

# Modal logics for reasoning about infinite unions and intersections of binary relations

Natasha Alechina\*, Philippe Balbiani<sup>†</sup> and Dmitry Shkatov<sup>‡</sup>

February 24, 2012

## Abstract

We consider multi-modal logic  $K_\omega$  (with countably infinite number of basic modalities) extended with additional modalities  $\langle \cup \rangle$  and  $\langle \cap \rangle$  corresponding to the union and intersection of all basic modalities. We present complete and sound axiomatic systems and polynomial-space terminating tableau-based decision procedures for the basic logic in this language,  $K_{\cup\cap}$  and its deterministic counterpart  $DK_{\cup\cap}$ . We also show that  $K_{\cup\cap}$  admits filtration, which can be used independently of our tableaux to establish its decidability.

Keywords: modal logic, infinite intersections and unions, axiomatic systems, filtration, tableau-based decision procedures.

## 1 Introduction

In this paper, we consider multi-modal logics that are interpreted over labelled graphs where the set of edge labels is countably infinite and that contain modalities corresponding to accessibility by a labelled edge ( $\langle i \rangle \varphi$  means ‘a node satisfying  $\varphi$  is reachable by an edge with label  $i$ ’), the union of all edge labels ( $\langle \cup \rangle \varphi$  means ‘a node satisfying  $\varphi$  is accessible by an edge

---

\*School of Computer Science, University of Nottingham, UK

<sup>†</sup>IRIT, France

<sup>‡</sup>School of Computer Science, University of the Witwatersrand, South Africa.

with *some* label’) and the intersection of all edge labels ( $\langle \cap \rangle \varphi$  means ‘a node satisfying  $\varphi$  is accessible by *all* edge labels’).

The original motivation for our work comes from using modal logic for expressing reachability constraints on web-based or semistructured data, with edge labels corresponding to URLs or link names [2, 3]. The constraints could be for example of the form ‘if a data item is reachable by a path described by any label followed by Publications label, then it is also reachable by a path MainPage.Publications’. For expressing such constraints, it turned out to be very useful to have a ‘wildcard’ or ‘some label’ modality  $\#$ , defined as the union of all possible labels; for example, using this modality, the first path above can be described as  $\#.Publications$ . The  $\langle \# \rangle$  modality (in this paper, we denote it by  $\langle \cup \rangle$ ) was introduced to express reachability by ‘some’ link. It corresponds to the (infinite) union of edge labels or to an existential quantifier over edge labels. In the short version of this paper ([1], which appeared in proceedings of HyLo 2007) we also introduced the universal quantifier over edge labels (or the infinite intersection of all edges) as well. Here we show that adding the infinite intersection does not increase the complexity of the original logic  $K_\omega$  with  $\langle \cup \rangle$ . We denote  $K_\omega$  extended with  $\langle \cup \rangle$  and  $\langle \cap \rangle$  by  $K_{\cup \cap}$ .

A related logic  $K_\omega^{\cap, \cup}$  which is  $K_\omega$  with finite unions and intersections of basic modalities was studied in [10]. Their results can be easily adapted to prove PSPACE decidability of  $K_{\cup \cap}$ , but in general the expressive power of the two logics is different:  $K_{\cup \cap}$  cannot express reachability by a finite intersection of modalities (and can only express reachability by a union at the cost of an exponential blow-up) and  $K_\omega^{\cap, \cup}$  cannot express the union and intersection of all basic modalities. Propositional dynamic logic with union and intersection of basic modalities was studied in [4]; it has considerably more complex axiomatization than  $K_{\cup \cap}$ .

We also consider deterministic version of  $K_{\cup \cap}$ , namely the set of formulas valid in models where for each state and label there is at most one edge with this label from the state. This was originally motivated by the interpretation of edge labels as URLs (which are deterministic) and raises some interesting technical challenges.

The paper is structured as follows. In Section 2, we introduce logics  $K_{\cup \cap}$  and  $DK_{\cup \cap}$ . In Section 3, we present axiomatic systems for these logics and prove their soundness and completeness. In Section 4, we show how to do filtration for  $K_{\cup \cap}$ . In Section 5, we present tableau-based decision procedure for  $K_{\cup \cap}$  and, in Section 6, we do the same for  $DK_{\cup \cap}$ . Both procedures run

in polynomial space, so we establish PSPACE-completeness of both logics (PSPACE-hardness is easily established).

## 2 Logics $\mathbf{K}_{\cup\cap}$ and $\mathbf{DK}_{\cup\cap}$

### 2.1 Syntax and standard semantics

Consider a propositional modal language  $\mathcal{L}_{\cup\cap}^I$  containing a countable set  $\mathbf{Par}$  of propositional parameters; a sufficient repertoire of classical propositional connectives, say  $\neg$  (“not”) and  $\vee$  (“or”); for every element  $i$  of a countable set  $I$  of *modal indices* or *labels*, a modal operator  $\langle i \rangle$ ; and finally a modal operators  $\langle \cup \rangle$  and  $\langle \cap \rangle$ . Formulas of  $\mathcal{L}_{\cup\cap}^I$  are thus defined by the following BNF expression:

$$\varphi := p \mid \neg\varphi \mid (\varphi_1 \vee \varphi_2) \mid \langle i \rangle \varphi \mid \langle \cup \rangle \varphi \mid \langle \cap \rangle \varphi,$$

where  $p \in \mathbf{Par}$  and  $i \in I$ . All the above-mentioned modal operators are “diamonds”, that is to say, they say that *there exists* a state reachable by an edge labeled with  $i$  (or reachable by the union or intersection of basic accessibility relations). Other Boolean connectives, as well as the “boxes”  $[i]$ ,  $[\cup]$ , and  $[\cap]$ , are defined in the usual way, namely  $[\pi]\varphi = \neg\langle\pi\rangle\neg\varphi$  for  $\pi \in I \cup \{\cup, \cap\}$ . To enhance readability of formulas, we usually omit the parentheses associated with binary connectives when it does not result in ambiguity. Given a language  $\mathcal{L}$  and a formula  $\varphi$ , we write  $\varphi \in \mathcal{L}$  to mean that  $\varphi$  is a formula of  $\mathcal{L}$ .

Formulas of  $\mathcal{L}_{\cup\cap}^I$  are evaluated on  $\mathcal{L}_{\cup\cap}^I$ -models.

**Definition 2.1** *An  $\mathcal{L}_{\cup\cap}^I$ -model is a tuple  $\mathcal{M} = (W, \{\mathcal{R}_i\}_{i \in I}, \mathcal{R}_{\cup}, \mathcal{R}_{\cap}, V)$ , where*

1.  $W \neq \emptyset$  is a set of nodes;
2.  $\mathcal{R}_i \subseteq W \times W$ , for every  $i \in I$ ;
3.  $\mathcal{R}_{\cup} = \bigcup_{i \in I} \mathcal{R}_i$ ;
4.  $\mathcal{R}_{\cap} = \bigcap_{i \in I} \mathcal{R}_i$ ;
5.  $V$  is a valuation function from  $\mathbf{Par}$  into  $2^W$ , the power-set of  $W$ . (Intuitively,  $V(p)$  is the set of nodes where  $p \in \mathbf{Par}$  is declared to be “true”.)

A model  $\mathcal{M}$  is said to be deterministic if, for every  $w \in W$  and every  $i \in I$ , there is at most one  $v$  such that  $w\mathcal{R}_i v$ .

Given a binary relation  $\mathcal{R}$  on  $W$  and a pair of nodes  $w, v \in W$  such that  $(w, v) \in \mathcal{R}$ , we call an ordered pair  $(w, v)$  an  $\mathcal{R}$ -edge.

The satisfaction relation between models, nodes, and formulas is defined in the standard way. In particular,

- $\mathcal{M}, w \Vdash \langle i \rangle \varphi$  iff  $\exists v \in W (w\mathcal{R}_i v$  and  $\mathcal{M}, v \Vdash \varphi)$ ;
- $\mathcal{M}, w \Vdash \langle \cup \rangle \varphi$  iff  $\exists v \in W (w\mathcal{R}_\cup v$  and  $\mathcal{M}, v \Vdash \varphi)$ ;
- $\mathcal{M}, w \Vdash \langle \cap \rangle \varphi$  iff  $\exists v \in W (w\mathcal{R}_\cap v$  and  $\mathcal{M}, v \Vdash \varphi)$ .

Satisfiability and validity in a model as well as a class of models (for example, the class of all deterministic models) are defined in the usual way.

It is well-known (see, for example, [12]) that intersection of (even two) relations is not modally definable — hence, the introduction of  $\langle \cap \rangle$  into a language with even a finite number of modal indices increases the expressive power of the language. Here we briefly state the undefinability of  $\langle \cap \rangle$  in the language  $\mathcal{L}_\cup^I$ , which contains  $\langle i \rangle$  ( $i \in I$ ) and  $\langle \cup \rangle$ .

**Theorem 2.2** *The modal operator  $\langle \cap \rangle$  is not (explicitly) definable in  $\mathcal{L}_\cup^I$  with infinite  $I$ .*

**Proof.** Suppose there is a formula  $\psi$  in  $\mathcal{L}_\cup^I$  such that  $\langle \cap \rangle p$  is equivalent to  $\psi$ . Consider  $\mathcal{M} = (W, \{\mathcal{R}_i\}_{i \in I}, \mathcal{R}_\cup, V)$  and  $\mathcal{M}' = (W', \{\mathcal{R}'_i\}_{i \in I}, \mathcal{R}'_\cup, V')$ , where  $W = W' = \{0\} \cup I$  (assume  $0 \notin I$ ),  $x\mathcal{R}_i y$  iff  $x = 0, y \in I$  and  $y = i$ ,  $x\mathcal{R}'_i y$  iff  $x = 0$  and  $y \in I$ ,  $V(p) = V'(p) = I$ . We have  $\mathcal{M}, 0 \not\models \langle \cap \rangle p$  and  $\mathcal{M}', 0 \models \langle \cap \rangle p$ . Thus  $\mathcal{M}, 0 \not\models \psi$  and  $\mathcal{M}', 0 \models \psi$ . But  $\mathcal{M}, 0$  and  $\mathcal{M}', 0$  satisfy the same formulas in  $\mathcal{L}_\cup^I$ : a contradiction.  $\square$

We next show that, assuming that the set of indices  $I$  is *infinite*, the introduction of  $\langle \cup \rangle$  enriches a language with modalities  $\langle i \rangle$  ( $i \in I$ ) and  $\langle \cap \rangle$ ; we denote this language by  $\mathcal{L}_\cap^I$ .

**Theorem 2.3** *The modal operator  $\langle \cup \rangle$  is not (explicitly) definable in  $\mathcal{L}_\cap^I$  with infinite  $I$ .*

**Proof.** For the sake of contradiction, assume that, for some formula  $\psi \in \mathcal{L}_\cap^I$ ,  $\langle \cup \rangle p \leftrightarrow \psi$  is valid. Denote the set of modal indices occurring in a formula  $\chi \in \mathcal{L}_\cap^I$  by  $\text{Ind}(\chi)$ . Consider the  $\mathcal{L}_\cap^I$ -models,  $\mathcal{M} = (W, \{\mathcal{R}_i\}_{i \in I}, \mathcal{R}_\cap, V)$  and  $\mathcal{M}' = (W, \{\mathcal{R}'_i\}_{i \in I}, \mathcal{R}'_\cap, V')$ , where

- $W = \{x_0, x_1\}$ ;
- $\mathcal{R}_i = (x_0, x_1)$  for some  $i \in I$  such that  $i \notin \text{Ind}(\psi)$ , whereas  $\mathcal{R}_j = \emptyset$  for all other  $j \in I$ ;
- $\mathcal{R}'_i = \emptyset$ , for all  $i \in I$ ;
- $V(p) = \{x_1\}$ ;
- $V'(p) = \{x_1\}$ .

It is then straightforward to show by induction on the structure of  $\psi$  that  $\mathcal{M}, x_0 \Vdash \chi$  iff  $\mathcal{M}', x_0 \Vdash \chi$  holds for every subformula  $\chi$  of  $\psi$  (including  $\psi$  itself). This contradicts the fact that  $\mathcal{M}, x_0 \Vdash \langle \cup \rangle p$  and  $\mathcal{M}', x_0 \not\Vdash \langle \cup \rangle p$ .  $\square$

**Remark 2.4** *If, on the other hand,  $I$  is finite, say  $I = \{i_1, \dots, i_n\}$ , then  $\langle \cup \rangle \varphi$  is definable as  $\langle i_1 \rangle \varphi \vee \dots \vee \langle i_n \rangle \varphi$ . Therefore, we assume through the paper that the set of modal indices  $I$  is (countably) infinite.*

## 2.2 Pseudo-models

In both proving completeness of an axiomatic system and designing a tableau-based decision procedure for  $\mathbf{K}_{\cup\cap}$ , we will use  $\mathcal{L}_{\cup\cap}^I$ -pseudo-models, which we now define and show to be satisfiability-wise equivalent to models.

**Definition 2.5** *An  $\mathcal{L}_{\cup\cap}^I$ -pseudo-model is a tuple  $\mathcal{M} = (W, \{\mathcal{R}_i\}_{i \in I}, \mathcal{R}_\cup, \mathcal{R}_\cap, V)$ , where*

1.  $W \neq \emptyset$ ;
2.  $\mathcal{R}_i \subseteq W \times W$ , for every  $i \in I$ ;
3.  $\mathcal{R}_\cup \supseteq \bigcup_{i \in I} \mathcal{R}_i$ ;
4.  $\mathcal{R}_\cap \subseteq \bigcap_{i \in I} \mathcal{R}_i$ ;

5.  $V$  is a function from  $\mathbf{Par}$  into  $2^W$ .

Thus, pseudo-models differ from models only in the conditions imposed on relations  $\mathcal{R}_\cup$  and  $\mathcal{R}_\cap$ . The satisfaction relation and satisfiability of formulas are defined for pseudo-models exactly as for models.

As every model is a pseudo-model, it is obvious that if  $\varphi \in \mathcal{L}_{\cup\cap}^I$  is satisfiable in a model, then it is satisfiable in a pseudo-model. For our purposes in the present paper, the other direction is of far greater importance. As before, we denote by  $\text{Ind}(\varphi)$  the set of modal indices occurring in a formula  $\varphi$ .

**Lemma 2.6** *Let  $\varphi \in \mathcal{L}_{\cup\cap}^I$  be satisfiable in an  $\mathcal{L}_{\cup\cap}^I$ -pseudo-model. Then,  $\varphi$  is satisfiable in an  $\mathcal{L}_{\cup\cap}^I$ -model.*

**Proof.** Let  $\varphi$  be satisfied in a pseudo-model  $\mathcal{M} = (W, \{\mathcal{R}_i\}_{i \in I}, \mathcal{R}_\cup, \mathcal{R}_\cap, V)$  at node  $w$ . We show how to convert  $\mathcal{M}$  into a bona-fide model  $\mathcal{M}'$  with the set of nodes  $W$  such that  $\varphi$  is still satisfied at  $w$  in  $\mathcal{M}'$ .

Let  $\{I_1, I_2\}$  be an arbitrary partition of the set  $I \setminus \text{Ind}(\varphi)$ ; as  $I$  is infinite, such a partition obviously exists. (We remind the reader that, according to the definition of partition, both  $I_1$  and  $I_2$  are non-empty). Let  $\mathcal{M}' = (W, \{\mathcal{R}'_i\}_{i \in I}, \mathcal{R}_\cup, \mathcal{R}_\cap, V)$  be the model obtained from  $\mathcal{M}$  as follows. For all  $i \in I$ , we stipulate that

- if  $i \in \text{Ind}(\varphi)$ , then  $\mathcal{R}'_i = \mathcal{R}_i$ ;
- if  $i \in I_1$ , then  $\mathcal{R}'_i = \mathcal{R}_\cup$ ,
- if  $i \in I_2$ , then  $\mathcal{R}'_i = \mathcal{R}_\cap$ .

We now show that  $\mathcal{M}'$  is an  $\mathcal{L}_{\cup\cap}$ -model, i.e. equalities

- $\mathcal{R}_\cup = \bigcup_{i \in I} \mathcal{R}'_i$
- $\mathcal{R}_\cap = \bigcap_{i \in I} \mathcal{R}'_i$

hold in  $\mathcal{M}'$ . We start by noting that in  $\mathcal{M}$ , and hence in  $\mathcal{M}'$ , we have

$$(\ddagger) \quad \mathcal{R}_\cap \subseteq \mathcal{R}_\cup.$$

Now, consider the first of the two equalities above. Let  $(w, v) \in \mathcal{R}_\cup$ ; then,  $(w, v) \in \mathcal{R}'_i$  for some  $i \in I_1 \subset I$ ; thus,  $\mathcal{R}_\cup \subseteq \bigcup_{i \in I} \mathcal{R}'_i$ . Let, on the other hand,  $(w, v) \in \bigcup_{i \in I} \mathcal{R}'_i$ , i.e.  $(w, v) \in \mathcal{R}'_i$  for some (fixed)  $i \in I$ . There are three

cases to consider. If  $i \in \text{Ind}(\varphi)$ , then  $(w, v) \in \mathcal{R}_i$  and hence  $(w, v) \in \mathcal{R}_\cup$  due to condition 3 of definition 2.5. If  $i \in I_1$ , then the conclusion immediately follows from the definition of relations  $\mathcal{R}'_i$ . Lastly, if  $i \in I_2$ , then the desired conclusion follows from the definition of relations  $\mathcal{R}'_i$  together with  $(\ddagger)$ .

As for the second equality,  $(w, v) \in \bigcap_{i \in I} \mathcal{R}'_i$  implies that  $(w, v) \in \mathcal{R}'_i$  for some  $i \in I_2$  and hence  $\bigcap_{i \in I} \mathcal{R}'_i \subseteq \mathcal{R}_\cap$ . Suppose, on the other hand, that  $(w, v) \in \mathcal{R}_\cap$ . Then, that  $(w, v) \in \mathcal{R}'_i$  for every  $i \in I_2$  is immediate; that  $(w, v) \in \mathcal{R}'_i$  for every  $i \in I_1$  follows from  $(\ddagger)$ ; lastly, that  $(w, v) \in \mathcal{R}'_i$  for every  $i \in \text{Ind}(\varphi)$  follows from condition 4 of definition 2.5. Therefore,  $\mathcal{R}_\cap \subseteq \bigcap_{i \in I} \mathcal{R}'_i$ .

Thus,  $\mathcal{M}'$  is an  $\mathcal{L}_{\cup\cap}$ -model. To complete the proof of the lemma, all that remains to show is that  $\varphi$  is satisfiable in  $\mathcal{M}'$ . Well, as  $\mathcal{M}$  and  $\mathcal{M}'$  are identical with respect to all  $\mathcal{R}_i$  for every  $i \in \text{Ind}(\varphi)$  and relations associated with  $\langle \cup \rangle$  and  $\langle \cap \rangle$  are identical in both models, it immediately follows that  $\mathcal{M}', w \Vdash \varphi$ , and thus we are done.  $\square$

Analogously, when proving completeness of an axiomatic system and designing a tableau-based decision procedure for  $\text{DK}_{\cup\cap}$ , we will use deterministic  $\mathcal{L}_{\cup\cap}^I$ -pseudo-models, which we now define and show to be satisfiability-wise equivalent to deterministic models.

**Definition 2.7** *A deterministic  $\mathcal{L}_{\cup\cap}^I$ -pseudo-model is a tuple  $\mathcal{M} = (W, \{\mathcal{R}_i\}_{i \in I}, \mathcal{R}_\cup, \mathcal{R}_\cap, V)$ , where*

1.  $W \neq \emptyset$ ;
2. for all  $w, u, v \in W$  and all  $i \in I$ , if  $w\mathcal{R}_i v$  and  $w\mathcal{R}_i u$ , then  $v = u$ ;
3.  $\mathcal{R}_\cup \supseteq \bigcup_{i \in I} \mathcal{R}_i$ ;
4.  $\mathcal{R}_\cap \subseteq \bigcap_{i \in I} \mathcal{R}_i$ ;
5. for all  $w, u, v \in W$ , if  $w\mathcal{R}_\cup v$  and  $w\mathcal{R}_\cap u$ , then  $v = u$ .

Once again, as every deterministic model is a deterministic pseudo-model, if  $\varphi \in \mathcal{L}_{\cup\cap}$  is satisfiable in a deterministic model, it is also satisfiable in a deterministic pseudo-model. We now show that the opposite also holds. We say that a (pseudo-)model is countable if its set of nodes is.

**Lemma 2.8** *Let  $\varphi \in \mathcal{L}_{\cup\cap}$  be satisfiable in a deterministic pseudo-model. Then, it is satisfiable in a countable deterministic pseudo-model.*

**Proof.** First, it is easy to check that the class of deterministic pseudo-models is first-order axiomatizable.

Now, suppose that  $\varphi \in \mathcal{L}_{\cup\cap}$  is satisfiable in a deterministic pseudo-model  $\mathcal{M}$ . We can obviously assume that  $\mathcal{M}$  is infinite seeing that if, initially,  $\mathcal{M}$  contains finitely many possible worlds, then it is always possible to add to  $\mathcal{M}$  an infinite number of inaccessible worlds. The standard translation (see, for example, [6, Chapter 2]) of  $\varphi$ , denoted  $\text{ST}(\varphi)$ , is satisfied in  $\mathcal{M}$  viewed as a first-order model. As  $\text{ST}(\varphi)$  is a first-order formula and  $\mathcal{M}$  belongs to a first-order definable class of models (i.e., the class of deterministic pseudo-models), in virtue of the downward Skolem-Löwenheim theorem for first-order logic,  $\text{ST}(\varphi)$  is satisfiable in a countable model of this class, say  $\mathcal{M}'$ . But then  $\varphi$  is also satisfied in  $\mathcal{M}'$ , and thus we are done.  $\square$

**Lemma 2.9** *Let  $\varphi \in \mathcal{L}_{\cup\cap}$  be satisfiable in a deterministic pseudo-model. Then, it is satisfiable in a deterministic model.*

**Proof.** Suppose that  $\varphi$  is satisfiable in a deterministic pseudo-model. Then, in virtue of lemma 2.8,  $\varphi$  is satisfiable in a countable deterministic pseudo-model  $\mathcal{M}$  with the set of nodes  $W$  at, say,  $w \in W$ .

For an  $x \in W$  and a binary relation  $\mathcal{R}$  on  $W$ , denote by  $\mathcal{R}(x)$  the set  $\{y \in W \mid x\mathcal{R}y\}$ . Now, notice that, in deterministic pseudo-models, the relation  $\mathcal{R}_{\cap}$  is deterministic, and moreover, for all  $x \in W$ , if  $\mathcal{R}_{\cap}(x) \neq \emptyset$  then  $\mathcal{R}_{\cap}(x) = \mathcal{R}_{\cup}(x) = \{y\}$  for some  $y \in W$ .

Let  $\{I_1, I_2\}$  be a partition of the set  $I \setminus \text{Ind}(\varphi)$  such that  $I_1$  is infinite (as  $I$  is infinite, such a partition clearly exists). We can, therefore, assume that elements of  $I_1$  are positive integers. Now, let  $\mathcal{M}' = (W, \{\mathcal{R}'_i\}_{i \in I}, \mathcal{R}_{\cup}, \mathcal{R}_{\cap}, V)$  be the model obtained from  $\mathcal{M}$  as follows. For all  $i \in I$ , we stipulate that

- if  $i \in \text{Ind}(\varphi)$ , then  $\mathcal{R}'_i = \mathcal{R}_i$ ;
- if  $i \in I_1$  then we define  $\mathcal{R}'_i$  as follows. For every  $x \in W$ , let  $\{y_1, y_2, \dots\}$  be an enumeration of  $\mathcal{R}_{\cup}(x)$ . Note that this enumeration can be finite or even empty. We stipulate that
  - if  $\mathcal{R}_{\cap}(x) = \emptyset$  and  $|\mathcal{R}_{\cup}| \geq i$ , then  $\mathcal{R}'_i(x) = \{y_i\}$ ;
  - if  $\mathcal{R}_{\cap}(x) = \emptyset$  and  $|\mathcal{R}_{\cup}| < i$ , then  $\mathcal{R}'_i(x) = \emptyset$ ;
  - otherwise,  $\mathcal{R}'_i(x) = \mathcal{R}_{\cap}(x)$ ;

- if  $i \in I_2$  then  $\mathcal{R}'_i = \mathcal{R}_\cap$ .

Clearly, all so defined relations  $\mathcal{R}'_i$  are deterministic. It is, moreover, straightforward to check that the following hold:

- $\mathcal{R}_\cup = \bigcup_{i \in I} \mathcal{R}'_i$ ;
- $\mathcal{R}_\cap = \bigcap_{i \in I} \mathcal{R}'_i$ ;
- $\mathcal{M}', w \Vdash \varphi$ .

Therefore,  $\mathcal{M}'$  is a deterministic  $\mathcal{L}_{\cup\cap}$ -model satisfying  $\varphi$ , and we are done.  $\square$

Let us denote the set of formulas true in the class of all  $\mathcal{L}_{\cup\cap}$ -models by  $\mathbf{K}_{\cup\cap}$  and the set of formulas true in the class of all deterministic  $\mathcal{L}_{\cup\cap}$ -models as  $\mathbf{DK}_{\cup\cap}$ . It is straightforward to ascertain that both  $\mathbf{K}_{\cup\cap}$  and  $\mathbf{DK}_{\cup\cap}$  are normal modal logics. We now turn to the question of designing axiomatic systems for these logics.

### 3 Axiomatic systems

We start with the axiomatic system for  $\mathbf{K}_{\cup\cap}$ . In the axiom schemata and inference rules below,  $\pi$  stands for an arbitrary member of the set  $I \cup \{\cup, \cap\}$ , and as before,  $i \in I$ :

*Axiom schemata:*

- (A0) All classical tautologies;
- (K)  $[\pi](\varphi \rightarrow \psi) \rightarrow ([\pi]\varphi \rightarrow [\pi]\psi)$ ;
- (ER)  $\langle i \rangle \varphi \rightarrow \langle \cup \rangle \varphi$ ;
- (RE)  $\langle \cap \rangle \varphi \rightarrow \langle i \rangle \varphi$

*Inference rules:*

$$\text{(MP)} \frac{\vdash \varphi \rightarrow \psi, \vdash \varphi}{\vdash \psi}; \quad \text{(N)} \frac{\vdash \varphi}{\vdash [\pi]\varphi}.$$

Note in particular that we do not require an inference rule for  $\langle \cup \rangle$  corresponding to the standard rule for the existential quantifier  $\exists$  in axiomatic systems for first-order logic, despite the obvious similarities between  $\langle \cup \rangle$  and  $\exists$  (both can be thought of as infinite disjunctions).

The axiomatic system for  $\text{DK}_{\cup\cap}$  can be obtained by adding to the axiom schemata and rules above the following “axiom of determinism” (also referred to in the literature as “axiom of functionality”):

$$\text{(D1)} \quad \langle i \rangle \varphi \rightarrow [i]\varphi;$$

plus the following axiom:

$$\text{(D2)} \quad \langle \cap \rangle \varphi \rightarrow [\cup]\varphi.$$

Notice that absense in both axiomatic systems of the “obvious” inference rule, analogous to the rule for the existential quantifier in the axiomatic systems for first-order logic, used in [3]:

$$\text{(EL)} \quad \frac{\vdash \langle i \rangle \varphi \rightarrow \psi}{\vdash \langle \cup \rangle \varphi \rightarrow \psi}, \text{ provided } i \text{ does not occur in } \psi.$$

We have been able to emilinate this rule without affecting the completeness of our axiomatic systems due to the pseudo-model characterization of  $\text{K}_{\cup\cap}$  and  $\text{DK}_{\cup\cap}$  in sciton 2.2. We should stress that the application of any standard techniques in modal logic would result in producing an axiomatic system with the rule **(EL)**. Thus, developments in section 2.2 are crucial to the more economical axiomatic characterization of  $\text{K}_{\cup\cap}$  and  $\text{DK}_{\cup\cap}$  provided in the present section.

In what follows, we use the names  $\text{K}_{\cup\cap}$  and  $\text{DK}_{\cup\cap}$  both for the logics as sets of formulas valid in a respective class of models and for the corresponding axiomatic systems, relying on the context for disambiguation.

Given an axiomatic system  $\text{L}$ , the notion of an  $\text{L}$ -proof is defined in the usual way as a finite sequence of formulas each of which is either an instance of an axiom schema or obtained from the previous members of the sequence by one of the inference rules. A formula  $\varphi$  is  $\text{L}$ -provable if there exists an  $\text{L}$ -proof containing  $\varphi$  as a last member. A formula  $\varphi$  is  $\text{L}$ -consistent if  $\neg\varphi$  is not  $\text{L}$ -provable. A set of formulas  $\Gamma$  is  $\text{L}$ -consistent if the negation of a conjunction of any finite subset of  $\Gamma$  is not  $\text{L}$ -provable.

We now show that axiomatic systems  $\text{K}_{\cup\cap}$  and  $\text{DK}_{\cup\cap}$  are sound and (weakly) complete with respect to the class of all (respectively, deterministic)  $\mathcal{L}_{\cup\cap}$ -models. We briefly remind the reader what soundness and completeness mean in the context of axiomatic systems. Let  $\text{L}$  be an axiomatic system and  $\text{M}$  be class of models. Then,

- $L$  is *sound* with respect to  $M$  if every  $L$ -provable formula is valid in every model in  $M$ ;
- $L$  is *strongly complete* with respect to  $M$  if every  $L$ -consistent set of formulas is satisfiable in some model in  $M$ ;
- $L$  is *weakly complete* with respect to  $M$  if every  $L$ -consistent formula is satisfiable in some model in  $M$ .

**Theorem 3.1**  $K_{\cup\cap}$  is sound with respect to the class of all  $\mathcal{L}_{\cup\cap}$ -models and  $DK_{\cup\cap}$  is sound with respect to the class of all deterministic  $\mathcal{L}_{\cup\cap}$ -models.

**Proof.** Routine induction on the length of the proof of a formula.  $\square$

As for completeness, we first notice that neither of the logics is strongly complete with respect to *any* class of models. Indeed, it is not hard to see that both  $K_{\cup\cap}$  and  $DK_{\cup\cap}$  are non-compact. Consider the set  $\Gamma = \{\langle \cup \rangle p\} \cup \{\neg \langle i \rangle p : i \in I\}$ ; clearly, every finite subset of  $\Gamma$  is satisfiable, but  $\Gamma$  itself is not. As a non-compact logic cannot be strongly complete (with respect to any class of models), neither  $K_{\cup\cap}$  nor  $DK_{\cup\cap}$  is. Therefore, we are going to prove weak completeness for both logics.

**Theorem 3.2**  $K_{\cup\cap}$  is weakly complete with respect to the class of all  $\mathcal{L}_{\cup\cap}$ -models.

**Proof.** Consider a  $K_{\cup\cap}$ -consistent formula  $\varphi$ . We want to show that  $\varphi$  is satisfiable in an  $\mathcal{L}_{\cup\cap}$ -model.

Let  $\mathcal{M}^c = (W^c, \{\mathcal{R}_i^c\}_{i \in I}, \mathcal{R}_\cup, \mathcal{R}_\cap, V^c)$  be the canonical model for  $K_{\cup\cap}$ , defined in the standard way. As all our modalities are “normal”, we can, using the standard truth lemma argument, show that there exists  $w \in W^c$  such that  $\mathcal{M}^c, w \Vdash \varphi$ . Now, it is straightforward to check that formulas **(ER)** and **(RE)** are canonical for the following conditions, respectively:

$$(\dagger) \mathcal{R}_\cup \supseteq \bigcup_{i \in I} \mathcal{R}_i$$

$$(\ddagger) \mathcal{R}_\cap \subseteq \bigcap_{i \in I} \mathcal{R}_i.$$

Therefore,  $\mathcal{M}^c$  is an  $\mathcal{L}_{\cup\cap}$ -pseudo-model satisfying  $\varphi$ . The conclusion of the theorem then immediately follows from lemma 2.6.  $\square$

Now, we turn to completeness for  $DK_{\cup\cap}$ .

**Theorem 3.3** *DK<sub>U $\cap$</sub>  is weakly complete with respect to the class of all deterministic  $\mathcal{L}_{U\cap}$ -models.*

**Proof.** Let  $\varphi$  be a DK<sub>U $\cap$</sub> -consistent formula. We want to show that  $\varphi$  is satisfiable in a deterministic  $\mathcal{L}_{U\cap}$ -model.

Using a standard completeness-via-canonicity argument, we can show that  $\varphi$  is satisfiable in a deterministic pseudo-model. The desired conclusion then immediately follows from lemma 2.9.  $\square$

## 4 Admissibility of filtration for $\mathbf{K}_{U\cap}$

Before we present tableau-based decision procedures for our logics, we study the question of how they relate to the traditional method of proving decidability of modal logics — establishing a finite model property through filtration. The question of whether logics containing an axiom of determinism admit filtration is notoriously difficult, see [5], for example, in the case of deterministic propositional dynamic logic. Therefore, in the present section, we only confine our attention to providing a suitable filtration for  $\mathbf{K}_{U\cap}$ .

Consider a formula  $\varphi$  satisfiable in an  $\mathcal{L}_{U\cap}^I$ -pseudo-model (we know from lemma 2.6 that this is equivalent to  $\mathcal{L}_{U\cap}^I$ -satisfiability). Let  $\Gamma_\varphi$  be the least set of formulas such that

- $\varphi \in \Gamma_\varphi$ ;
- $\Gamma_\varphi$  is subformula-closed;
- if  $[\cup]\psi \in \Gamma_\varphi$  and  $i \in \text{Ind}(\varphi)$ , then  $[i]\psi \in \Gamma_\varphi$ ;
- if  $[i]\psi \in \Gamma_\varphi$ , then  $[\cap]\psi \in \Gamma_\varphi$ ;
- if  $[\cup]\psi \in \Gamma_\varphi$ , then  $[\cap]\psi \in \Gamma_\varphi$ .

Clearly,  $\Gamma_\varphi$  is finite — more precisely, its size is polynomial in the length of  $\varphi$ . Notice also that, for all  $i \in I$ , if  $[i]\psi \in \Gamma_\varphi$  then  $i \in \text{Ind}(\varphi)$ . Now, let  $\equiv_{\Gamma_\varphi}$  be the equivalence relation on  $W$  defined by the following condition:

$$x \equiv_{\Gamma_\varphi} y \text{ iff, for all } \psi \in \Gamma_\varphi, \mathcal{M}, x \Vdash \psi \text{ iff } \mathcal{M}, y \Vdash \psi.$$

For all  $x \in W$ , let  $|x|$  be the equivalence class of  $x$  modulo  $\equiv_{\Gamma_\varphi}$ . We define a structure  $\mathcal{M}' = (W', \{\mathcal{R}'_i\}_{i \in I}, \mathcal{R}'_\cup, \mathcal{R}'_\cap, V')$  as follows:

1.  $W' = \{ |x| \mid x \in W \}$ ;
2.  $(|x|, |y|) \in \mathcal{R}'_i$  iff
  - $i \in \text{Ind}(\varphi)$  and, for all  $[i]\psi \in \Gamma_\varphi$ , if  $\mathcal{M}, x \Vdash [i]\psi$ , then  $\mathcal{M}, y \Vdash \psi$   
or
  - $i \notin \text{Ind}(\varphi)$  and, for all  $[\cup]\psi \in \Gamma_\varphi$ , if  $\mathcal{M}, x \Vdash [\cup]\psi$ , then  $\mathcal{M}, y \Vdash \psi$ ;
3.  $(|x|, |y|) \in \mathcal{R}'_\cup$  iff, for all  $[\cup]\psi \in \Gamma_\varphi$ , if  $\mathcal{M}, x \Vdash [\cup]\psi$ , then  $\mathcal{M}, y \Vdash \psi$ ;
4.  $(|x|, |y|) \in \mathcal{R}'_\cap$  iff, for all  $[\cap]\psi \in \Gamma_\varphi$ , if  $\mathcal{M}, x \Vdash [\cap]\psi$ , then  $\mathcal{M}, y \Vdash \psi$ ;
5. if  $p \in \Gamma_\varphi$ , then  $V'(p) = \{ |x| \mid x \in V(p) \}$ .

Notice that  $\mathcal{M}'$  is a filtration of  $\mathcal{M}$  in the usual sense. Hence,  $\mathcal{M}, x \Vdash \psi$  iff  $\mathcal{M}', |x| \Vdash \psi$  holds for all  $\psi \in \Gamma_\varphi$  and all  $x \in W$ . Moreover,  $\mathcal{R}'_\cup = \bigcup_{i \in I} \mathcal{R}'_i$  and  $\mathcal{R}'_\cap \subseteq \bigcap_{i \in I} \mathcal{R}'_i$ ; therefore,  $\mathcal{M}'$  is a finite pseudo-model. Now, using the construction of lemma 2.6, we can obtain a finite model for  $\varphi$ . Thus,  $\mathbf{K}_{\cup\cap}$  has the finite model property and, being finitely axiomatizable, is, therefore, decidable.

## 5 Tableau-based decision procedure for $\mathbf{K}_{\cup\cap}$

We now turn to presenting tableau-based decision procedures for testing  $\mathcal{L}_{\cup\cap}$ -formulas for  $\mathbf{K}_{\cup\cap}$ - and  $\mathbf{DK}_{\cup\cap}$ -satisfiability. Our procedures build upon the tableaux for  $\mathbf{K}_n$  from [7], which in turn are based on the tableau algorithm for  $\mathbf{K}$  from [9].

In this section, we present a deterministic tableau-based algorithm for  $\mathbf{K}_{\cup\cap}$ -satisfiability — inspired by tableau algorithms from [7] —, which can be implemented using polynomial space and which is, moreover, meant to be as time-efficient as possible. Thus, our main goal in the present section is to develop an algorithm for testing for  $\mathbf{K}_{\cup\cap}$ -satisfiability that can be of immediate practical use.

We should mention that PSPACE-completeness of  $\mathbf{K}_{\cup\cap}$  can be straightforwardly deduced from a PSPACE result for the logic  $\mathbf{K}_n^{\cup\cap}$  in [10]. Also, tableaux for  $\mathbf{K}_{\cup\cap}$  can be obtained from resolution-base decision procedure given in [11]. The reason for our introducing the tableau system for  $\mathbf{K}_{\cup\cap}$  is two-fold. First, as already mentioned, we incorporate some “efficiency

tricks” into our procedure, which while not improving the theoretical worst-time complexity, make the testing for satisfiability in  $\mathbf{K}_{\cup\cap}$  practically more efficient. Secondly, the tableaux for  $\mathbf{K}_{\cup\cap}$  lay the groundwork for tableaux for  $\mathbf{DK}_{\cup\cap}$ , for which the corresponding complexity result can not be deduced from known results. It should also be noted that the tableau-style procedure from [8] can be used, after a straightforward translation, to test  $\mathbf{K}_{\cup\cap}$ -, but not  $\mathbf{DK}_{\cup\cap}$ -formulas, for satisfiability; however, the procedure from [8] runs in exponential time and is, thus, suboptimal.

In what follows, we only describe the conceptual core of our tableau algorithm, leaving implementational issues aside; the polynomial-space implementation of our tableaux is identical to the way  $\mathbf{K}$ -tableaux are implemented in [9, pp.476–477].

It is well known that the property that makes it possible to design polynomial-space algorithms for quite a number of modal logics lacking polynomial-size model property (such as  $\mathbf{K}$ ) is that satisfiable formulas of these logics have models that look like trees whose branches have length bounded by the modal degree, and hence the length, of the formula. Exploiting this property, the algorithms try, in broad terms, to check for existence of such a model for the input formula in a depth-first manner, thus only storing information about the branch of the tree-like model that is currently being explored. What we mean by saying that our tableau-based procedure is meant to be as time-efficient as possible is that we try to cut down on the number of branches being explored by the procedure; how this is accomplished will be clear from the presentation below.

## 5.1 Hintikka structures for $\mathbf{K}_{\cup\cap}$

To establish  $\mathbf{K}_{\cup\cap}$ -satisfiability or otherwise of an  $\mathcal{L}_{\cup\cap}^I$ -formula  $\theta$ , our tableaux will not be directly building a model satisfying  $\theta$ , but a different — albeit, as we shall see, closely related — kind of semantic structure, which we refer to as *Hintikka structures*. There are two major differences between Hintikka structures and (pseudo-)models. First, while the latter specify the truth value of every  $\mathcal{L}_{\cup\cap}^I$ -formula at every node, the former provide values of only those formulas that are relevant to evaluating a fixed formula  $\theta$ . Consequently, unlike the case of (pseudo-)models, we define  $\mathbf{K}_{\cup\cap}$ -Hintikka structures *for a formula*  $\theta$ . Second, while (pseudo-)models place explicit conditions on relations  $\mathcal{R}_{\cup}$  and  $\mathcal{R}_{\cap}$  (conditions 3 and 4 of definitions 2.5 and 2.1 respectively), Hintikka structures only impose conditions on the sets of formulas in the labels of

the nodes (the label  $L(w)$  of a node  $w$  contains formulas that are considered “true” at  $w$ ); the conditions placed on labels are meant to implicitly correspond to the desirable conditions on  $\mathcal{R}_\cup$  and  $\mathcal{R}_\cap$ . Thus, even though no conditions are explicitly placed on  $\mathcal{R}_\cup$  and  $\mathcal{R}_\cap$  in Hintikka structures, the labelling is carried out in such a way that every Hintikka structure generates, by a construction described in the proof of lemma 5.4, a pseudo-model in such a way that the “truth” of the formulas in the labels is preserved in the resultant pseudo-model. We then can, using the construction of lemma 2.6, obtain a bona-fide model for  $\theta$ . The converse also holds: satisfiability of  $\theta$  in a model implies the existence of a Hintikka structure for  $\theta$ . Thus, from the point of view of the satisfiability of a stand-alone formula, models and Hintikka structures are interchangeable, and our tableau procedure will take advantage of this fact by checking for the existence of a Hintikka structure rather than a model for the input formula  $\theta$ . To define  $\mathsf{K}_{\cup\cap}$ -Hintikka structures, we need an auxiliary notion of  $\mathcal{L}_{\cup\cap}^I$ -structure (which are different from  $\mathcal{L}_{\cup\cap}^I$ -models defined in Definition 2.1 since the latter do impose conditions on  $\mathcal{R}_\cup$  and  $\mathcal{R}_\cap$ ).

**Definition 5.1** *An  $\mathcal{L}_{\cup\cap}^I$ -structure is a tuple  $(W, \{\mathcal{R}_i\}_{i \in I'}, \mathcal{R}_\cup, \mathcal{R}_\cap)$ , where*

- $W \neq \emptyset$ ;
- $I' \subseteq I$ ;
- $\mathcal{R}_\cup, \mathcal{R}_\cap$  and  $\mathcal{R}_i$ , for every  $i \in I'$ , are binary relations on  $W$ .

Thus, in structures, unlike in both models and pseudo-models, no conditions whatsoever are imposed on  $\mathcal{R}_\cup$  and  $\mathcal{R}_\cap$ . We are now ready to define  $\mathsf{K}_{\cup\cap}$ -Hintikka structures for a formula  $\theta$ . By  $\text{Ind}^+(\theta)$  we denote the set  $\text{Ind}(\theta)$  if the latter is non-empty; otherwise,  $\text{Ind}^+(\theta)$  is a singleton containing an arbitrary  $i \in I$ ; this ensures that  $\text{Ind}^+(\theta)$  is always non-empty.

**Definition 5.2** *A  $\mathsf{K}_{\cup\cap}$ -Hintikka structure for a formula  $\theta \in \mathcal{L}_{\cup\cap}^I$  is a tuple  $\mathcal{H} = (W, \{\mathcal{R}_i\}_{i \in \text{Ind}^+(\theta)}, \mathcal{R}_\cup, \mathcal{R}_\cap, L)$ , where*

- $(W, \{\mathcal{R}_i\}_{i \in \text{Ind}^+(\theta)}, \mathcal{R}_\cup, \mathcal{R}_\cap)$  is an  $\mathcal{L}_{\cup\cap}^I$ -structure;
- $L$  is a labeling function, associating with every  $w \in W$  a set of formulas  $L(w)$ , satisfying the following conditions:

- (H1) if  $\varphi \in L(w)$ , then  $\neg\varphi \notin L(w)$ ;
  - (H2) if  $\neg\neg\varphi \in L(w)$ , then  $\varphi \in L(w)$ ;
  - (H3) if  $\varphi \vee \psi \in L(w)$ , then  $\varphi \in L(w)$  or  $\psi \in L(w)$ ;
  - (H4) if  $\neg(\varphi \vee \psi) \in L(w)$ , then  $\neg\varphi \in L(w)$  and  $\neg\psi \in L(w)$ ;
  - (H5) if  $\langle \cap \rangle \varphi \in L(w)$ , then there exists  $v \in W$  such that  $(w, v) \in \mathcal{R}_\cap$  and  $\varphi \in L(v)$ ;
  - (H6) if  $\neg\langle \cap \rangle \varphi \in L(w)$  and  $(w, v) \in \mathcal{R}_\cap$ , then  $\neg\varphi \in L(v)$ ;
  - (H7) if  $\langle \cap \rangle \varphi \in L(w)$ , then  $\langle i \rangle \varphi \in L(w)$  for every  $i \in \text{Ind}^+(\theta)$ ;
  - (H8) if  $\langle i \rangle \varphi \in L(w)$  and  $\langle \cap \rangle \varphi \notin L(w)$ , then there exists  $v \in W$  such that  $(w, v) \in \mathcal{R}_i$  and  $\varphi \in L(v)$ ;
  - (H9) if  $\neg\langle i \rangle \varphi \in L(w)$ , then  $\neg\varphi \in L(v)$  for every  $v \in W$  such that  $(w, v) \in \mathcal{R}_i$  or  $(w, v) \in \mathcal{R}_\cap$ ;
  - (H10) if  $\langle i \rangle \varphi \in L(w)$ , then  $\langle \cup \rangle \varphi \in L(w)$ ;
  - (H11) if  $\langle \cup \rangle \varphi \in L(w)$ ,  $\langle \cap \rangle \varphi \notin L(w)$  and  $\langle i \rangle \varphi \notin L(w)$  for every  $i \in \text{Ind}^+(\theta)$ , then there exists  $v \in W$  such that  $(w, v) \in \mathcal{R}_\cup$  and  $\varphi \in L(v)$ ;
  - (H12) if  $\neg\langle \cup \rangle \varphi \in L(w)$  and  $(w, v) \in \mathcal{R}_i$  for some  $i \in \text{Ind}^+(\theta)$  or  $(w, v) \in \mathcal{R}_\cap$  or  $(w, v) \in \mathcal{R}_\cup$ , then  $\neg\varphi \in L(v)$ .
- Lastly,  $\theta \in L(w)$  for some  $w \in W$ .

We next show that, from the point of view of the satisfiability of a stand-alone formula, Hintikka structures are equivalent to models.

**Lemma 5.3** *Let  $\theta \in \mathcal{L}_{\cup\cap}^I$ . If  $\theta$  is satisfiable in an  $\mathcal{L}_{\cup\cap}^I$ -model, then there exists a  $\mathcal{K}_{\cup\cap}$ -Hintikka structure for  $\theta$ .*

**Proof.** Immediate, as every model  $\mathcal{M}$  gives rise to a Hintikka structure in which  $L(w) = \{\varphi \mid \mathcal{M}, w \Vdash \varphi\}$ .  $\square$

**Lemma 5.4** *Let  $\theta \in \mathcal{L}_{\cup\cap}^I$ . If there exists a  $\mathcal{K}_{\cup\cap}$ -Hintikka structure for  $\theta$ , then  $\theta$  is satisfiable in an  $\mathcal{L}_{\cup\cap}^I$ -pseudo-model.*

**Proof.** Let  $\mathcal{H} = (W, \{\mathcal{R}_i\}_{i \in \text{Ind}^+(\theta)}, \mathcal{R}_\cup, \mathcal{R}_\cap, L)$  be a  $\mathbf{K}_{\cup\cap}$ -Hintikka structure for  $\theta$ . We first define a pseudo-model  $\mathcal{M} = (W, \{\mathcal{R}'_i\}_{i \in I}, \mathcal{R}'_\cup, \mathcal{R}'_\cap, V)$  using  $\mathcal{H}$  and then show that  $\mathcal{M}$  satisfies  $\theta$ .

We start by defining the accessibility relations of  $\mathcal{M}$ . First, for every  $i \in I$ , if  $i \in \text{Ind}^+(\theta)$ , then  $\mathcal{R}'_i = \mathcal{R}_i \cup \mathcal{R}_\cap$ ; otherwise  $\mathcal{R}'_i = \mathcal{R}_\cap$ . Second, we define  $\mathcal{R}'_\cup$  and  $\mathcal{R}'_\cap$  as follows:

- $\mathcal{R}'_\cup = \mathcal{R}_\cup \cup \mathcal{R}_\cap \cup \bigcup_{i \in \text{Ind}^+(\theta)} \mathcal{R}_i$ ;
- $\mathcal{R}'_\cap = \mathcal{R}_\cap$ .

To complete the definition of  $\mathcal{M}$ , define  $V : \mathbf{Par} \mapsto 2^W$  as follows:  $V(p) = \{v \in W : p \in L(v)\}$  for every  $p \in \mathbf{Par}$ . The above definition of accessibility relations guarantees that we have the following:

- $\bigcup_{i \in I} \mathcal{R}'_i \subseteq \mathcal{R}'_\cup$ ;
- $\mathcal{R}'_\cap \subseteq \bigcap_{i \in I} \mathcal{R}'_i$ .

Therefore,  $\mathcal{M}$  is a pseudo-model.

It remains to show that  $\mathcal{M}$  satisfies  $\theta$ . To that end, we show by structural induction on  $\theta$  that, for every subformula  $\chi$  of  $\theta$  (including  $\theta$  itself) and every  $w \in W$ , the following hold:

- (i) if  $\chi \in L(w)$ , then  $\mathcal{M}, w \Vdash \chi$ ;
- (ii) if  $\neg\chi \in L(w)$ , then  $\mathcal{M}, w \Vdash \neg\chi$ .

Suppose that  $\chi = p \in \mathbf{Par}$ . If  $p \in L(w)$ , then  $w \in V(p)$  and thus  $\mathcal{M}, w \Vdash p$ . If, on the other hand,  $\neg p \in L(w)$ , then, by (H1),  $p \notin L(w)$ ; hence,  $w \notin V(p)$  and thus  $\mathcal{M}, w \not\Vdash p$ ; therefore,  $\mathcal{M}, w \Vdash \neg p$ .

Suppose that  $\chi = \neg\psi$ . If  $\neg\psi \in L(w)$  then  $\mathcal{M}, w \Vdash \neg\psi$  follows immediately from the inductive hypothesis. If, on the other hand,  $\neg\neg\psi \in L(w)$ , then by (H2),  $\psi \in L(w)$  and, by inductive hypothesis,  $\mathcal{M}, w \Vdash \psi$ ; hence,  $\mathcal{M}, w \Vdash \neg\neg\psi$ .

The case of  $\chi = \varphi \vee \psi$  is straightforward using (H3) and (H4).

Suppose that  $\chi = \langle i \rangle \psi$  (and, thus,  $i \in \text{Ind}^+(\theta)$ ). Assume that  $\langle i \rangle \psi \in L(w)$ . There are two cases to consider. If  $\langle \cap \rangle \psi \in L(w)$ , then, by (H5), there exists  $v \in W$  such that  $(w, v) \in \mathcal{R}_\cap$  and  $\varphi \in L(v)$ . If, on the other hand,  $\langle \cap \rangle \psi \notin L(w)$ , then by (H8), there exists  $v \in W$  such that  $(w, v) \in \mathcal{R}_i$  and

$\varphi \in L(v)$ . As  $\mathcal{R}'_i = \mathcal{R}_i \cup \mathcal{R}_\cap$ , in either case,  $(w, v) \in \mathcal{R}'_i$ , which, together with the inductive hypothesis, gives us the desired  $\mathcal{M}, w \Vdash \langle i \rangle \psi$ .

Assume next that  $\neg \langle i \rangle \psi \in L(w)$ . Then, by (H9),  $\neg \varphi \in L(v)$  holds for every  $v \in W$  such that  $(w, v) \in \mathcal{R}_i$  or  $(w, v) \in \mathcal{R}_\cap$ . Therefore,  $\neg \varphi \in L(v)$  holds for every  $v \in W$  such that  $(w, v) \in \mathcal{R}'_i$ , which, together with the inductive hypothesis, gives us  $\mathcal{M}, w \Vdash \neg \langle i \rangle \psi$ .

Suppose that  $\chi = \langle \cup \rangle \psi$ . Assume that  $\langle \cup \rangle \psi \in L(w)$ . We have to show that  $\mathcal{M}, w \Vdash \langle \cup \rangle \psi$ . There are three cases to consider. If  $\langle i \rangle \psi \in L(w)$  for some  $i \in \text{Ind}^+(\theta)$  and  $\langle \cap \rangle \psi \notin L(w)$ , then, the conclusion follows from (H8), the definition of  $\mathcal{R}'_\cup$ , and the inductive hypothesis. If  $\langle \cap \rangle \psi \in L(w)$ , then the conclusion follows from (H5), the definition of  $\mathcal{R}'_\cup$ , and the inductive hypothesis. Lastly, if neither of the two previous cases apply, we get the desired conclusion from (H11), the definition of  $\mathcal{R}'_\cup$ , and the inductive hypothesis.

Now assume that  $\neg \langle \cup \rangle \psi \in L(w)$ . The desired conclusion then follows from (H12), definition of  $\mathcal{R}'_\cup$ , and inductive hypothesis.

Lastly, suppose that  $\chi = \langle \cap \rangle \psi$ . Assume that  $\langle \cap \rangle \psi \in L(w)$ . Then, (H5), definition of  $\mathcal{R}'_\cap$ , and the inductive hypothesis imply that  $\mathcal{M}, w \Vdash \langle \cap \rangle \psi$ .

Assume next that  $\neg \langle \cap \rangle \psi \in L(w)$ . This case follows from (H6), definition of  $\mathcal{R}'_\cap$ , and the inductive hypothesis.  $\square$

**Theorem 5.5** *Let  $\theta \in \mathcal{L}_{\cup\cap}^I$ . Then,  $\theta$  is satisfiable in an  $\mathcal{L}_{\cup\cap}^I$ -model iff there exists a  $K_{\cup\cap}$ -Hintikka structure for  $\theta$ .*

**Proof.** Immediate from lemmas 5.3, 5.4 and 2.6.  $\square$

In the next section, we will use the following definition.

**Definition 5.6** *Let  $\mathcal{S} = (W, \{\mathcal{R}_j\}_{j \in J})$  be a relational structure. We say that  $\mathcal{S}$  is tree-like if the structure  $(W, \bigcup_{j \in J} \mathcal{R}_j)$  is an irreflexive tree. We say that  $\mathcal{S}$  is forest-like if it is a disjoint union of tree-like structures.*

## 5.2 Tableau procedure

As already mentioned, we only describe the conceptual core of the tableau procedure; the polynomial-space implementation using a stack simulation on the tape of a deterministic Turing machine is the same as in the tableau procedure for logic  $\mathbf{K}$  in [9, pp.476–477].

Conceptually, the tableau procedure for testing a formula  $\theta \in \mathcal{L}_{\cup\cap}$  for satisfiability in an  $\mathcal{L}_{\cup\cap}$ -model is an attempt to check for the existence of a non-empty forest-like  $\mathcal{L}_{\cup\cap}$ -structure  $\mathcal{T}^\theta$ , referred to as a *tableau* for  $\theta$ , representing all possible tree-like Hintikka structures for  $\theta$ . If the attempt succeeds,  $\theta$  is declared satisfiable; otherwise, it is pronounced unsatisfiable.

The whole procedure is made up of three sub-procedures, or phases: *construction*, *prestate elimination*, and *state elimination*. During the construction phase, the construction rules are used to produce a directed tree-like graph  $\mathcal{P}^\theta$  — called the *pretableau* for  $\theta$  — whose set of nodes properly contains the set of nodes of the tableau  $\mathcal{T}^\theta$  we are ultimately building. Nodes of  $\mathcal{P}^\theta$  are sets of formulas, some of which, called *states*, are meant to represent states of a Hintikka structure, while others, *prestates*, are deployed for implementational convenience. During the prestate elimination phase, we create a non-empty forest-like  $\mathcal{L}_{\cup\cap}$ -structure  $\mathcal{T}_0^\theta$  out of  $\mathcal{P}^\theta$ , called the *initial tableau for  $\theta$* , by eliminating all the prestates of  $\mathcal{P}^\theta$  (and tweaking its edges). Lastly, during the state elimination phase, we interactively remove from  $\mathcal{T}_0^\theta$  all the states, if any, that cannot be satisfied in any Hintikka structure, for one of the following two reasons: either they are inconsistent or they do not have all the successors needed for their satisfaction. The result is a (possibly empty) forest-like  $\mathcal{L}_{\cup\cap}$ -structure  $\mathcal{T}^\theta$ , called the *final tableau for  $\theta$* . Then, if we have some state  $\Delta$  in  $\mathcal{T}^\theta$  containing  $\theta$ , we declare  $\theta$  satisfiable; otherwise, we declare it unsatisfiable.

Now, we describe the three phases in more detail. As has already been mentioned, during the construction phase, we build a tree-like graph, called a *pretableau*, whose set of nodes properly contains the set of nodes of the tableau we are building. The nodes of the pretableau come in two varieties — *prestates* and *states*. States are meant to correspond to the states of the Hintikka structures the procedure is trying to build, while prestates are “embryo states”, which in the course of the construction are unwound into states. Technically, states are downward saturated sets, defined as follows.

**Definition 5.7** *Let  $\Delta$  be a set of  $\mathcal{L}_{\cup\cap}^I$ -formulas. We say that  $\Delta$  is downward saturated if it satisfies the following conditions:*

1. *if  $\neg\neg\varphi \in \Delta$ , then  $\varphi \in \Delta$ ;*
2. *if  $\varphi \vee \psi \in \Delta$ , then  $\varphi \in \Delta$  or  $\psi \in \Delta$ ;*
3. *if  $\neg(\varphi \vee \psi) \in \Delta$ , then  $\neg\varphi \in \Delta$  and  $\neg\psi \in \Delta$ ;*

4. if  $\langle \cap \rangle \varphi \in \Delta$ , then  $\langle i \rangle \varphi \in \Delta$  for every  $i \in \text{Ind}^+(\theta)$ ;
5.  $\langle i \rangle \varphi \in \Delta$  for some  $i \in \text{Ind}^+(\theta)$ , then  $\langle \cup \rangle \varphi \in \Delta$ .

Prestates, on the other hand, are not required to be downward saturated.

As well as containing two types of node, the pretableau contains two kinds of edge. One type of edge, unmarked one, represents the “exhaustive search” for a Hintikka structure for the input formula. The exhaustive search is set in motion when a prestate  $\Gamma$  contains a disjunction, say  $\psi_1 \vee \psi_2$ , — then, we create at least two states out of  $\Gamma$ , one of which contains  $\psi_1$ , the other  $\psi_2$ . Thus, if prestate  $\Gamma$  is connected by unmarked edges to states  $\Delta_1$  and  $\Delta_2$ , denoted  $\Gamma \longrightarrow \Delta_1$  and  $\Gamma \longrightarrow \Delta_2$ , this intuitively means that every node of a Hintikka structure whose label contains (as a subset)  $\Gamma$ , has to contain (as a subset) either  $\Delta_1$  or  $\Delta_2$ . The other type of edge, which is marked by some  $\pi \in \text{Ind}^+(\theta) \cup \{\cup, \cap\}$  represents a transition edge in the Hintikka structure the overall procedure is trying to build. Thus, if state  $\Delta$  is connected by a  $\pi$ -arrow to prestate  $\Gamma$ , denoted  $\Delta \xrightarrow{\pi} \Gamma$ , this means that in every Hintikka structure a node whose label contains  $\Delta$  is related by relation  $\mathcal{R}_\pi$  to a node whose label contains  $\Gamma$ . We note that in the pretableau unmarked arrows, representing exhaustive search, always connect prestates to states (as each prestate — in general, bar the case that it does not contain any disjunctions — can be unwound into different states), while marked arrows, representing transitions in Hintikka structures, always connect states to prestates (thus, we never create in one go states from states).

The procedure for formula  $\theta$  starts off with creating a prestate  $\{\theta\}$  and then alternatingly creating states from prestates by downward saturation — formalized in the rule **(PR)** below — and prestates from states according to the rule **(SR)** stated below. As in [7], a prestate is created from a state  $\Delta$  when  $\Delta$  contains a diamond formula that requires the existence of a reachable state satisfying a formula under the diamond. The main difference between our procedure and the one from [7] is that we divide all the diamond formulas into “active” and “dormant”, and only create prestates from a state  $\Delta$  if its existence is required by an active diamond in  $\Delta$ , which allows us to cut down on the number of branches our algorithm has to explore, which makes it more time-efficient. Intuitively, a diamond formula is dormant if it has been added to a state in the course of downward saturation. Thus, if  $\langle \cap \rangle \varphi \in \Delta$ , we add to  $\Delta$ , in the course of its downward saturation, a formula  $\langle i \rangle \varphi$ ; then,  $\langle i \rangle \varphi$  is a dormant diamond formula at  $\Delta$ . On the other hand, as the definition of a downward saturated state reveals, formulas of the form  $\langle \cap \rangle \varphi$  are never

added to states in the course of downward saturation; hence, formulas of the form  $\langle \cap \rangle \varphi$  are always active. Therefore, in our tableaux, the said state  $\Delta$  will have an  $\cap$ -successor containing  $\varphi$ , but not an  $i$ -successor containing  $\varphi$ , which minimizes the number of successors.

We now formally describe the *construction stage*. It is given by two rules, the specification in which order the rules should be applied, and the initial state of the procedure. We call a set of formulas containing, for some formula  $\varphi$ , both  $\varphi$  and  $\neg\varphi$  *patently inconsistent*. The first rule applies to prestates (moreover, the instance of the rule for a given prestate is unique):

**(PR)** Given a prestate  $\Gamma$  that is not patently inconsistent, create all the downward saturated extensions of  $\Gamma$  as *states* and connect  $\Gamma$  to each newly created state by an unmarked arrow.

The second rule applies to states; several of its instances can be applicable in a given state. Before formulating the rule, we formally define the notions of active and dormant diamond formula.

**Definition 5.8** *Let  $\Delta$  be a state of a pretableau.*

- if  $\langle \cap \rangle \varphi \in \Delta$ , then  $\langle \cap \rangle \varphi$  is active at  $\Delta$ ;
- if  $\langle i \rangle \varphi \in \Delta$ , then  $\langle i \rangle \varphi$  is active at  $\Delta$  iff  $\langle \cap \rangle \varphi \notin \Delta$ ;
- if  $\langle \cup \rangle \varphi \in \Delta$ , then  $\langle \cup \rangle \varphi$  is active at  $\Delta$  iff both  $\langle \cap \rangle \varphi \notin \Delta$  and  $\langle i \rangle \varphi \notin \Delta$  for any  $i \in \text{Ind}^+(\varphi)$ .

We can now formulate our second construction rule.

**(SR)** Given a state  $\Delta$  that is not patently inconsistent and a formula  $\langle \pi \rangle \varphi$  that is active at  $\Delta$ , where  $\pi \in \text{Ind}^+(\theta) \cup \{\cap, \cup\}$ , create a prestate  $\Gamma$  as follows:

- if  $\pi = \cup$ , then  $\Gamma = \{\varphi\} \cup \{\neg\psi : \neg\langle \cup \rangle \psi \in \Delta\}$ ;
- if  $\pi = i \in \text{Ind}^+(\theta)$ , then  $\Gamma = \{\varphi\} \cup \{\neg\psi : \neg\langle i \rangle \psi \in \Delta\} \cup \{\neg\psi : \neg\langle \cup \rangle \psi \in \Delta\}$ ;
- if  $\pi = \cap$ , then  $\Gamma = \{\varphi\} \cup \{\neg\psi : \neg\langle \cap \rangle \psi \in \Delta\} \cup \{\neg\psi : \neg\langle i \rangle \psi \in \Delta \text{ for some } i \in \text{Ind}^+(\theta)\} \cup \{\neg\psi : \neg\langle \cup \rangle \psi \in \Delta\}$ .

and connect  $\Delta$  to  $\Gamma$  by an edge marked by  $\pi$ .

As there might be several active diamonds in  $\Delta$ , we might have to create several successor prestates from  $\Delta$ ; the order of their creation is immaterial.

As to the order of applying the above rules, the construction stage consists of alternatingly applying **(PR)** and **(SR)** to pending (pre)states, starting with a prestate  $\{\theta\}$ , where  $\theta$  is the input formula.

As every application of **(SR)** strips off at least one diamond of a formula with respect to which it is being applied and **(PR)** never increases a modal depth of any formula, the result of the construction phase is a tree-like graph  $\mathcal{P}^\theta$  with root  $\{\theta\}$  whose depth is at most  $2|\theta|+1$ , where  $|\theta|$  denotes the length of  $\theta$ .

During the *prestate elimination phase*, we remove all the prestates from  $\mathcal{P}^\theta$  by applying the following rule (**states**( $\Gamma$ ) stands for the set of states created from a prestate  $\Gamma$  by **(PR)**).

**(PE)** For each prestate  $\Gamma$  of  $\mathcal{P}^\theta$ , remove  $\Gamma$  from  $\mathcal{P}^\theta$  and, if  $\Gamma$  is not the root of  $\mathcal{P}^\theta$ , connect the  $\Delta$  with  $\Delta \xrightarrow{\pi} \Gamma$  to every  $\Delta' \in \mathbf{states}(\Gamma)$  by the arrow marked with  $\pi$ .

Note that since  $\mathcal{P}^\theta$  is tree-like, the  $\Delta$  referred to in **(PE)** is unique. We call the structure obtained by applying **(PE)** to  $\mathcal{P}^\theta$  *the initial tableau* and denote it by  $\mathcal{T}_0^\theta$ . Note that as the root of  $\mathcal{P}^\theta$  is a pre-state, it gets removed at this stage; therefore,  $\mathcal{T}_0^\theta$  is a forest composed of trees whose roots contain the input formula  $\theta$ . If, in the course of the construction phase, a state  $\Delta$  has been connected by a  $\pi$ -arrow to a prestate  $\Gamma$ , which has in turn been unwound into at least two states, say  $\Delta_1$  and  $\Delta_2$ , (so that we have both  $\Gamma \longrightarrow \Delta_1$  and  $\Gamma \longrightarrow \Delta_2$ ), then after removing  $\Gamma$ , we want to remember that  $\Delta_1$  and  $\Delta_2$  are not  $\pi$ -successors of  $\Delta$  in the same Hintikka structure, but rather represent two alternative ways to build a Hintikka structure involving  $\Delta$ . Therefore, we mark  $\pi$ -arrows now connecting  $\Delta$  to both  $\Delta_1$  and  $\Delta_2$  by the same “color”. This information will come in handy when we build a Hintikka structure for a satisfiable formula out of a tableau.

During the *state-elimination phase*, we inspect the tree components of  $\mathcal{T}_0^\theta$  from the bottom up and eliminate states that are not satisfiable in any Hintikka structure. Formally, the state elimination phase is composed of stages, each stage consisting of applying one of the two state elimination rules, **(SE1)** and **(SE2)**, stated below. We denote the tableau obtained at stage  $n$  of the elimination process by  $\mathcal{T}_n^\theta$  and its set of nodes by  $S_n$ .

(SE1) If  $\{\varphi, \neg\varphi\} \subseteq \Delta \in S_n$ , for some formula  $\varphi$ , create  $\mathcal{T}_{n+1}^\theta$  by removing  $\Delta$  from  $\mathcal{T}_n^\theta$ .

(SE2) If  $\langle \pi \rangle \varphi \in \Delta \in S_n$  is active at  $\Delta$  and there is no  $\Delta' \in S_n$  such that  $\Delta \xrightarrow{\pi} \Delta'$ , then create  $\mathcal{T}_{n+1}^\theta$  by removing  $\Delta$  from  $\mathcal{T}_n^\theta$ .

The application of the two rules above results in a possibly empty forest  $\mathcal{T}^\theta$ , which we call the *final tableau for  $\theta$* . If the forest is non-empty, its every tree represents (possibly, more than one) Hintikka structure satisfying  $\theta$  at its root. Formally, the outcome of the tableau procedure is determined as follows.

**Definition 5.9** *Let  $\mathcal{T}^\theta$  be a final tableau for  $\theta$  and  $S$  be its set of nodes. If there exists  $\Delta \in S$  such that  $\theta \in \Delta$ , then  $\mathcal{T}^\theta$  is open; otherwise,  $\mathcal{T}^\theta$  is closed.*

The tableau procedure returns “no”, if  $\mathcal{T}^\theta$  is closed; otherwise, it returns “yes” and, moreover, contains all the information needed to build a Hintikka structure (and, hence, a model) for  $\theta$ .

### 5.3 Soundness and completeness

In the present section, we show that the tableau procedure described above is correct; in logical terms, we show that the procedure is sound and complete. We briefly remind the reader what soundness and completeness mean in the context of tableau procedures.

- A tableau procedure is said to be *sound* with respect to the class of models  $\mathbf{M}$  if satisfiability of a formula  $\theta$  in  $\mathbf{M}$  implies that the tableau for  $\theta$  is open.
- A tableau procedure is said to be *complete* with respect to the class of models  $\mathbf{M}$  if the following holds: if the tableau for a formula  $\theta$  is open, then  $\theta$  is satisfiable in  $\mathbf{M}$ .

We start with soundness. First, we state two auxiliary lemmas whose proofs are straightforward.

**Lemma 5.10** *Let  $\Gamma$  be a prestate of  $\mathcal{P}^\theta$  such that  $\mathcal{M}, w \Vdash \Gamma$  for some model  $\mathcal{M}$  and node  $w \in \mathcal{M}$ . Then,  $\mathcal{M}, w \Vdash \Delta$  for at least one  $\Delta \in \mathbf{states}(\Gamma)$ .*

**Lemma 5.11** *Let  $\langle \pi \rangle \varphi \in \Delta \in S_n$ , let  $\langle \pi \rangle \varphi$  be active at  $\Delta$ , and let, finally,  $\mathcal{M}, w \Vdash \Delta$  for some model  $\mathcal{M}$  and node  $w \in \mathcal{M}$ . Then, there exists  $v$  such that  $(w, v) \in \mathcal{R}_\pi$  and  $\mathcal{M}, v \Vdash \Delta'$  holds for at least one  $\Delta'$  with  $\Delta \xrightarrow{\pi} \Delta'$ .*

**Theorem 5.12 (Soundness)** *If  $\theta$  is satisfiable in an  $\mathcal{L}_{\cup\cap}$ -model, then  $\mathcal{T}^\theta$  is open.*

**Proof.** First, an easy induction on the stage of the state elimination procedure shows that we never remove a state  $\Delta \in S_0$  if it is satisfiable in a  $\mathcal{L}_{\cup\cap}$ -model.

Indeed, suppose we are at stage 0. No eliminations have been done thus far, and hence all the satisfiable states are still in place. Now, inductively assume that if  $\Delta'$  is satisfiable, then it has not been removed during the previous  $n$  stages of the state elimination procedure, and thus  $\Delta' \in S_n$ . Consider stage  $n+1$  and a satisfiable state  $\Delta$ . First, it is clear that, since no satisfiable set of formulas can be patently inconsistent,  $\Delta$  cannot be removed from  $\mathcal{T}_n^\theta$  due to **(SE1)**. Second, it follows from lemma 5.11 together with the inductive hypothesis that  $\Delta$  cannot be removed from  $\mathcal{T}_n^\theta$  due to **(SE2)**.

Now, suppose that  $\theta$  is satisfiable. Then, due to lemma 5.10, at least one state in  $\mathbf{states}(\{\theta\})$ , say  $\Delta$ , is satisfiable. Since, as has been argued in the preceding paragraph, satisfiable states never get removed during the state elimination phase,  $\Delta$  is in  $\mathcal{T}^\theta$ . Since  $\theta \in \Delta$ , the tableau  $\mathcal{T}^\theta$  is open.  $\square$

Next, we establish completeness.

**Theorem 5.13 (Completeness)** *If  $\mathcal{T}^\theta$  is open, then  $\theta$  is satisfiable in an  $\mathcal{L}_{\cup\cap}$ -model.*

**Proof.** Suppose that  $\mathcal{T}^\theta$  is open, i.e. there exists  $\Delta \in S$  such that  $\theta \in \Delta$ . We show how to build a Hintikka structure for  $\theta$  out of  $\mathcal{T}^\theta$ . The claim of the theorem then follows from theorem 5.5.

Consider the tree  $\mathcal{T}$  generated by  $\Delta$  (denote the set of nodes of  $\mathcal{T}$  by  $T$ ). To obtain a Hintikka structure out of  $\mathcal{T}$ , first, starting from the root and moving downwards, do the following: whenever a state  $\Delta' \in T$  is connected by arrows marked by the same label  $\pi$  and colored by the same color to more than one state (i.e.,  $\Delta'$  might have several satisfiable  $\pi$ -successors, each of which would be part of a different Hintikka structure), arbitrarily choose one

of those states as an  $\pi$ -successor of  $\Delta'$  and discard all the subtrees of  $\mathcal{T}$  generated by all the other  $\pi$ -successors of the same color. Denote the resulting structure by  $\mathcal{T}'$  and its set of nodes by  $T'$ . Now, define the labelling function on  $T'$  as follows: for each  $\Delta' \in T'$ , put  $L(\Delta') = \Delta'$ . It is straightforward to check that  $(\mathcal{T}', L)$  is a Hintikka structure for  $\theta$ .  $\square$

## 6 Tableau-based decision procedure for $\mathbf{DK}_{\cup\cap}$

In this section, we indicate how the above tableau procedure for  $\mathbf{K}_{\cup\cap}$  should be amended to obtain a tableau procedure for  $\mathbf{DK}_{\cup\cap}$ .

### 6.1 Hintikka structures for $\mathbf{DK}_{\cup\cap}$

We start with defining Hintikka structures for  $\mathbf{DK}_{\cup\cap}$ . Given a finite set of formulas  $\Delta$ , we use the following notation:

- $\Delta_x^\wedge = \bigwedge_{\langle\pi\rangle\varphi \in \Delta} \varphi$ , where  $\pi$  ranges over  $\text{Ind}^+(\theta) \cup \{\cup, \cap\}$ ;
- $\Delta_{\langle i \rangle}^\wedge = \bigwedge_{\langle i \rangle\varphi \in \Delta} \varphi$ , where  $i \in \text{Ind}^+(\theta)$ .

That is,  $\Delta_x^\wedge$  is a conjunction of all the formulas  $\varphi$  such that a formula  $\langle\pi\rangle\varphi$ , where  $\pi$  is *any* modality, appears in  $\Delta$ . Analogously,  $\Delta_{\langle i \rangle}^\wedge$  is a conjunction of all the formulas  $\varphi$  such that a formula  $\langle i \rangle\varphi$ , where  $i$  is a specific modal index from  $\text{Ind}^+(\theta)$ , appears in  $\Delta$ .

**Definition 6.1** A  $\mathbf{DK}_{\cup\cap}$ -Hintikka structure for a formula  $\theta \in \mathcal{L}_{\cup\cap}^I$  is a tuple  $\mathcal{H} = (W, \{\mathcal{R}_i\}_{i \in \text{Ind}^+(\theta)}, \mathcal{R}_\cup, \mathcal{R}_\cap, L)$ , where

- $(W, \{\mathcal{R}_i\}_{i \in I}, \mathcal{R}_\cup, \mathcal{R}_\cap)$  is an  $\mathcal{L}_{\cup\cap}^I$ -structure satisfying the following conditions:
  - (R1) Relations  $\mathcal{R}_\cap$  and  $\mathcal{R}_i$ , for every  $i \in \text{Ind}^+(\theta)$ , are deterministic;
  - (R2) for every  $i \in \text{Ind}^+(\theta)$ , the domains of  $\mathcal{R}_i$  and  $\mathcal{R}_\cap$  are disjoint;
  - (R3) the domains of  $\mathcal{R}_\cup$  and  $\mathcal{R}_\cap$  are disjoint.
- $L$  is a labeling function, associating with every  $w \in W$  a set of formulas  $L(w)$ , satisfying conditions (H1)–(H4), (H6), (H9), (H10), and (H12) of definition 5.2, as well as the following conditions:

- (DH1) if  $\langle \cap \rangle \chi \in L(w)$  for some  $\chi$ , then  $\langle \cap \rangle L(w)_{\chi}^{\wedge} \in L(w)$ ;
  - (DH2) if  $\langle \cap \rangle \varphi \in L(w)$  and  $\langle \cap \rangle \psi \in L(w)$  does not hold for any conjunct  $\psi$  of  $\varphi$ , then there exists  $v \in W$  such that  $(w, v) \in \mathcal{R}_{\cap}$  and  $\varphi \in L(v)$ ;
  - (DH3) if  $\langle i \rangle \chi \in L(w)$  for some  $\chi$  and  $\langle \cap \rangle \psi \in L(w)$  for no  $\psi$ , then  $\langle i \rangle L(w)_{\langle i \rangle}^{\wedge} \in L(w)$ ;
  - (DH4) if  $\langle i \rangle \varphi \in L(w)$  and neither  $\langle \cap \rangle \chi \in L(w)$  nor  $\langle i \rangle \psi \in L(w)$  holds for any formula  $\chi$  and any conjunct  $\psi$  of  $\varphi$ , then there exists  $v \in W$  such that  $(w, v) \in \mathcal{R}_i$  and  $\varphi \in L(v)$ ;
  - (DH5) if  $\langle \cap \rangle \chi \in L(w)$  for some  $\chi$  and  $\langle \cup \rangle \varphi \in L(w)$ , then  $\langle \cap \rangle \varphi \in L(w)$ ;
  - (DH6) if  $\langle \cap \rangle \varphi \in L(w)$ , then  $\langle \cup \rangle \varphi \in L(w)$ ;
  - (DH7) if  $\langle i \rangle \varphi \in L(w)$ , for some  $i \in \text{Ind}^+(\theta)$ , then  $\langle \cup \rangle \varphi \in L(w)$ ;
  - (DH8) if  $\langle \cup \rangle \varphi \in L(w)$  and  $\langle \cap \rangle \chi \in L(w)$  does not hold for any  $\chi$ , then there exists  $v \in W$  such that  $(w, v) \in \mathcal{R}_{\cup}$  and  $\varphi \in L(v)$ .
- Lastly,  $\theta \in L(w)$  for some  $w \in W$ .

It is straightforward to check that every deterministic model gives rise to a  $\text{DK}_{\cup}$ -Hintikka structure. For the other direction, in view of lemma 2.9, it suffices to show that every  $\text{DK}_{\cup}$ -Hintikka structure can be turned into a deterministic pseudo-model.

**Lemma 6.2** *Let  $\theta \in \mathcal{L}_{\cup \cap}^I$ . If there exists a  $\text{DK}_{\cup \cap}$ -Hintikka structure for  $\theta$ , then  $\theta$  is satisfiable in a deterministic  $\mathcal{L}_{\cup \cap}^I$ -pseudo-model.*

**Proof.** Define a pseudo-model  $\mathcal{M}$  out of  $\mathcal{H}$  exactly as in the proof of lemma 5.4. The properties of accessibility relations mentioned in definition 6.1 guarantee that  $\mathcal{M}$  is deterministic. The proof that the “truth” of formulas in the labels of nodes of  $\mathcal{H}$  is preserved in  $\mathcal{M}$  is a straightforward modification of the inductive argument in the proof of lemma 5.4. This is enough to prove the claim of the lemma.  $\square$

Thus, we have the following theorem.

**Theorem 6.3** *Let  $\theta \in \mathcal{L}_{\cup \cap}^I$ . Then,  $\theta$  is satisfiable in a deterministic  $\mathcal{L}_{\cup \cap}^I$ -model iff there exists a  $\text{DK}_{\cup \cap}$ -Hintikka structure for  $\theta$ .*

## 6.2 Tableau procedure

We only indicate the changes that need to be made to the tableau procedure for  $K_{\cup\cap}$  described in section 5.2.

As before, the nodes of a pretableau are divided into prestates and states, states being downward saturated sets of formulas; the conditions on such sets, however, are now different.

**Definition 6.4** *Let  $\Delta$  be a set of  $\mathcal{L}_{\cup\cap}^I$ -formulas. We say that  $\Delta$  is deterministically downward saturated if it satisfies conditions 1–3 of definition 5.7 as well as the following conditions:*

1. *if  $\langle\cap\rangle\chi \in L(w)$  for some  $\chi$ , then  $\langle\cap\rangle L(w)_{\chi}^{\wedge} \in L(w)$ ;*
2. *if  $\langle i\rangle\chi \in L(w)$  for some  $\chi$  and  $\langle\cap\rangle\psi \in L(w)$  for no  $\psi$ , then  $\langle i\rangle L(w)_{\langle i\rangle}^{\wedge} \in L(w)$ ;*
3. *if  $\langle\cap\rangle\chi \in L(w)$  for some  $\chi$  and  $\langle\cup\rangle\varphi \in L(w)$ , then  $\langle\cap\rangle\varphi \in L(w)$ ;*
4. *if  $\langle\cap\rangle\varphi \in L(w)$ , then  $\langle\cup\rangle\varphi \in L(w)$ ;*
5. *if  $\langle i\rangle\varphi \in L(w)$ , for some  $i \in \text{Ind}^+(\theta)$ , then  $\langle\cup\rangle\varphi \in L(w)$ .*

Accordingly, the new state-creation rules are as follows:

**(DPR)** Given a prestate  $\Gamma$  that is not patently inconsistent, create all the deterministically downward saturated extensions of  $\Gamma$  as *states* and connect  $\Gamma$  to each newly created state by an unmarked arrow.

Analogously, to state the new prestate creation rule, we need a new definition of active and dormant diamond formulas.

**Definition 6.5** *Let  $\Delta$  be a state of a pretableau.*

- *if  $\langle\cap\rangle\varphi \in \Delta$  and  $\langle\cap\rangle\psi \in \Delta$  does not hold for any conjunct  $\psi$  of  $\varphi$ , then  $\langle\cap\rangle\varphi \in \Delta$  is active at  $\Delta$ ;*
- *if  $\langle i\rangle\varphi \in \Delta$ , and neither  $\langle\cap\rangle\chi \in \Delta$  holds for any formula  $\chi$  nor  $\langle i\rangle\psi \in \Delta$  holds for any conjunct  $\psi$  of  $\varphi$ , then  $\langle i\rangle\varphi$  is active at  $\Delta$ ;*
- *if  $\langle\cup\rangle\varphi \in \Delta$  and  $\langle\cap\rangle\chi \in \Delta$  does not hold for any  $\chi$ , then  $\langle\cup\rangle\varphi$  is active at  $\Delta$ .*

Given this new definition of active diamonds, the wording of our prestate creation rule does not need any change.

**(SR)** Given a state  $\Delta$  that is not patently inconsistent and a formula  $\langle \pi \rangle \varphi$  that is active at  $\Delta$ , where  $\pi \in \text{Ind}^+(\theta) \cup \{\cap, \cup\}$ , create a prestate  $\Gamma$  as follows:

- if  $\pi = \cup$ , then  $\Gamma = \{\varphi\} \cup \{\neg\psi : \neg\langle \cup \rangle \psi \in \Delta\}$ ;
- if  $\pi = i \in \text{Ind}^+(\theta)$ , then  $\Gamma = \{\varphi\} \cup \{\neg\psi : \neg\langle i \rangle \psi \in \Delta\} \cup \{\neg\psi : \neg\langle \cup \rangle \psi \in \Delta\}$ ;
- if  $\pi = \cap$ , then  $\Gamma = \{\varphi\} \cup \{\neg\psi : \neg\langle \cap \rangle \psi \in \Delta\} \cup \{\neg\psi : \neg\langle i \rangle \psi \in \Delta \text{ for some } i \in \text{Ind}^+(\theta)\} \cup \{\neg\psi : \neg\langle \cup \rangle \psi \in \Delta\}$ .

and connect  $\Delta$  to  $\Gamma$  by an edge marked by  $\pi$ .

The major difference from the analogous rule in the  $\mathbf{K}_{\cup\cap}$ -tableau procedure is that now only one formula of the form  $\langle \cap \rangle \varphi$  can be active at a give  $\Delta$  and, likewise, for every  $i \in \text{Ind}^+(\theta)$ , only one formula of the form  $\langle i \rangle \varphi$  can be active at a given  $\Delta$  — the latter only if no formula of the form  $\langle \cap \rangle \chi$  belongs to  $\Delta$ .

The rest of the procedure — prestate and state elimination subprocedures — is essentially the same as in the case of  $\mathbf{K}_{\cup\cap}$ -tableaux.

The soundness and completeness arguments are also along the same lines as in the case of  $\mathbf{K}_{\cup\cap}$ -tableaux: to prove soundness, we need to show that no state satisfiable in a deterministic model ever gets removed during the state elimination phase; for completeness, we need to show that every open tableau can be turned into a deterministic Hintikka structure. Both are straightforward adaptations of the corresponding proofs for  $\mathbf{K}_{\cup\cap}$ -tableaux and are, thus, left to the reader. The only issue worth drawing attention to in the completeness proof is that conditions (R1) through (R3) of definition 6.1 are satisfied; these easily follow from definition 6.5 together with the rule **(SR)**.

We can use essentially the same argument as in [9] to show that our  $\mathbf{DK}_{\cup\cap}$ -tableau procedure can be implemented as a deterministic algorithm using space polynomial in the size of the input formula.

## 7 Conclusions

In this paper, we have given a complete and sound axiomatization for modal logics with infinitely many basic modalities and extra modalities corresponding to their infinite union and intersection, both for the general case and for the case when the basic modalities are deterministic. We have also given tableau decision procedures for both logics and proved that their satisfiability problem is in PSPACE. The proof that their satisfiability problem is PSPACE-hard is very simple, seeing that both logics are obviously conservative extensions of modal logic **K**. To be more precise, since the set  $I$  of labels is countably infinite, then a formula  $\varphi$  containing only the modality  $\langle \cup \rangle$  is satisfiable in a (deterministic) model  $(W, \{R_i\}_{i \in I}, R_\cup, R_\cap, V)$  iff  $\varphi$  is satisfied in a model  $(W', R'_\cup, V')$  where  $R'_\cup$  is any binary relation on  $W'$ .

## References

- [1] Natasha Alechina, Philippe Balbiani, and Dmitry Shkatov. Logics with modalities corresponding to infinite unions and intersections of accessibility relations. In *Proceedings of the International Workshop on Hybrid Logic 2007 (HyLo 2007)*, 2007.
- [2] Natasha Alechina, Maarten de Rijke, and Stéphane Demri. A modal perspective on path constraints. *Journal of Logic and Computation*, 13(6):939–956, 2003.
- [3] Natasha Alechina and Dmitry Shkatov. Logics with an existential modality. In Guido Governatori, Ian Hodkinson, and Yde Venema, editors, *Advances in Modal Logic*, volume 6, pages 31–48. College Publications, 2006.
- [4] Philippe Balbiani, Dimiter Vakarelov. PDL with Intersection of Programs: A Complete Axiomatization. *Journal of Applied Non-Classical Logics*, 13(3-4): 231–276, 2003.
- [5] Mordechai Ben-Ari, Joseph Halpern, and Amir Pnueli. Deterministic propositional dynamic logic: finite models, complexity, and completeness. *Journal of Computer and Systems Science*, 25(3):402–417, 1982.

- [6] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*. CUP, 2001.
- [7] Joseph Y. Halpern and Yoram Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54:319–379, 1992.
- [8] Rajeev Gore and Linh Anh Nguyen. A tableau calculus with automaton-labeled formulae for regular grammar logics. In Bernhard Beckert, editor, *Tableaux 2005: Proceedings of on the International Conference on Automatic Reasoning with Analytic Tableaux and Related Methods*, Lecture Notes in Artificial Intelligence, volume 3702, pages 138–152. Springer-Verlag, 2005.
- [9] Richard E. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM Journal of Computation*, 6(3):467–480, 1977.
- [10] Carsten Lutz and Ulrike Sattler. The complexity of reasoning with boolean modal logics. In Frank Wolter, Heinrich Wansing, Maarten de Rijke, and Michael Zakharyashev, editors, *Advances in Modal Logic*, volume 3, pages 329–348. World Scientific, 2002.
- [11] Renate Schmidt and Ulrich Hustadt. Using resolution for testing modal satisfiability and building models. *Journal of Automated Reasoning*, volume 28(2):205–232, 2002.
- [12] Wiebe van der Hoek and John-Jules Ch. Meyer. Making some issues of implicit knowledge explicit. *International Journal of Foundations of Computer Science*, 3(2):193–224, 1992.