))$ for all $x \in \{1, \dots, k\}$, $y \in \{1, \dots, k - 1\}$;
- (iii) $right(\tau(x, y)) = left(\tau(x + 1, y))$ for all $x \in \{1, \dots, k - 1\}$, $y \in \{1, \dots, k\}$.

Proposition 4.2 *The satisfiability problem is NEXPTIME-hard.*

Proof. Without loss of generality we may assume that $k = 2^n$. Let us consider an instance (T, t_0, k) of the tiling problem. We now define three formulas ϕ, ϕ', ϕ'' that are computable in polynomial time in $|T|$ and n such that it is possible to tile a $k \times k$ grid with the tile types of T and t_0 being placed onto $(0, 0)$ iff (ϕ, ϕ', ϕ'') is satisfiable.

There is a modal formula χ of length $O(n^2)$ which is satisfied in a frame iff the model contains as a submodel a binary tree of depth $2n$, for instance:

$$\chi = \bigwedge_{l < 2n} B_j^l ((\langle B_j \rangle p_l \wedge \langle B_j \rangle \neg p_l) \wedge \bigwedge_{i < l} ((p_i \rightarrow B_j p_i) \wedge (\neg p_i \rightarrow B_j \neg p_i))).$$

The 2^{2n} leaves of the tree are labeled by $2n$ -tuples containing either p_i or $\neg p_i$ for $i < 2n$. The 2^{2n} leaves correspond to the 2^{2n} tile locations (x, y) in the following sense: the values of the propositions p_i , where $i < n$, correspond to the binary representation of the abscissa x and the values of the propositions p_i , where $n \leq i < 2n$, correspond to the binary representation of the ordinate y . For instance, for $n = 4$ the location where $x = 4$ and $y = 3$ is represented by the following valuation:

$$\underbrace{\neg p_0, p_1, \neg p_2, \neg p_3}_4 \underbrace{\neg p_4, \neg p_5, p_6, p_7}_3$$

The idea of encoding the existence of a $k \times k$ tiling is as follows:

- ϕ encodes a tiling τ_1 with such a binary tree such that $\tau_1(0, 0) = t_0$;
- ϕ' also encodes a tiling τ_2 with such a binary tree;
- ϕ'' encodes that $\tau_1 = \tau_2 = \tau$, and constraints (ii) and (iii) of the tiling τ .

Defining ϕ

We define the following formula: $path = \langle B_j \rangle^{2n+|T|} \top \wedge \bigwedge_{i < 2n+|T|} B_j^i \langle B_j \rangle \top$. The formula $path$ says that there is a path whose length is greater than $2n + |T|$ but no shorter path in the model.

In order to define ϕ , each tiling type t is used as a proposition in the language, and means : ‘for the current location (x, y) , we have $\tau_1(x, y) = t$ ’.

We define ϕ by:

$$\phi = \chi \wedge B_j^{2n} \left(\text{path} \wedge \bigvee_{t \in T} t \wedge \bigwedge_{t \in T} (t \rightarrow \bigwedge_{u \in T, u \neq t} \neg u) \wedge ((\bigwedge_{i < 2n} \neg p_i) \rightarrow t_0) \right)$$

Defining ϕ'

For all i , we define $l'_i = \langle B_j \rangle^{i+1} B_j \perp$. Let $\chi' = \bigwedge_{i < 2n} B_j^i (\langle B_j \rangle B_j^{2n-i-1} l'_i \wedge \langle B_j \rangle B_j^{2n-i-1} \neg l'_i)$. The formula χ' has the same aim as χ and enables to enforce the existence of a binary tree where leaves correspond to the locations (x, y) of the tiling τ_2 . Formulas l'_i for $i < 2n$ represent the binary representation of (x, y) .

Let $t_1, \dots, t_{|T|}$ be an enumeration of elements of T . In order to define ϕ' , for each tiling type t_i we use the formula $t'_i = l'_{i+2n}$ in the language whose intuitive meaning is ‘for the current location (x, y) , we have $\tau_2(x, y) = t_i$ ’.

We define ϕ' by

$$\phi' = \chi' \wedge \text{goodProduct} \wedge B_j^{2n} \left(\bigvee_{i \in \{1, \dots, |T|\}} t'_i \wedge \bigwedge_{i \in \{1, \dots, |T|\}} (t'_i \rightarrow \bigwedge_{k \in \{1, \dots, |T|\}, k \neq i} \neg t'_k) \right).$$

where $\text{goodProduct} = \bigwedge_{i \leq 2n+2n+|T|+1} B_j^i p'_\top$ ensures that all worlds (w, w') appear in the product model.

Defining ϕ''

The formula ϕ'' will consider all the leaves (w, w') of the product model where w is a leaf of the model M and w' is a leaf of the model M' in order to encode the fact that $\tau_1 = \tau_2$ and the constraints (ii) and (iii).

We define ϕ'' by:

$$\begin{aligned} \phi'' = B_j^{2n} [& (\alpha \wedge \beta \rightarrow \bigwedge_{j \in \{1, \dots, |T|\}} (t_j \leftrightarrow t'_j)) \wedge \\ & (\alpha \wedge \beta_1 \rightarrow \bigwedge_{j \in \{1, \dots, |T|\}} (t_j \rightarrow \bigvee_{k \in \{1, \dots, |T|\} | \text{down}(t_k) = \text{up}(t_j)} t'_k)) \wedge \\ & (\alpha_1 \wedge \beta \rightarrow \bigwedge_{j \in \{1, \dots, |T|\}} (t_j \rightarrow \bigvee_{k \in \{1, \dots, |T|\} | \text{left}(t_k) = \text{right}(t_j)} t'_k))] \end{aligned}$$

where:

- $\alpha = \bigwedge_{i < n} (p_i \leftrightarrow l'_i)$ means ‘the abscissa x of the tile location of w is equal to the abscissa x' of the tile location of w' ’;
- $\beta = \bigwedge_{n \leq i < 2n} (p_i \leftrightarrow l'_i)$ means ‘the ordinate y of the tile location of w is equal to the ordinate y' of the tile location of w' ’;
- $\alpha_1 = \bigvee_{i < n} \left(\bigwedge_{j < i} (p_j \leftrightarrow l'_j) \wedge \neg p_i \wedge l'_i \wedge \bigwedge_{i < j < n} (p_j \wedge \neg l'_j) \right)$ means ‘the abscissa x of the tile location of w and the abscissa x' of the tile location of w' are such that $x' = x + 1$ ’;
- $\beta_1 = \bigvee_{n \leq i < 2n} \left(\bigwedge_{n \leq j < i} (p_j \leftrightarrow l'_j) \wedge \neg p_i \wedge l'_i \wedge \bigwedge_{i < j < 2n} (p_j \wedge \neg l'_j) \right)$ means ‘the ordinate y of the tile location of w and the ordinate y' of the tile location of w' are such that $y' = y + 1$ ’.

We leave the reader prove that we can tile a $k \times k$ grid with the tile types of T and t_0 being placed onto $(0, 0)$ iff (ϕ, ϕ', ϕ'') is satisfiable.

□

5 Implementation

The tableau method described in Remark 3.1 of Section 3.2 is implemented in LoTREC Scheme (a variant of LoTREC [Gasquet et al., 2005] written in Scheme). Contrary to LoTREC, the system of LoTREC Scheme allows the name of a node to be a couple (w, w') and this functionality is suitable for our tableau rules. You can find the implementation at the following web page:

<http://www.irisa.fr/prive/fschwarz/publications/m4m2011/>.

6 Concluding remarks and related work

This paper contributes to the proof theory and the study of the computational complexity of DEL, which has been rather neglected so far. Indeed, most work in this field has often been inspired or applied to logico-philosophical puzzles such as for example the muddy children riddle, Fitch paradox, or Moorean sentences. Up to our knowledge, the only known results of computational complexity are the PSPACE-completeness of the satisfiability problem for public announcement logic [Lutz, 2006] and the polynomial time upper bound of the model-checking problem for public announcement logic. As for proof theory, a sound and complete sequent calculus for DEL has been developed in [Baltag et al., 2004], yet in an algebraic setting. Because of this different setting, the comparison cannot be systematic, but, unlike our DEL-sequents, 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 δ . Some tableau methods have been proposed for DEL, but only for public announcement logic [Balbiani et al., 2010, de Boer, 2007] and hybrid public announcement logic [Hansen, 2010]. A terminating tableau method has also been proposed for the full BMS framework in [Hansen, 2010] by encoding the reduction axioms as tableau rules. However, none of these tableau methods can somehow address the three questions raised in the introduction, because the BMS language of [Baltag and Moss, 2004] 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.

References

- [Aucher, 2011] Aucher, G. (2011). DEL-sequents for progression. *Journal of Applied Non-Classical Logics*. To appear.
- [Balbiani et al., 2010] Balbiani, P., van Ditmarsch, H., Herzig, A., and de Lima, T. (2010). Tableaux for public announcement logic. *Journal of Logic and Computation*, 20(1):55–76.
- [Baltag et al., 2004] Baltag, A., Coecke, B., and Sadrzadeh, M. (2004). Algebra and sequent calculus for epistemic actions. In *Proceedings of Workshop on Logic and Communication in Multi-Agent Systems (LCMAS'04)*, pages 60–78.
- [Baltag and Moss, 2004] Baltag, A. and Moss, L. (2004). Logic for epistemic programs. *Synthese*, 139(2):165–224.
- [Baltag et al., 1998] Baltag, A., Moss, L., and Solecki, S. (1998). The logic of common knowledge, public announcement, and private suspicions. In Gilboa, I., editor, *Proceedings of the 7th conference on theoretical aspects of rationality and knowledge (TARK98)*, pages 43–56.
- [Baltag et al., 1999] Baltag, A., Moss, L., and Solecki, S. (1999). The logic of public announcements, common knowledge and private suspicions. Technical report, Indiana University.
- [Boas, 1997] Boas, P. (1997). The convenience of tilings. In *Complexity, Logic, and Recursion Theory*, pages 331–363. Marcel Dekker Inc.
- [de Boer, 2007] de Boer, M. (2007). KE tableaux for public announcement logic. In *Proceedings of Formal Approaches to Multi-Agent Systems Workshop (FAMAS 07)*, Durham UK.
- [Gasquet et al., 2005] Gasquet, O., Herzig, A., Longin, D., and Saade, M. (2005). LoTREC: Logical Tableaux Research Engineering Companion. In Beckert, B., editor, *International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX 2005)*, Koblenz, Germany, 14/09/05-17/09/05, pages 318–322. Springer Verlag, LNCS 3702.
- [Hansen, 2010] Hansen, J. U. (2010). Terminating tableaux for dynamic epistemic logic. *Electronic Notes in Theoretical Computer Science*, 262:141–156.
- [Lutz, 2006] Lutz, C. (2006). Complexity and succinctness of public announcement logic. In Nakashima, H., Wellman, M. P., Weiss, G., and Stone, P., editors, *5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006)*, Hakodate, Japan, May 8-12, 2006, pages 137–143. ACM.
- [van Ditmarsch et al., 2007] van Ditmarsch, H., van der Hoek, W., and Kooi, B. (2007). *Dynamic Epistemic Logic*, volume 337 of *Synthese library*. Springer.