Tableau-based decision procedure for the multiagent epistemic logic with all coalitional operators for common and distributed knowledge

MAI AJSPUR*, *Department of Communication, Business and Information Technologies, Roskilde University, Roskilde 4000, Denmark*

VALENTIN GORANKO†, *Department of Informatics and Mathematics, Technical University of Denmark, Kongens Lyngby 2800, Denmark, and Department of Mathematics, University of Johannesburg, Johannesburg 2006, South Africa*

DMITRY SHKATOV‡, *School of Computer Science, University of the Witwatersrand, Johannesburg 2050, South Africa*

Abstract

We develop a conceptually clear, intuitive and feasible decision procedure for testing satisfiability in the full multiagent epistemic logic **CMAEL(CD)** with operators for common and distributed knowledge for all coalitions of agents mentioned in the language. To that end, we introduce Hintikka structures for **CMAEL(CD)** and prove that satisfiability in such structures is equivalent to satisfiability in standard models. Using that result, we design an incremental tableau-building procedure that eventually constructs a satisfying Hintikka structure for every satisfiable input set of formulae of **CMAEL(CD)** and closes for every unsatisfiable input set of formulae.

Keywords: multi-agent epistemic logic, satisfiability, tableau, decision procedure

1 Introduction

Over the last three decades, multiagent epistemic logics [9], [27] have been playing an increasingly important role in computer science and AI. The earliest prominent applications have been to specification, design and verification of distributed protocols [23] and [24]; a number of other applications are described in, among others, [9], [10], and [27]. The most recent, and perhaps more important ones are to specification, design and verification of *multiagent systems*—a research area that has emerged on the borderline between distributed computing, AI, and game theory [36], [45], [47].

1.1 Multiagent epistemic logics and decision methods for them

Languages of multiagent epistemic logics considered in the literature contain various repertoires of epistemic operators. We refer to the basic multiagent epistemic logic, containing only operators of

*E-mail: ajspur@ruc.dk
†E-mail: vfgo@imm.dtu.dk
‡E-mail: dmitry@cs.wits.ac.za

doi:10.1093/jigpal/jzs048 Advance Access published 20 December 2012

individual knowledge for a finite non-empty set Σ of agents, as **MAEL** (**M**ulti **A**gent **E**pistemic **L**ogic). Since all epistemic operators of this logic are **S5**-type modalities, it is also referred to in the literature as $\mathbf{S5}_n$, where n is the number of agents in the language. The logic obtained from **MAEL** by adding the operator of common knowledge among all agents in Σ is then called **MAEL(C)**. This logic, along with **MAEL**, was studied in [25]. Analogously, if **MAEL** is augmented with the operator of distributed knowledge for all agents, then the resulting logic will be called **MAEL(D)**. It was studied in [10] and [38]. **MAEL** augmented with operators of both common and distributed knowledge for the set of all agents, hereafter called **MAEL(CD)**, was studied in [39], and a tableau-based decision procedure for it was first presented in [14]. Thus, all logics mentioned so far either do not have both operators of common and distributed knowledge, or only have those operators for the whole set of agents in the language.

At the same time, there has recently been an increasing interest in the study of *coalitional* multiagent logics (see [30], [31], [32], [2], [40], [13]), i.e. logics whose languages refer to any groups (*coalitions*) of agents. These are important, inter alia, in multiagent systems, where agents may 'cooperate' (i.e. form a coalition) in order to achieve a certain goal. Most of the so far studied logical formalisms referring to coalitions of agents have only been concerned with formalizing reasoning about *strategic abilities* of coalitions. (A notable exception is [40], where the Alternating-time Temporal Epistemic Logic **ATEL** was introduced, whose language contains both common knowledge and strategic operators for coalitions of agents.) Clearly, real cooperation can only be achieved by communication, i.e. exchange of knowledge. Thus, it is particularly natural and important to consider multiagent epistemic logics with operators for both common and distributed knowledge among *any (non-empty) coalitions of agents*. This is the logic under consideration in the present article, hereby called **CMAEL(CD)** (for **C**oalitional **M**ulti-**A**gent **E**pistemic **L**ogic with operators of **C**ommon and **D**istributed knowledge). It subsumes all multiagent epistemic logics mentioned above, except **ATEL**.

In order to be practically useful for such tasks as specification and design of distributed or multiagent systems, the respective logic need to be equipped with algorithms solving (constructively) its satisfiability problem, i.e. testing whether a given input formula φ of that logic is satisfiable and, if so, providing enough information for the construction of a model for φ. Decidability of modal logics, including epistemic logics, is usually proved by establishing a 'small model property', which provides a brute force decision procedure consisting of exhaustive search for a model amongst all those whose size is within the theoretically prescribed bounds. The two most common practically feasible general methods for satisfiability checking of modal logics are based on *automata* [41] and on tableaux (see e.g. [33], [3], [11], [46], [8], [19], [12]).

There are various styles of tableau-based decision procedures; see [11], [19] and [12] for detailed exposition and surveys. An easy to describe but somewhat less efficient and practically unfeasible approach, that we will call *maximal tableau* (also called *top-down* in [8]), consists in trying to build in one step a 'canonical' finite model for any given formula out of *all* maximal consistent subsets of the closure of that formula. This method always works in (at least) exponential time and usually produces a wastefully large model, if any exists. A more flexible and more practically applicable version, adopted in the present paper, is a so called *incremental* (aka, 'bottom-up') tableau building procedure. While in all known cases, the worst-case time complexity for maximal and incremental tableaux are the same, the crucial difference is that maximal tableaux *always* require the amount of resources predicted by the theoretical worst-case time estimate, while incremental tableaux work on average much more efficiently[1].

[1]This claim cannot be made mathematically precise due to the lack of an a priori probability distribution on formulae of a logic. The interested reader may consult [17] for comparison of efficiency of the two types of tableaux in the context of Alternating-time temporal logic **ATL**.

1.2 Related work and comparison

The present work is part of a series of papers ([17], [14], [18], [15]) where we have embarked on the project of developing practically efficient yet intuitive and conceptually clear incremental-tableau-based satisfiability checking procedures for a range of multiagent logics. This article builds on the conference papers [14] and [18] by substantially extending, revising and improving them.

There are three inherent complications affecting the construction of a tableau procedure for the logic **CMAEL(CD)**, arising respectively from the common knowledge (fixpoint-definable operator), the distributed knowledge (with associated epistemic relation being the intersection of the individual knowledge epistemic relations), and the interactions between the knowledge operators over different coalitions of agents.

Several tableau-based methods for satisfiability-checking for modal logics with fixpoint-definable operators have been developed and published over the past 30 years, all going back to the tableau-based decision methods developed for the Propositional Dynamic Logic **PDL** in [34], for the branching-time temporal logics **UB** in [3] and **CTL** in [8, Section 5] and [7]. In terms of handling eventualities arising from the fixed-point operators our tableau method follows more closely on the incremental tableaux for the linear time temporal logic **LTL** in [46] and for **CTL** in [8, Section 7].

A particular complication arising in the tableau for **CMAEL(CD)** stems from the fact that the epistemic operators, being **S5** modalities, are symmetric, and thus the epistemic boxes have global effect on the model, too. This requires a special mechanism for propagating their effect backwards when occurring in states of the tableau. In the present article, we have chosen to implement such mechanism by using *analytic cut rules*, going back to Smullyan [37] and Fitting [11], see also [19] and [28]. More recently, tableaux with analytic cut rules for modal logics with symmetric relations have been developed in [21], [20], [5].

We note that there is a natural tradeoff between conceptual clarity and simplicity of (tableau-based) decision procedures on the one hand, and their technical sophistication and optimality on the other hand. We emphasize that the main objective of developing the tableau procedure presented here is the conceptual clarity, intuitiveness and ease of implementation, rather than practical optimality. While being optimal in terms of worst-case time complexity and incorporating some new and non-trivial optimizing features (such as restricted applications of cut rules) this procedure is amenable to various improvements and further optimizations. Most important known such optimizations are *on-the-fly* techniques for elimination of bad states and *one-pass* tableau methods developed for some related logics in [35], [1] and *cut-free* versions of tableau as in [1] for **MAEL(C)**, [22] for PDL with converse operators, [29] for the description logic SHI and of sequent calculi, in [26] for **MAEL(C)** and in [4] for LTL and CTL. We discuss briefly the possible modifications of our procedure, implementing such optimizing techniques in Section 6.

Here is a summary (in a roughly chronological order) of the more closely related previous work, besides our own, on tableau-based decision procedures for multiagent epistemic logics with common and/or distributed knowledge:

- the maximal tableaux for **MAEL(C)**, presented in [25];
- the semantic construction used in [10, Appendix A1] to prove completeness of an axiomatic system for **MAEL(D)**;
- the proof of decidability of **MAEL(CD)** based on finite model property via filtration in [39];
- the maximal tableau-like decision procedure for **ATL**, presented in [44] and extended to **ATEL** in [43];

- the exponential-time tableau-based procedure developed in [6] for testing satisfiability in the BDI logic, that has some common features with **CMAEL(C)**;
- the optimized cut-free single-pass tableaux for the multi-agent logic of common knowledge **MAEL(C)**, in [1]. on tableaux for multiagent logics using global caching and analytic cuts in [5].

1.3 Structure of the paper

In Section 2, we introduce the syntax and semantics of the logic **CMAEL(CD)**. In Section 3, we introduce Hintikka structures for **CMAEL(CD)** and show that Hintikka structures are equivalent to Kripke models with respect to satisfiability of formulae. Then, in Section 4, we develop the tableau procedures checking for satisfiability of formulae of **CMAEL(CD)**. In Section 5, we prove the correctness of our procedure in Section 6 we estimate its complexity, discuss it efficiency and indicate some possible technical improvements. We end with concluding remarks pointing out some directions for further development.

2 Syntax and semantics

2.1 Syntax of ***CMAEL(CD)***

The language of **CMAEL(CD)** contains a fixed, at most countable, set $\mathtt{AP}$ of *atomic propositions*, typically denoted by $p,q,r,\ldots$; a finite, non-empty set Σ of (names for) *agents*[2], typically denoted by $a,b,\ldots$, while sets of agents, called *coalitions*, will be usually denoted by $A,B,\ldots$; a sufficient repertoire of the Boolean connectives, say $\neg$ ('not') and $\wedge$ ('and'); and, for every non-empty coalition A, the epistemic operators $\mathbf{D}_A$ ('*it is distributed knowledge among A that …*') and $\mathbf{C}_A$ ('*it is common knowledge among A that …*'). The formulae of **CMAEL(CD)** are thus defined by the following BNF expression:

$$\varphi := p \mid \neg\varphi \mid (\varphi_1 \wedge \varphi_2) \mid \mathbf{D}_A\varphi \mid \mathbf{C}_A\varphi,$$

where p ranges over $\mathtt{AP}$ and A ranges over the set $\mathcal{P}^+(\Sigma)$ of non-empty subsets of Σ. The other Boolean connectives can be defined as usual. We denote formulae of **CMAEL(CD)** by $\varphi,\psi,\chi,\ldots$ and omit parentheses in formulae whenever it does not result in ambiguity.

The distributed knowledge operator $\mathbf{D}_A\varphi$ intuitively means that an 'A-superagent', who knows everything that any of the agents in A knows, can obtain φ as a logical consequence of their knowledge. For example, if agent a knows that ψ and agent b knows that $\psi\to\chi$, then $\mathbf{D}_{\{a,b\}}\chi$ is true even though neither a nor b knows χ. The operators of individual knowledge $\mathbf{K}_a\varphi$ ('*the agent a knows that* φ'), for $a\in\Sigma$, can be defined as $\mathbf{D}_{\{a\}}\varphi$, henceforth simply written $\mathbf{D}_a\varphi$. Then, we define $\mathbf{K}_A\varphi := \bigwedge_{a\in A}\mathbf{D}_a\varphi$.

The common knowledge operator $\mathbf{C}_A\varphi$ intuitively means that φ is 'public knowledge' among A, i.e. that every agent in A knows that φ and knows that every agent in A knows that φ, etc. Formulae of the form $\neg\mathbf{C}_A\varphi$ are referred to as *(epistemic) eventualities*, for the reasons given later on.

[2]The notion of agent used in the present article is an abstract one; in the context of distributed systems, for example, agents can be thought of as processes making up the system; in the context of multiagent systems, they can be thought of as independent software components of the system.

2.2 Coalitional multiagent epistemic models

Formulae of **CMAEL(CD)** are interpreted in *coalitional multiagent epistemic models*. In order to define those, we first need to introduce *coalitional multiagent epistemic structures* and *frames*.

DEFINITION 2.1
A *coalitional multiagent epistemic structure* (CMAES) is a tuple

$$\mathfrak{S} = (\Sigma, S, \{\mathcal{R}^D_A\}_{A\in\mathcal{P}^+(\Sigma)}, \{\mathcal{R}^C_A\}_{A\in\mathcal{P}^+(\Sigma)})$$

where

1. Σ is a finite, non-empty set of *agents*;[3]
2. $S \neq \emptyset$ is a set of *states*;
3. for every $A \in \mathcal{P}^+(\Sigma)$, $\mathcal{R}^D_A$ is a binary relation on S;
4. for every $A \in \mathcal{P}^+(\Sigma)$, $\mathcal{R}^C_A$ is the reflexive, transitive closure of $\bigcup_{B\subseteq A}\mathcal{R}^D_B$.

DEFINITION 2.2
A *coalitional multiagent epistemic frame* (CMAEF) is a CMAES

$$\mathfrak{F} = (\Sigma, S, \{\mathcal{R}^D_A\}_{A\in\mathcal{P}^+(\Sigma)}, \{\mathcal{R}^C_A\}_{A\in\mathcal{P}^+(\Sigma)}),$$

where each $\mathcal{R}^D_A$ is an equivalence relation satisfying the following condition:

$$(\dagger) \qquad \mathcal{R}^D_A = \bigcap\nolimits_{a\in A}\mathcal{R}^D_a$$

(Here, and further, we write $\mathcal{R}^D_a$ instead of $\mathcal{R}^D_{\{a\}}$, where $a \in \Sigma$.)

If condition ($\dagger$) above is replaced by the following, weaker one:

$$(\dagger\dagger) \qquad \mathcal{R}^D_A \subseteq \mathcal{R}^D_B \text{ whenever } B \subseteq A,$$

then $\mathfrak{F}$ is a *coalitional multiagent epistemic pseudo-frame* (pseudo-CMAEF).

Note that in every (pseudo-)CMAEF $\mathcal{R}^D_A \subseteq \bigcap_{a\in A}\mathcal{R}^D_a$, and hence $\bigcup_{B\subseteq A}\mathcal{R}^D_B = \bigcup_{a\in A}\mathcal{R}^D_a$. Hence, condition 4 of Definition 2.1 in (pseudo-) CMAEFs is equivalent to requiring that $\mathcal{R}^C_A$ is the transitive closure of $\bigcup_{a\in A}\mathcal{R}^D_a$, for every $A \in \mathcal{P}^+(\Sigma)$. Also, note that each $\mathcal{R}^C_A$ in a (pseudo-)CMAEF is an equivalence relation.

DEFINITION 2.3
A *coalitional multiagent epistemic model* (CMAEM) is a tuple $\mathcal{M} = (\mathfrak{F}, \mathtt{AP}, L)$, where $\mathfrak{F}$ is a CMAEF with a set of states S, $\mathtt{AP}$ is a set of atomic propositions, and $L : S \mapsto \mathcal{P}(\mathtt{AP})$ is a *labelling function*, assigning to every state s the set $L(s)$ of atomic propositions true at s.

If $\mathfrak{F}$ is a pseudo-CMAEF, then $\mathcal{M} = (\mathfrak{F}, \mathtt{AP}, L)$ is a *multiagent coalitional pseudo-model* (pseudo-CMAEM).

The notion of truth, or satisfaction, of a **CMAEL(CD)**-formula at a state of a (pseudo-)CMAEM is defined in the standard Kripke semantics style. In particular:

- $\mathcal{M}, s \Vdash \mathbf{D}_A\varphi$ iff $(s,t) \in \mathcal{R}^D_A$ implies $\mathcal{M}, t \Vdash \varphi$;
- $\mathcal{M}, s \Vdash \mathbf{C}_A\varphi$ iff $(s,t) \in \mathcal{R}^C_A$ implies $\mathcal{M}, t \Vdash \varphi$.

[3]Notice that we use the same symbol, 'Σ', both for the set of *names of agents* in the language and for the set of *agents* in CMAES's. It will always be clear from the context which set we refer to.

Definition 2.4

Given a (pseudo-)CMAEM $\mathcal{M}$, a **CMAEL(CD)**-formula φ is *satisfiable* in $\mathcal{M}$ if $\mathcal{M}, s \Vdash \varphi$ holds for some $s \in \mathcal{M}$; φ is *valid* in $\mathcal{M}$ if $\mathcal{M}, s \Vdash \varphi$ holds for every $s \in \mathcal{M}$.

A formula φ is *satisfiable* if it is satisfiable in some CMAEM; it is *valid*, denoted $\Vdash \varphi$, if it is valid in every CMAEM.

The satisfaction condition for the operator $\mathbf{C}_A$ can be re-stated in terms of reachability. Let $\mathcal{M}$ be a (pseudo-)CMAEM with state space S and let $s, t \in S$. We say that t is *A-reachable from s* if either $s = t$ or, for some $n \geq 1$, there exists a sequence $s = s_0, s_1, \ldots, s_{n-1}, s_n = t$ of elements of S such that, for every $0 \leq i < n$, there exists $a_i \in A$ such that $(s_i, s_{i+1}) \in \mathcal{R}^D_{a_i}$. It is then easy to see that the satisfaction condition for $\mathbf{C}_A$ is equivalent to the following one:

- $\mathcal{M}, s \Vdash \mathbf{C}_A\varphi$ iff $\mathcal{M}, t \Vdash \varphi$ for every t that is A-reachable from s.

The following claim be easily verified.

Proposition 2.5

$\Vdash \mathbf{C}_A\varphi \leftrightarrow (\varphi \wedge \bigwedge_{a\in A} \mathbf{D}_a \mathbf{C}_A \varphi)$.

Remark If $\Sigma = \{a\}$, then $\mathbf{D}_a\varphi \leftrightarrow \mathbf{C}_a\varphi$ is valid for all φ. Thus, the single-agent case is essentially trivialized and, therefore, we assume throughout the remainder of the article that the set Σ of names of agents in the language of **CMAEL(CD)** contains at least 2 agents.

3 Hintikka structures for CMAEL(CD)

We are ultimately interested in (constructive) satisfiability of (finite sets of) formulae in models. However, the tableau procedure we present in this article checks for the existence of a more general kind of semantic structure for the input formula, namely a *Hintikka structure*. In Section 3.1, we introduce Hintikka structures for **CMAEL(CD)**. In Section 3.2, we show that satisfiability in Hintikka structures is equivalent to satisfiability in models; consequently, testing for satisfiability in a Hintikka structure can replace testing for satisfiability in a model.

3.1 Fully expanded sets and Hintikka structures

There are two fundamental differences between (pseudo-)models and Hintikka structures for **CMAEL(CD)**, which make working with the latter substantially easier than working directly with models. First, while models specify the truth value of every formula of the language at each state, Hintikka structures only do so for the formulae relevant to the evaluation of a fixed formula θ (or, a finite set of formulae Θ) at a distinguished state. Second, the relations in (pseudo-) models have to satisfy certain conditions (see Definition 2.2), while in Hintikka structures conditions are only placed on the labels of states. These labelling conditions ensure, however, that every Hintikka structure generates, through the constructions described in Section 3.2, a pseudo-model so that membership of formulae in the labels is compliant with the truth in the resultant pseudo-model. We then show how to convert a pseudo-model into a bona fide model in a 'truth-preserving' way.

To describe Hintikka structures, we need the concept of fully expanded set. Such sets contain all the formulae that have to be satisfied locally at the state under consideration. We divide all the formulae that are not elementary in the sense that their satisfaction at the state does not imply satisfaction of any other formulae at the same state (such as $p \in \mathtt{AP}$ or $\neg\mathbf{D}_A\varphi$) into α-formulas and β-formulas. The former are formulae of a conjunctive type, i.e. their truth implies the truth of all

TABLE 1. α- and β-formulas of **CMAEL(CD)** with their respective components

α-formula	α-components
$\neg\neg\varphi$	$\{\varphi\}$
$\varphi\wedge\psi$	$\{\varphi,\psi\}$
$\mathbf{D}_A\varphi$	$\{\mathbf{D}_A\varphi,\varphi\}$
$\mathbf{C}_A\varphi$	$\{\varphi\}\cup\{\mathbf{D}_a\mathbf{C}_A\varphi \mid a\in A\}$

β-formula	β-components
$\neg(\varphi\wedge\psi)$	$\{\neg\varphi,\neg\psi\}$
$\neg\mathbf{C}_A\varphi$	$\{\neg\varphi\}\cup\{\neg\mathbf{D}_a\mathbf{C}_A\varphi \mid a\in A\}$

their α-components at the same state, while the latter are of a disjunctive type: their truth implies the truth of at least one of their β-components at the same state. Table 1 shows the α- and β-formulas of **CMAEL(CD)** together with their α- and β-components. The following claims are straightforward, the cases of common knowledge using Proposition 2.5.

LEMMA 3.1
1. Every α-formula is equivalent to the conjunction of its α-components.
2. Every β-formula is equivalent to the disjunction of its β-components.

DEFINITION 3.2
The *closure of the formula* φ is the smallest set of formulae $\mathsf{cl}(\varphi)$ such that:

1. $\varphi\in\mathsf{cl}(\varphi)$;
2. $\mathsf{cl}(\varphi)$ is closed with respect to α- and β-components of all α- and β-formulae, respectively;
3. for any formula ψ and coalition A, if $\neg\mathbf{D}_A\psi\in\mathsf{cl}(\varphi)$ then $\neg\psi\in\mathsf{cl}(\varphi)$.

DEFINITION 3.3
For any set of formulae Δ we define $\mathsf{cl}(\Delta):=\bigcup\{\mathsf{cl}(\varphi) \mid \varphi\in\Delta\}$. A set of formulae Δ is *closed* if $\Delta=\mathsf{cl}(\Delta)$.

REMARK 3.4
Intuitively, the closure of a set of formulae Γ consists of all formulae that may appear in the tableau whose input is the set of formulae Γ.

DEFINITION 3.5
A set of formulae is *patently inconsistent* if it contains a contradictory pair of formulae φ and $\neg\varphi$.

DEFINITION 3.6
A set Δ of **CMAEL(CD)**-formulae is *fully expanded* if it satisfies the following conditions:

- Δ is not *patently inconsistent*;
- if φ is an α-formula and $\varphi\in\Delta$, then all α-components of φ are in Δ.
- if φ is a β-formula and $\varphi\in\Delta$, then at least one β-component of φ is in Δ.

Intuitively, a non-patently inconsistent set is fully expanded if it is closed under applications of all *local* (pertaining to the same state of a structure) formula decomposition rules.

DEFINITION 3.7
The procedure FULLEXPANSION applies to a set of formulae Γ and produces a (possibly empty) family of sets $\mathcal{FE}(\Gamma)$, called the *family of full expansions of* Γ, obtained as follows: start with the singleton family $\{\Gamma\}$; if Γ is patently inconsistent, halt and return $\mathcal{FE}(\Gamma)=\emptyset$; otherwise repeatedly apply, until saturation, the following *set replacement operations*, each time to a non-deterministically chosen

set Φ from the current family of sets $\mathcal{F}$ and a formula $\varphi \in \Phi$; though, we prioritize the eventualities in Γ so that these formulae are processed first:

1. If φ is an α-formula with α-components φ_1 and φ_2, then replace Φ by $\Phi \cup \{\varphi_1, \varphi_2\}$.
2. If φ is a β-formula such that none of its β-components is in Φ, then replace Φ with the family of extensions

$$\{\Phi \cup \{\psi\} \mid \psi \text{ is a } \beta\text{-component of } \varphi\}$$

3. If $\varphi = \neg \mathbf{C}_A \psi$ and $\neg \psi \notin \Phi$, but some of the other β-components of φ is in Φ, then add to $\mathcal{F}$ the set $\Phi \cup \{\neg \psi\}$

The following proviso applies to the procedure above: if a patently inconsistent set is added to $\mathcal{F}$ as a result of such application, it is removed immediately thereafter.

Saturation occurs when no application of a set replacement operation can change the current family $\mathcal{F}$. At that stage, the family $\mathcal{FE}(\Gamma)$ of sets of formulae is produced and returned. Reaching a stage of saturation is guaranteed to occur because all sets of formulae produced during the procedure FullExpansion are subsets of the finite set $\mathsf{cl}(\Gamma)$.

Notice that the procedure FullExpansion allows adding not more than one β-component of a formula $\varphi = \neg \mathbf{C}_A \psi$ to the initial set, besides $\neg \psi$.

In what follows, we will need the following proposition.

Proposition 3.8
For any finite set of formulae Γ:

$$\Vdash \bigwedge \Gamma \leftrightarrow \bigvee \left\{ \bigwedge \Delta \mid \Delta \in \mathcal{FE}(\Gamma) \right\}.$$

Proof. By Lemma 3.1, every set replacement operation applied to a family $\mathcal{F}$ preserves the formula $\bigvee \{\bigwedge \Delta \mid \Delta \in \mathcal{FE}(\Gamma)\}$ up to logical equivalence. At the beginning, that formula is $\bigwedge \Gamma$, hence the claim follows. ∎

We now define Hintikka structures for **CMAEL(CD)**:

Definition 3.9
A *coalitional multiagent epistemic Hintikka structure* (CMAEHS) is a tuple

$$(\Sigma, S, \{\mathcal{R}_A^D\}_{A \in \mathcal{P}^+(\Sigma)}, \{\mathcal{R}_A^C\}_{A \in \mathcal{P}^+(\Sigma)}, \mathbf{AP}, H)$$

such that:

- $(\Sigma, S, \{\mathcal{R}_A^D\}_{A \in \mathcal{P}^+(\Sigma)}, \{\mathcal{R}_A^C\}_{A \in \mathcal{P}^+(\Sigma)})$ is a CMAES (recall Definition 2.1);
- $\mathbf{AP}$ is a set of atomic propositions;
- H is a labelling of the elements of S with sets of **CMAEL(CD)**-formulae that satisfy the following constraints, for every $s, s' \in S$:

CH1 $H(s)$ is fully expanded;
CH2 If $\neg \mathbf{D}_A \varphi \in H(s)$, then $(s,t) \in \mathcal{R}_A^D$ and $\neg \varphi \in H(t)$, for some $t \in S$;
CH3 If $(s,s') \in \mathcal{R}_A^D$, then $\mathbf{D}_B \varphi \in H(s)$ iff $\mathbf{D}_B \varphi \in H(s')$, for every $B \subseteq A$;
CH4 If $\neg \mathbf{C}_A \varphi \in H(s)$, then $(s,t) \in \mathcal{R}_A^C$ and $\neg \varphi \in H(t)$, for some $t \in S$.

Definition 3.10
Let $\mathcal{H}$ be a CMAEHS with state space S. A **CMAEL(CD)**-formula θ is *satisfiable* in $\mathcal{H}$ if $\theta \in H(s)$, for some $s \in S$. Likewise, a set of **CMAEL(CD)**-formulae Θ is satisfiable in $\mathcal{H}$ if $\Theta \subseteq H(s)$, for some $s \in S$.

3.2 Equivalence of Hintikka structures and models for *CMAEL(CD)*

Here we show that satisfiability in Hintikka structures is equivalent to satisfiability in models. For brevity, we only deal with single formulae; the extension to finite sets of formulae is straightforward. The main complications in the proofs below arise due to the presence of distributed knowledge operators in the language of a logic.

Here we will prove that a **CMAEL(CD)**-formula θ is satisfiable in a CMAEM iff it is satisfiable in a CMAEHS. First, we show that satisfiability in a CMAEM implies satisfiability in a CMAEHS. Then, we show that satisfiability in a CMAEHS implies satisfiability in a pseudo-CMAEM, which in turn implies satisfiability in a CMAEM.

That satisfiability in a CMAEM implies satisfiability in a CMAEHS is almost immediate. Given a CMAEM $\mathcal{M}$ with a set of states S, define the *extended labelling function* $L^+_{\mathcal{M}}$ from S to the power-set of **CMAEL(CD)**-formulae as follows: $L^+_{\mathcal{M}}(s) = \{\varphi \mid \mathcal{M}, s \Vdash \varphi\}$. It is then routine to check the following.

Lemma 3.11
Let $\mathcal{M} = (\Sigma, S, \{\mathcal{R}^D_A\}_{A \in \mathcal{P}^+(\Sigma)}, \{\mathcal{R}^C_A\}_{A \in \mathcal{P}^+(\Sigma)}, \mathbf{AP}, L)$ be a CMAEM satisfying θ and let $L^+_{\mathcal{M}}$ be the extended labelling on $\mathcal{M}$. Then, $(\Sigma, S, \{\mathcal{R}^D_A\}_{A \in \mathcal{P}^+(\Sigma)}, \{\mathcal{R}^C_A\}_{A \in \mathcal{P}^+(\Sigma)}, \mathbf{AP}, L^+_{\mathcal{M}})$ is a CMAEHS satisfying θ. Therefore, satisfiability in a CMAEM implies satisfiability in a CMAEHS.

For the converse direction we need two steps, done in Lemma 3.12 and Lemma 3.14.

Lemma 3.12
Let θ be a **CMAEL(CD)**-formula satisfiable in a CMAEHS. Then, θ is satisfiable in a pseudo-CMAEM.

Proof. Let $\mathcal{H} = (\Sigma, S, \{\mathcal{R}^D_A\}_{A \in \mathcal{P}^+(\Sigma)}, \{\mathcal{R}^C_A\}_{A \in \mathcal{P}^+(\Sigma)}, \mathbf{AP}, H)$ be an CMAEHS for θ. We construct a pseudo-CMAEM $\mathcal{M}'$ satisfying θ out of $\mathcal{H}$ as follows.

First, for every $A \in \mathcal{P}^+(\Sigma)$, let $\mathcal{R}'^D_A$ be the reflexive, symmetric and transitive closure of $\bigcup_{A \subseteq B} \mathcal{R}^D_B$ and let $\mathcal{R}'^C_A$ be the transitive closure of $\bigcup_{a \in A} \mathcal{R}'^D_a$. Thus, both $\mathcal{R}'^D_A$ and $\mathcal{R}'^C_A$ are equivalence relations and $\mathcal{R}^D_A \subseteq \mathcal{R}'^D_A$ and $\mathcal{R}^C_A \subseteq \mathcal{R}'^C_A$, for every $A \in \mathcal{P}^+(\Sigma)$. Second, let $L(s) = H(s) \cap \mathbf{AP}$, for every $s \in S$. It is then immediate to check that $B \subseteq A$ implies $\mathcal{R}'^D_A \subseteq \mathcal{R}'^D_B$, and hence, $\mathcal{M}' = (\Sigma, S, \{\mathcal{R}'^D_A\}_{A \in \mathcal{P}^+(\Sigma)}, \{\mathcal{R}'^C_A\}_{A \in \mathcal{P}^+(\Sigma)}, L)$ is a pseudo-CMAEM.

Basically this construction relabels the edges of a Hintikka structure such that if a directed edge is labelled with a coalition A, it is made bidirectional and is further labelled with all coalitions that are subsets of A. Hereafter the relation is then made transitive and reflexive. The labels of the states are reduced to only containing (positive) atoms. Figure 1 illustrates the process of transforming the Hintikka structure on the left into the pseudo-model on the right.

To complete the proof of the lemma, we show, by induction on the structure of the formulae in $\mathsf{cl}(\theta)$ that, for every $s \in S$ and every formula χ, the following hold:

$$(i) \quad \chi \in H(s) \text{ implies } \mathcal{M}', s \Vdash \chi;$$
$$(ii) \quad \neg\chi \in H(s) \text{ implies } \mathcal{M}', s \Vdash \neg\chi.$$

The statement of the lemma then follows.

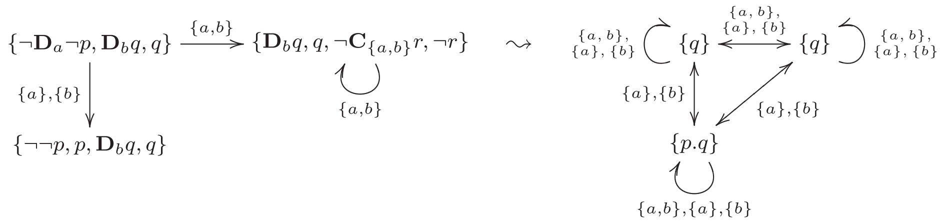


FIG. 1. Example on transforming a Hintikka structure to a pseudo-model using the construction from the proof of Lemma 3.12.

Let χ be some $p \in \mathtt{AP}$. Then, $p \in H(s)$ implies $p \in L(s)$ and, thus, $\mathcal{M}', s \Vdash p$; if, on the other hand, $\neg p \in H(s)$, then due to (CH1), $p \notin H(s)$ and thus $p \notin L(s)$; hence, $\mathcal{M}', s \Vdash \neg p$.

Assume that the claim holds for all subformulae of χ; then, we have to prove that it holds for χ, as well.

Suppose that χ is $\neg\varphi$. If $\neg\varphi \in H(s)$, then the inductive hypothesis immediately gives us $\mathcal{M}', s \Vdash \neg\varphi$; if, on the other hand, $\neg\neg\varphi \in H(s)$, then by virtue of (CH1), $\varphi \in H(s)$ and hence, by inductive hypothesis, $\mathcal{M}', s \Vdash \varphi$ and thus $\mathcal{M}', s \Vdash \neg\neg\varphi$.

The case of $\chi = \varphi \wedge \psi$ is straightforward, using (CH1).

Suppose that χ is $\mathbf{D}_A\varphi$. Assume, first, that $\mathbf{D}_A\varphi \in H(s)$. In view of the inductive hypothesis, it suffices to show that $(s,t) \in \mathcal{R}'^D_A$ implies $\varphi \in H(t)$. So, assume that $(s,t) \in \mathcal{R}'^D_A$. There are two cases to consider. If $s = t$, then the conclusion immediately follows from (CH1). If, on the other hand, $s \neq t$, then there exists an undirected path between s and t along the relations of the form $\mathcal{R}^D_B$, where each B is a superset of A. Then, in view of (CH3), $\mathbf{D}_A\varphi \in H(t)$; hence, by (CH1), $\varphi \in H(t)$, as desired.

Assume, next, that $\neg\mathbf{D}_A\varphi \in H(s)$. In view of the inductive hypothesis, it suffices to show that there exists $t \in S$ such that $(s,t) \in \mathcal{R}'^D_A$ and $\neg\varphi \in H(t)$. By (CH2), there exists $t \in S$ such that $(s,t) \in \mathcal{R}^D_A$ and $\neg\varphi \in H(t)$. As $\mathcal{R}^D_A \subseteq \mathcal{R}'^D_A$, the desired conclusion follows.

Suppose now that χ is $\mathbf{C}_A\varphi$. Assume that $\mathbf{C}_A\varphi \in H(s)$. In view of the inductive hypothesis, it suffices to show that if t is A-reachable from s in $\mathcal{M}'$, then $\varphi \in H(t)$. So, assume that either $s = t$ or, for some $n \geq 1$, there exists a sequence of states $s = s_0, s_1, \ldots, s_{n-1}, s_n = t$ such that, for every $0 \leq i < n$, there exists $a_i \in A$ such that $(s_i, s_{i+1}) \in \mathcal{R}'^D_{a_i}$. In the former case, the desired conclusion follows from (CH1). In the latter, we can show by induction on i, for $0 \leq i < n$, using (CH3) and (CH1), that $\mathbf{D}_{a_i}\mathbf{C}_A\varphi \in H(s_i)$. Then, in particular, $\mathbf{D}_{a_{n-1}}\mathbf{C}_A\varphi \in H(s_{n-1})$, and again, by (CH3), $\mathbf{D}_{a_{n-1}}\mathbf{C}_A\varphi \in H(t)$ and thus by (CH1), $\mathbf{C}_A\varphi \in H(t)$ and $\varphi \in H(t)$.

Assume, on the other hand, that $\neg\mathbf{C}_A\varphi \notin H(s)$. Then, the desired conclusion follows from (CH4), the inclusion $\mathcal{R}^C_A \subseteq \mathcal{R}'^C_A$, and the inductive hypothesis.

∎

We now prove that satisfiability in a pseudo-CMAEM implies satisfiability in a CMAEM. To that end, we use a modification of the construction from [10, Appendix A1] to show that if θ is satisfiable in a pseudo-CMAEM, then it is satisfiable in a 'tree-like' pseudo-CMAEM that actually turns out to be a bona fide CMAEM. To present the proof, we need some preliminary definitions.

DEFINITION 3.13
Let $\mathcal{M} = (\Sigma, S, \{\mathcal{R}^D_A\}_{A \in \mathcal{P}^+(\Sigma)}, \{\mathcal{R}^C_A\}_{A \in \mathcal{P}^+(\Sigma)}, \mathtt{AP}, L)$ be a (pseudo-) CMAEM and let $s, t \in S$. A *maximal path from s to t* in $\mathcal{M}$ is a sequence $s_0, A_0, s_1, A_1, \ldots, s_{n-1}, A_{n-1}, s_n$ where $s = s_0$ and $t = s_n$, such that $n = 0$ and $s = t$ or, for every $0 \leq i < n$, $(s_i, s_{i+1}) \in \mathcal{R}^D_{A_i}$, but $(s_i, s_{i+1}) \in \mathcal{R}^D_B$ does not hold for any B with $A_i \subset B \subseteq \Sigma$. A segment ρ' of a maximal path ρ starting and ending with a state is a *sub-path* of ρ.

Notice that, in general, there might be several maximal paths between a pair of states.

For a path $\tau = s_0, A_0, s_1, \ldots, s_{n-1}, A_{n-1}, s_n$, we denote by $\tau_{|i}$ the sub-path of τ starting in s_0 and ending in s_i, i.e. $\tau_{|i} = s_0, A_0, s_1, \ldots, A_{i-1}, s_i$ and by $|\tau|$ the length of τ, i.e. n. We denote the last element of a path τ, which is a state, by $l(\tau)$ and the second last element of τ, which is a coalition, by $sl(\tau)$.

Lemma 3.14
Let θ be a **CMAEL(CD)**-formula satisfiable in a pseudo-CMAEM; then, θ is satisfiable in a CMAEM.

Proof. Suppose that θ is satisfied in a pseudo-CMAEM $\mathcal{M}$ at state s. Let $\mathcal{M}_s = (\Sigma, S, \{\mathcal{R}^D_A\}_{A\in\mathcal{P}^+(\Sigma)}, \{\mathcal{R}^C_A\}_{A\in\mathcal{P}^+(\Sigma)}, \mathtt{AP}, L)$ be the submodel of $\mathcal{M}$ generated by s. Then, $\mathcal{M}_s, s \Vdash \theta$ since $\mathcal{M}_s$ and $\mathcal{M}$ are locally bisimilar at s. Next, we unravel $\mathcal{M}_s$ into a model $\mathcal{M}^* = (\Sigma, S^*, \{\mathcal{R}^{*D}_A\}_{A\in\mathcal{P}^+(\Sigma)}, \{\mathcal{R}^{*C}_A\}_{A\in\mathcal{P}^+(\Sigma)}, \mathtt{AP}, L^*)$, as follows.

First, call a maximal path ρ in $\mathcal{M}_s$ an s-max-path if the first component of ρ is s, and let S^* be the set of all s-max-paths in $\mathcal{M}_s$. Notice that s by itself is an s-max-path with $l(s) = s$.

For every $A \in \mathcal{P}^+(\Sigma)$, let

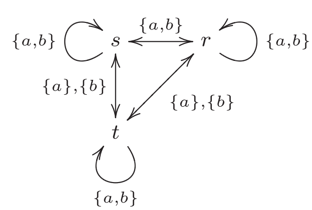
$$\mathcal{R}'^D_A = \{(\rho,\tau) \mid \rho,\tau \in S^*,\ \tau_{||\tau|-1} = \rho \text{ and } sl(\tau) \supseteq A\},$$

i.e. $(\rho,\tau) \in \mathcal{R}'^D_A$ if τ extends ρ with one step labelled by a coalition containing A. Next, let $\mathcal{R}^{*D}_A$ be a reflexive, symmetric and transitive closure of $\mathcal{R}'^D_A$. Notice that $(\rho,\tau) \in \mathcal{R}^{*D}_A$ holds for two distinct paths ρ and τ iff there exists a sequence $\rho_0, \ldots, \rho_n \in S^*$ with $\rho = \rho_0$ and $\tau = \rho_n$ such that for all $i < n$, either $(\rho_i, \rho_{i+1}) \in \mathcal{R}'^D_A$ or $(\rho_{i+1}, \rho_i) \in \mathcal{R}'^D_A$. It then follows that the following *downward closure condition* holds:

$$\textbf{(DC)}\ \text{If } (\rho,\tau) \in \mathcal{R}^{*D}_A \text{ and } B \subseteq A, \text{ then } (\rho,\tau) \in \mathcal{R}^{*D}_B.$$

The relations $\mathcal{R}^{*C}_A$ are defined as in any CMAEF. To complete the definition of $\mathcal{M}^*$, we put $L^*(\rho) = L(l(\rho))$, for every $\rho \in S^*$. Notice that $\mathcal{M}^*$ is *tree-like* in the sense that the structure $(S^*, \{\mathcal{R}'^D_A\}_{A\in\mathcal{P}^+(\Sigma)})$ is a tree.

By this construction we basically remove all 'non-maximal' edges between two vertices from the part of the given pseudomodel that can be reached by the given state s. Then we build paths by starting in s and then traversing the resulting graph via the edges. E.g. if we consider the pseudo-model $\mathcal{M}$ in Figure 1, and we let the top-left-most state be s, then $\mathcal{M}, s \Vdash \neg\mathbf{D}_a\neg p \wedge \mathbf{D}_b q$. S^* will in this case be all paths starting in s and following the links in the graph.



I.e. $\rho = (s,\{a,b\},s,\{a\},t)$ and $\tau = (s,\{a,b\},r,\{b\},t)$ are in S^*, while $\rho' = (s,\{a\},s,\{b\},t) \notin S^*$.

We have $(\rho,\tau) \notin \mathcal{R}^{*D}_a$, $(\rho,\tau) \notin \mathcal{R}^{*D}_b$ and $(\rho,\tau) \notin \mathcal{R}^{*D}_{a,b}$. On the other hand, $(\tau,(s,\{a,b\},r,\{a,b\},s)) \in \mathcal{R}^{*D}_b$.

In this example, $L^*(\rho) = L^*(\tau) \stackrel{\text{def.}}{=} L(t) = \{p,q\}$.

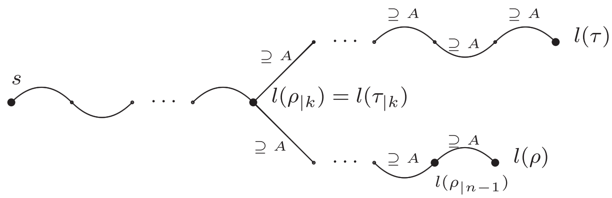


FIG. 2. The situation from (3.1) drawn in $\mathcal{M}$, i.e. the dots/circles belongs to S, and the links are links in $\mathcal{R}^D$.

It is clear from the construction, namely from **(DC)**, that $\mathcal{M}^*$ is a pseudo-CMAEM, and in the following, we will show that condition (†) of Definition 2.2 also holds, so that $\mathcal{M}^*$ is a CMAEM.

First, we notice that, since $\mathcal{M}^*$ is tree-like, we have $(\rho,\tau)\in\mathcal{R}^{*D}_A$ iff there exists $k\geq 0$, with $k\leq|\rho|$ and $k\leq|\tau|$, such that

$$\begin{gathered}\rho_{|k}=\tau_{|k},\ \text{and}\\ \text{for all } k<i\leq|\tau| \text{ and } k<j\leq|\rho|, A\subseteq sl(\tau_{|i}) \text{ and } A\subseteq sl(\rho_{|j}).\end{gathered}\tag{3.1}$$

(The situation is depicted in Figure 2.) As stated, we have to prove that $\mathcal{R}^{*D}_A=\bigcap_{a\in A}\mathcal{R}^{*D}_a$ for every $A\in\mathcal{P}^+(\Sigma)$. The left-to-right inclusion immediately follows from **(DC)**. For the converse, assume that $(\rho,\tau)\in\mathcal{R}^{*D}_a$ holds for every $a\in A$. Then, for every $a\in A$, according to (3.1), there exists $k_a\geq 0$ such that $\rho_{|k_a}=\tau_{|k_a}$ and $\{a\}\subseteq sl(\tau_{|i}), sl(\rho_{|j})$ for every $|\tau|\geq i>k_a$ and every $|\rho|\geq j>k_a$. Now, let k be the largest k_a satisfying this condition (such a k exists since $\mathcal{M}^*$ is tree-like). Then, $\rho_{|k}=\tau_{|k}$, and for every $a\in A$, the inclusions $\{a\}\subseteq sl(\tau_{|i})$ and $\{a\}\subseteq sl(\rho_{|j})$ hold for every $|\tau|\geq i>k$ and every $|\rho|\geq j>k$. Therefore, condition (3.1) is fulfilled for A and k, and hence $(\rho,\tau)\in\mathcal{R}^{*D}_A$, as desired.

Finally, it remains to prove that $\mathcal{M}^*$ satisfies θ. From (3.1) we see, that if $(\rho,\tau)\in\mathcal{R}^{*D}_A$, then $(l(\rho),l(\tau))\in\mathcal{R}^D_A$, since every $\mathcal{R}^D_A$ is an equivalence relation. It is now easy to check that the relation $Z=\{(\rho,l(\rho))\mid\rho\in S^*\}$ is a bisimulation between $\mathcal{M}^*$ and $\mathcal{M}_s$. Since $(s,l(s))\in Z$, it follows that $\mathcal{M}^*,s\Vdash\theta$, and we are done. ∎

THEOREM 3.15

Let θ be a **CMAEL(CD)**-formula. Then, θ is satisfiable in a CMAEHS iff it is satisfiable in a CMAEM.

PROOF. Immediate from Lemmas 3.11, 3.12 and 3.14. ∎

4 Tableau procedure for testing satisfiability in CMAEL(CD)

In this section, we present our tableau algorithm for checking (constructive) satisfiability of formulae of **CMAEL(CD)**. We start off by explaining the general philosophy underlying our tableau procedure and then present it in detail.

4.1 Basic ideas and overview of the tableau procedure

Traditionally, the propositional tableau method works by decomposing the formula whose satisfiability is being tested into its α-, resp. β- components—repeatedly, until producing all full expansions of that formula. All these components belong to the closure of the input formula. When the closure is finite (as it is usually the case with modal and temporal logics) the termination of the tableau-building procedure is guaranteed because there are only finitely many full expansions.

Furthermore, in the tableau method for the classical propositional logic that decomposition into components produces a tree representing an exhaustive search for a Hintikka set, the propositional analogue of Hintikka structures, for the input formula. If at least one branch of the tree remains open, it produces a full expansion of the input formula, which is a Hintikka set for this formula. In this case, the formula is pronounced satisfiable; otherwise, it is declared unsatisfiable. In the case of modal and temporal logics, local decomposition steps, producing full expansions, are interleaved with steps along the accessibility/transition relations, producing sets of formulae that are supposed to be true at *successors* of the current state. These sets are subjected, again, to local decomposition into components, eventually producing their full expansions, etc. In order to distinguish fully expanded sets from those produced after transition to successors, we will deal with two types of nodes of the tableau, respectively called 'states' and 'prestates'. In order to ensure termination of the construction process, we will systematically reuse states and prestates labelled with the same sets of formulae.

The tableau procedure for testing a formula θ for satisfiability attempts to construct a non-empty graph $\mathcal{T}^\theta$ (called itself a *tableau*) representing 'sufficiently many' CMAEHSs for θ in the sense that if θ is satisfiable in any CMAEHS, then it is satisfiable in a CMAEHS represented by the tableau. The procedure consists of three major sub-procedures, or phases: *construction*, *prestate elimination* and *state elimination*. During the construction phase, we build the *pretableau* $\mathcal{P}^\theta$—a directed graph whose nodes are sets of formulae of two types: *states*[4] and *prestates*, as explained above. States represent (labels of) states of the CMAEHSs that the tableau attempts to construct, while prestates are only used temporarily, during the construction phase.

During the prestate elimination phase, we create a smaller graph $\mathcal{T}_0^\theta$ out of $\mathcal{P}^\theta$, called the *initial tableau for* θ, by eliminating all the prestates of $\mathcal{P}^\theta$ and adjusting its edges, as prestates have already fulfilled their role of keeping the graph finite and can, therefore, be discharged.

In the case of classical propositional logic, the only reason why it may turn out to be impossible to produce a Hintikka set for the input formula is that every attempt to build such a set results in a collection of formulae containing a patent inconsistency. In the case of logics with fixpoint-definable operators, such as **CMAEL(CD)**, there are two other reasons for a tableau not to correspond to any Hintikka structure for the input formula. The first one has to do with *realization of eventualities*—formulas of the form $\neg \mathbf{C}_A\varphi$, whose truth condition requires that $\neg\varphi$ 'eventually' becomes true—in the tableau graph. Applying decomposition rules to eventualities in the construction of the tableau can postpone indefinitely the realization by keeping 'promising' that the realization will happen further down the line, while that 'promise' never becomes fulfilled. Therefore, a 'good' tableau should not contain states with unrealized eventualities. The other additional reason for the resultant tableau not to represent a Hintikka structure is that some states do not have all the successors they would be required to have in a corresponding Hintikka structure (for example, because those successors have been removed for not realizing eventualities).

During the state elimination phase, we remove from $\mathcal{T}_0^\theta$ all states, if any, that cannot be satisfied in any CMAEHS for any of the reasons suggested above and discussed in more detail further (excluding patently inconsistent sets, which are removed 'on the fly' during the construction phase). The elimination procedure results in a (possibly empty) subgraph $\mathcal{T}^\theta$ of $\mathcal{T}_0^\theta$, called the *final tableau for* θ. If some state Δ of $\mathcal{T}^\theta$ contains θ, it is declared satisfiable; otherwise, θ is declared unsatisfiable.

[4]From now on we will use the term 'state' in two related but distinct senses: as a state of a tableau and as a state of a semantic structure (frame, model, Hintikka structure). The use of term 'state' will usually be clear from the context or explicitly specified.

The logic **CMAEL(CD)** involves modal operators over equivalence relations, and thus invokes some typical complications in the tableau-building procedures associated with inverse-looking modalities, see e.g. [19]: every box occurring in the label of a descendant state has a backwards effect on all predecessor states, incl. the current state. In order to deal with these complications we must either organize a mechanism for backtracking and *backwards propagation of box-formulae*, or a mechanism for anticipation of the occurrence of such boxes in the future, coming from subformulae of formulae in the label of the current state, based on *analytic cut rules*. We will adopt here the latter approach, which is easier to describe and implement into what we call a *diamond-propagating* procedure, by employing suitably restricted analytic cut rules to maintain the efficiency of the procedure, but later we will briefly discuss the former alternative, too. The two procedures only differ in the construction phase; the prestate and state elimination phases are common to both. The need and use of analytic cut rules is illustrated later in Example 5.6.

4.2 Cut-saturated sets and expansions

The application of the analytic cut, mentioned above, is implemented by imposing an additional *cut-saturating rule* on the construction of the full expansions of a given set of formulae. In order to prevent the unnecessary swelling and proliferation of states, we will restrict the application of that rule by imposing generic restrictions which, on the other hand, should be sufficiently relaxed to guarantee the completeness of the tableau procedure. These generic conditions, which will be specified later, will be imposed separately on the two types of box-formulae in **CMAEL(CD)**, viz. $\mathbf{D}_A$-formulae and on $\mathbf{C}_A$-formulae.

Definition 4.1
Given restrictive conditions C_1 and C_2, a set Δ of **CMAEL(CD)**-formulas is (C_1, C_2)-*cut-saturated* if it satisfies the following conditions, where $\mathsf{Sub}(\psi)$ is the set of subformulae of a formula ψ:

CS0 Δ is fully expanded (recall Definition 3.6).
CS1 For any $\mathbf{D}_A\varphi \in \mathsf{Sub}(\psi)$ where $\psi \in \Delta$, if condition C_1 holds then either $\mathbf{D}_A\varphi \in \Delta$ or $\neg\mathbf{D}_A\varphi \in \Delta$.
CS2 For any $\mathbf{C}_A\varphi \in \mathsf{Sub}(\psi)$ where $\psi \in \Delta$, if condition C_2 holds then either $\mathbf{C}_A\varphi \in \Delta$ or $\neg\mathbf{C}_A\varphi \in \Delta$.

We note that CS1 and CS2 are semantically sound rules, no matter what C_1 and C_2 are, as they cannot make a tableau closed if the input formula is satisfiable. On the other hand, if C_1 and C_2 are too strong, that may prevent the tableau from closing and thus yield an incomplete tableau procedure, as will become apparent later. Again, the reason we would want to make C_1 and C_2 as strong as possible is to avoid branching on too many formulae, causing an unnecessary large state space and resulting in a practically less efficient procedure.

Hereafter, we will omit the explicit mention of the conditions C_1 and C_2, unless necessary. In fact, for now we can assume both C_1 and C_2 to be True, but later we will introduce non-trivial restrictive conditions.

Definition 4.2
The family $\mathcal{CSE}(\Gamma)$ of *cut-saturated expansions (CS-expansions) of a set of formulae* Γ is defined by expanding the procedure FullExpansion with the following two set-replacement rules, again applied to a non-deterministically chosen set Φ from the current family and a formula $\psi \in \Phi$:

1. For any formula $\mathbf{D}_A\varphi$ that is a subformula of ψ such that C_1 is satisfied, replace Φ with the two extensions of Φ obtained by adding respectively $\mathbf{D}_A\varphi$ and $\neg\mathbf{D}_A\varphi$ to it.
2. For any formula $\mathbf{C}_A\varphi$ that is a subformula of ψ such that C_2 is satisfied, replace Φ with the two extensions of Φ obtained by adding respectively $\mathbf{C}_A\varphi$ and $\neg\mathbf{C}_A\varphi$ to it.

It is clear from the definition that all sets in $\mathcal{CSE}(\Gamma)$ are (C_1, C_2)-cut-saturated .

Definition 4.3
The *extended closure* of θ, denoted $\mathsf{ecl}(\theta)$, is the smallest set such that $\varphi, \neg\varphi \in \mathsf{ecl}(\theta)$ for every $\varphi \in \mathsf{cl}(\theta)$. The extended closure $\mathsf{ecl}(\Gamma)$ of a set of formulae Γ is defined likewise.

The following is immediate from the definitions.

Lemma 4.4
Every CS-expansion of a set of formulae Γ is a subset of $\mathsf{ecl}(\Gamma)$.

Lemma 4.5
For any **CMAEL(CD)**-formula θ, the size of (i.e. number of formulae in) the extended closure of θ is $\mathcal{O}(k \cdot |\theta|)$, where k is the number of agents occurring in θ.

Proof. Straightforward. ∎

4.2.1 Construction phase

As already mentioned, a tableau algorithm attempts to produce a compact representation of ‘sufficiently many’ CMAEHSs for the input formula; in this attempt, it sets in motion an exhaustive search for such CMAEHSs. As a result, the pretableau $\mathcal{P}^{\theta}$ built at this phase contains two types of edge, as well as two types of node (states and prestates; see above).

One type of edge, depicted by unmarked, dashed uni-directed arrows $\dashrightarrow$, represents the search dimension of the tableaux. The exhaustive search considers all possible alternatives arising when prestates are expanded into states by branching in the ‘disjunctive’ cases. Thus, when we draw unmarked arrows from a prestate Γ to each state from a set of states X, this intuitively means that, in any CMAEHS, a state satisfying Γ has to satisfy at least one of the states in X.

The second type of edge represents transition relations in the CMAEHSs that the procedure attempts to build. Accordingly, this type of edges is represented by solid, uni-directed arrows, $\longrightarrow$, marked with formulae whose presence in one of the end nodes requires the presence in the tableau of the other end node, reachable by a particular relation. Intuitively, if $\neg \mathbf{D}_A\varphi \in \Delta$ for some state Δ, then some (state obtained from a) prestate Γ containing $\neg\varphi$ must be accessible from Δ by relation $\mathcal{R}^D_A$. We mark these arrows with the respective formulae $\neg \mathbf{D}_A\varphi$ in order to keep track of the specific reason for creating that particular state. That information will be needed during the elimination phases.

We now turn to presenting the rules of the ‘diamond-propagating’ construction phase, each of which creates a different type of edge, as discussed above. The first rule, **(SR)**, prescribes how to create states from prestates, while **(DR)** expands prestates into states.

Rule (SR) Given a prestate Γ, such that **(SR)** has not been applied to it before, do the following:

1. Add to the pretableau all CS-expansions Δ of Γ; declare these to be *states*;
2. For each so obtained state Δ, put $\Gamma \dashrightarrow \Delta$;
3. If, however, the pretableau already contains a state $\Delta' = \Delta$, then do not create a new state, but put $\Gamma \dashrightarrow \Delta'$.

We denote by **states**(Γ) the (finite) set $\{\Delta \mid \Gamma \dashrightarrow \Delta\}$.

Rule (DR): Given a state Δ such that $\neg \mathbf{D}_A\varphi \in \Delta$ and **(DR)** has not been applied to Δ with respect to $\neg \mathbf{D}_A\varphi$ before, do the following:

1. Add to the pretableau the set $\Gamma = \{\neg\varphi\} \cup \{\mathbf{D}_{A'}\psi \in \Delta \mid A' \subseteq A\} \cup \{\neg\mathbf{D}_{A'}\psi \in \Delta \mid A' \subseteq A$ and $\neg\mathbf{D}_{A'}\psi \neq \neg\mathbf{D}_A\varphi\} \cup \{\neg\mathbf{C}_{A'}\psi \in \Delta \mid A' \cap A \neq \emptyset\}$ and declare this set to be a *prestate*.
2. Put $\Delta \xrightarrow{\neg\mathbf{D}_A\varphi} \Gamma$.
3. If, however, the pretableau already contains a prestate $\Gamma' = \Gamma$, then do not create a new prestate, but put $\Delta \xrightarrow{\neg\mathbf{D}_A\varphi} \Gamma'$.

When building a tableau for a formula θ, the construction phase begins with creating a single prestate $\{\theta\}$. Afterwards, we alternate between **(SR)** and **(DR)**: first, **(SR)** is applied to the prestates created at the previous stage of the construction, then **(DR)** is applied to the states created at the previous stage.

The construction phase is completed when every prestate required to be added to the pretableau has already been added (as prescribed in item 3 of **(SR)**) and **(DR)** does not apply to any of the states with respect to any of the formulae.

EXAMPLE 4.6
Let us construct the pretableau for the formula $\theta = \neg\mathbf{D}_{\{a,c\}}\mathbf{C}_{\{a,b\}}p \wedge \mathbf{C}_{\{a,b\}}(p\wedge q)$, assuming that $\Sigma = \{a,b,c\}$. To save space, we replace θ by the set of its conjuncts $\Theta = \{\neg\mathbf{D}_{\{a,c\}}\mathbf{C}_{\{a,b\}}p, \mathbf{C}_{\{a,b\}}(p\wedge q)\}$.

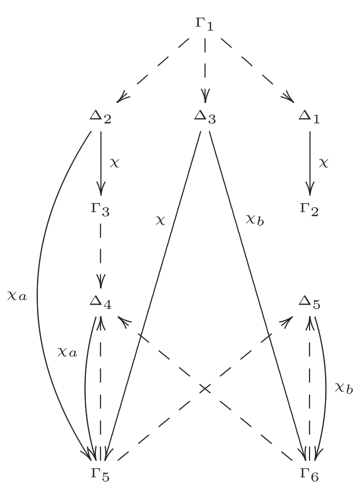
Here and further on in the examples, we let $\underline{\mathbf{C}_A\varphi}$ denote the set $\{\mathbf{C}_A\varphi, \varphi\} \cup \bigcup_{a\in A} \mathbf{D}_a\mathbf{C}_A\varphi$. Figure 3 shows the pretableau for Θ.

4.2.2 Prestate elimination phase

At this phase, we remove from pretableau $\mathcal{P}^\theta$ all the prestates and unmarked arrows, by applying the following rule (the resultant graph is denoted $\mathcal{T}_0^\theta$ and is called the *initial tableau*):

(PR) For every prestate Γ in $\mathcal{P}^\theta$, do the following:

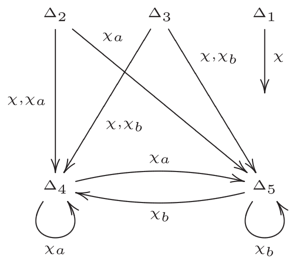
1. Remove Γ from $\mathcal{P}^\theta$;
2. If there is a state Δ in $\mathcal{P}^\theta$ with $\Delta \xrightarrow{\chi} \Gamma$, then for every state $\Delta' \in \mathbf{states}(\Gamma)$, put $\Delta \xrightarrow{\chi} \Delta'$;



$$\chi = \neg\mathbf{D}_{\{a,c\}}\mathbf{C}_{\{a,b\}}p$$
$$\chi_a = \neg\mathbf{D}_a\mathbf{C}_{\{a,b\}}p$$
$$\chi_b = \neg\mathbf{D}_b\mathbf{C}_{\{a,b\}}p$$

$$\Gamma_1 = \{\neg\mathbf{D}_{\{a,c\}}\mathbf{C}_{\{a,b\}}p, \mathbf{C}_{\{a,b\}}(p\wedge q)\}$$
$$\Delta_1 = \{\neg\mathbf{D}_{\{a,c\}}\mathbf{C}_{\{a,b\}}p, \underline{\mathbf{C}_{\{a,b\}}(p\wedge q)}, p, q, \underline{\mathbf{C}_{\{a,b\}}p}\}$$
$$\Delta_2 = \{\neg\mathbf{D}_{\{a,c\}}\mathbf{C}_{\{a,b\}}p, \underline{\mathbf{C}_{\{a,b\}}(p\wedge q)}, p, q, \neg\mathbf{C}_{\{a,b\}}p, \neg\mathbf{D}_a\mathbf{C}_{\{a,b\}}p\}$$
$$\Delta_3 = \{\neg\mathbf{D}_{\{a,c\}}\mathbf{C}_{\{a,b\}}p, \underline{\mathbf{C}_{\{a,b\}}(p\wedge q)}, p, q, \neg\mathbf{C}_{\{a,b\}}p, \neg\mathbf{D}_b\mathbf{C}_{\{a,b\}}p\}$$
$$\Gamma_2 = \{\neg\mathbf{C}_{\{a,b\}}p, \mathbf{D}_a\mathbf{C}_{\{a,b\}}(p\wedge q), \mathbf{D}_a\mathbf{C}_{\{a,b\}}p\}$$
$$\Gamma_3 = \{\neg\mathbf{C}_{\{a,b\}}p, \mathbf{D}_a\mathbf{C}_{\{a,b\}}(p\wedge q), \neg\mathbf{D}_a\mathbf{C}_{\{a,b\}}p\}$$
$$\Delta_4 = \{\neg\mathbf{C}_{\{a,b\}}p, \underline{\mathbf{C}_{\{a,b\}}(p\wedge q)}, p, q, \neg\mathbf{D}_a\mathbf{C}_{\{a,b\}}p\}$$
$$\Delta_5 = \{\neg\mathbf{C}_{\{a,b\}}p, \underline{\mathbf{C}_{\{a,b\}}(p\wedge q)}, p, q, \neg\mathbf{D}_b\mathbf{C}_{\{a,b\}}p\}$$
$$\Gamma_5 = \{\neg\mathbf{C}_{\{a,b\}}p, \mathbf{D}_a\mathbf{C}_{\{a,b\}}(p\wedge q)\}$$
$$\Gamma_6 = \{\neg\mathbf{C}_{\{a,b\}}p, \mathbf{D}_b\mathbf{C}_{\{a,b\}}(p\wedge q)\}$$

FIG. 3. The pretableau for $\{\neg\mathbf{D}_{\{a,c\}}\mathbf{C}_{\{a,b\}}p, \mathbf{C}_{\{a,b\}}(p\wedge q)\}$.



$$\chi = \neg\mathbf{D}_{\{a,c\}}\mathbf{C}_{\{a,b\}}p$$
$$\chi_a = \neg\mathbf{D}_a\mathbf{C}_{\{a,b\}}p$$
$$\chi_b = \neg\mathbf{D}_b\mathbf{C}_{\{a,b\}}p$$

$$\Delta_1 = \{\neg\mathbf{D}_{\{a,c\}}\mathbf{C}_{\{a,b\}}p, \underline{\mathbf{C}_{\{a,b\}}(p\wedge q)}, p, q, \underline{\mathbf{C}_{\{a,b\}}p}\}$$
$$\Delta_2 = \{\neg\mathbf{D}_{\{a,c\}}\mathbf{C}_{\{a,b\}}p, \underline{\mathbf{C}_{\{a,b\}}(p\wedge q)}, p, q, \neg\mathbf{C}_{\{a,b\}}p, \neg\mathbf{D}_a\mathbf{C}_{\{a,b\}}p\}$$
$$\Delta_3 = \{\neg\mathbf{D}_{\{a,c\}}\mathbf{C}_{\{a,b\}}p, \underline{\mathbf{C}_{\{a,b\}}(p\wedge q)}, p, q, \neg\mathbf{C}_{\{a,b\}}p, \neg\mathbf{D}_b\mathbf{C}_{\{a,b\}}p\}$$
$$\Delta_4 = \{\neg\mathbf{C}_{\{a,b\}}p, \underline{\mathbf{C}_{\{a,b\}}(p\wedge q)}, p, q, \neg\mathbf{D}_a\mathbf{C}_{\{a,b\}}p\}$$
$$\Delta_5 = \{\neg\mathbf{C}_{\{a,b\}}p, \underline{\mathbf{C}_{\{a,b\}}(p\wedge q)}, p, q, \neg\mathbf{D}_b\mathbf{C}_{\{a,b\}}p\}$$

FIG. 4. The initial tableau for $\{\neg\mathbf{D}_{\{a,c\}}\mathbf{C}_{\{a,b\}}p, \mathbf{C}_{\{a,b\}}(p\wedge q)\}$.

EXAMPLE 4.7
We continue Example 4.6 by creating the initial tableau for $\Theta = \{\neg\mathbf{D}_{\{a,c\}}\mathbf{C}_{\{a,b\}}p,\ \mathbf{C}_{\{a,b\}}(p\wedge q)\}$ out of the pretableau in Figure 3. Again we let $\underline{\mathbf{C}_A\varphi}$ denote the set consisting of $\mathbf{C}_A\varphi$ and its α-components. Figure 4 shows the resulting initial tableau.

4.2.3 State elimination phase

During this phase, we remove from $\mathcal{T}_0^\theta$ states that are not satisfiable in any CMAEHS. Of course, when a state is removed, so are all of its incoming and outgoing arrows.

There are two reasons why a state Δ of $\mathcal{T}_0^\theta$ might turn out to be unsatisfiable: either because Δ needs, in order to satisfy some diamond-formula, a successor state that has been eliminated, or because Δ contains an eventuality that is not realized in the tableau. Accordingly, we have two elimination rules **(E1)** and **(E2)**.

Formally, the state elimination phase is divided into stages; we start at stage 0 with $\mathcal{T}_0^\theta$; at stage $n+1$, we remove from the tableau $\mathcal{T}_n^\theta$ obtained at the previous stage exactly one state, by applying one of the elimination rules, thus obtaining the tableau $\mathcal{T}_{n+1}^\theta$. We state the rules below, where S_m^θ denotes the set of states of $\mathcal{T}_m^\theta$.

(E1) If $\Delta \in S_n^\theta$ contains a formula $\chi = \neg\mathbf{D}_A\varphi$ such that there is no $\Delta \xrightarrow{\chi} \Delta'$, where $\Delta' \in S_n^\theta$, then obtain $\mathcal{T}_{n+1}^\theta$ by eliminating Δ from $\mathcal{T}_n^\theta$.

For the other elimination rule, we need the concept of eventuality realization.

DEFINITION 4.8
The eventuality $\xi = \neg\mathbf{C}_A\varphi$ *is realized at* Δ *in* $\mathcal{T}_n^\theta$ if either $\neg\varphi \in \Delta$ or there exists in $\mathcal{T}_n^\theta$ a finite number of states $\Delta_0, \Delta_1, \ldots, \Delta_m$ such that $\Delta_0 = \Delta$; $\neg\varphi \in \Delta_m$; and, for every $0 \leq i < m$, $\xi \in \Delta_i$ and there exists $\chi_i = \neg\mathbf{D}_{a_i}\psi_i$ such that $a_i \in A$ and $\Delta_i \xrightarrow{\chi_i} \Delta_{i+1}$.

We can now state the rule.

(E2) If $\Delta \in S_n^\theta$ contains an eventuality $\neg\mathbf{C}_A\varphi$ that is not realized at Δ in $\mathcal{T}_n^\theta$, then obtain $\mathcal{T}_{n+1}^\theta$ by removing Δ from $\mathcal{T}_n^\theta$.

We check for realization of $\neg\mathbf{C}_A\varphi$ by running the following, *global* procedure that marks all states of $\mathcal{T}_n^\theta$ realizing $\neg\mathbf{C}_A\varphi$ in $\mathcal{T}_n^\theta$. Initially, we mark all $\Delta \in S_n^\theta$ such that $\neg\varphi \in \Delta$. Then, we repeatedly do the following: if $\Delta \in S_n^\theta$ contains $\neg\mathbf{C}_A\varphi$ and is unmarked yet, but there exists at least one Δ' such that $\Delta \xrightarrow{\neg\mathbf{D}_a\psi} \Delta'$, for some formula ψ and $a \in A$ and Δ' is marked, we mark Δ. The procedure is over when no more states get marked. Note that marking is carried out with respect to a fixed eventuality

ξ and is, therefore, repeated each time we want to check realization of an eventuality (see reasons further).

We have so far described elimination rules; to describe the state elimination phase as a whole, we need to specify the order of their application. We have to be careful since, having applied **(E2)**, we could have removed all the states accessible from some Δ along the arrows marked with some formula χ; hence, we need to reapply **(E1)** to the resultant tableau to remove such Δ's. Conversely, after having applied **(E1)**, we could have thrown away some states that were needed for realizing certain eventualities; hence, we need to reapply **(E2)**. Moreover, we cannot terminate the procedure unless we have checked that *all* eventualities are realized. Therefore, we apply **(E1)** and **(E2)** in a dovetailed sequence that cycles through all the eventualities. More precisely, we arrange all eventualities occurring in the states of $\mathcal{T}_0^\theta$ in a list $\xi_1, \ldots, \xi_m$. Then, we proceed in cycles. Each cycle consists of alternatingly applying **(E2)** to the pending eventuality (starting with ξ_1), and then applying **(E1)** to the resulting tableau, until all the eventualities have been dealt with. These cycles are repeated until no state is removed throughout a whole cycle. When that happens, the state elimination phase is over.

The graph produced at the end of the state elimination phase is called the *final tableau for* θ, denoted by $\mathcal{T}^\theta$, and its set of states is denoted by S^θ.

Definition 4.9
The final tableau $\mathcal{T}^\theta$ is *open* if $\theta \in \Delta$ for some $\Delta \in S^\theta$; otherwise, $\mathcal{T}^\theta$ is *closed*.

The tableau procedure returns 'no' (not satisfiable) if the final tableau is closed; otherwise, it returns 'yes' (satisfiable) and, moreover, provides sufficient information for producing a finite model satisfying θ; that construction is sketched in Section 5.2.

Example 4.10
We will continue to make the final tableau for the formulae Θ considered in Example 4.6 and Example 4.7. The state elimination procedure starts with the initial tableau given in Figure 4. During the state-elimination phase, state Δ_1 gets removed due to **(E1)**, since it does not have any successor states along an arrow labelled with χ, while states $\Delta_2, \Delta_3, \Delta_4$ and Δ_5 are eliminated due to **(E2)**, as all of them contain the unrealized eventuality $\neg\mathbf{C}_{\{a,b\}}p$. Thus, the final tableau for Θ is an empty graph; therefore, Θ is *unsatisfiable*.

5 Soundness and completeness of the tableau

5.1 Soundness

Technically, soundness of a tableau procedure amounts to claiming that if the input formula θ is satisfiable, then the final tableau $\mathcal{T}^\theta$ is open.

Before going into the technical details, we give an informal outline of the proof. The tableau procedure for the input formula θ starts off with creating a single prestate $\{\theta\}$. Then, we expand $\{\theta\}$ into states, each of which contains θ. To establish soundness, it suffices to show that at least one of these states survives to the end of the procedure and is, thus, part of the final tableau.

We start out by showing (Lemma 5.1) that if a prestate Γ is satisfiable, then at least one state created from Γ using **(SR)** is also satisfiable. In particular, this ensures that if θ is satisfiable, then so is at least one state obtained by **(SR)** from $\{\theta\}$. To ensure soundness, it suffices to prove that this state never gets eliminated from the tableau.

To that end, we first show (Lemma 5.2) that, given a satisfiable state Δ, all the prestates created from Δ in accordance with **(DR)**—each prestate being associated with a formula of the form $\neg\mathbf{D}_A\varphi$—are satisfiable; according to Lemma 5.1, each of these prestates will give rise to at least one satisfiable state. It follows that, if a tableau state Δ is satisfiable, then every successor of Δ in the initial tableau will have at least one satisfiable successor reachable by an arrow associated with each formula of the form $\neg\mathbf{D}_A\varphi$ belonging to Δ. Hence, if Δ is satisfiable, it will not be eliminated on account of **(E1)**.

Second, we show that no satisfiable states contain unrealized eventualities (in the sense of Definition 4.8), and thus cannot be removed from the tableau on account of **(E2)**. Thus, we show that a satisfiable state of the pretableau (equivalently, initial tableau) cannot be removed on account of any of the state elimination rules and, therefore, survives to the end of the procedure. In particular, this means that at least one state obtained from the initial prestate θ, and thus containing θ, survives to the end of the procedure. Hence, the final tableau for θ is open, as desired.

We emphasize again that the claims mentioned above, and their proofs, do not depend on the application (or not) of the cut rules CS1 and CS2, because they are sound, since $\gamma\vee\neg\gamma$ is valid for any formula γ. Therefore, these results are unaffected by the restrictive conditions C_1 and C_2 for their application.

We now proceed with the technical details.

Lemma 5.1
Let Γ be a prestate of $\mathcal{P}^\theta$ such that $\mathcal{M},s\Vdash\Gamma$ for some CMAEM $\mathcal{M}$ and $s\in\mathcal{M}$. Then:

1. $\mathcal{M},s\Vdash\Delta$ holds for at least one $\Delta\in\mathbf{states}(\Gamma)$.
2. Moreover, if $\neg\mathbf{C}_A\varphi\in\Gamma$ and $\mathcal{M},s\Vdash\neg\varphi$, then Δ can be chosen so that $\neg\varphi\in\Delta$.
3. If $\neg\mathbf{C}_A\varphi\in\Gamma$ while none of $\neg\mathbf{C}_A\varphi$'s β-components are in Γ, then for every $a\in A$, if $\mathcal{M},s\Vdash\neg\mathbf{D}_a\mathbf{C}_A\varphi$ then Δ can be chosen so that either $\neg\mathbf{D}_a\mathbf{C}_A\varphi\in\Delta$ or $\neg\varphi\in\Delta$.

Proof. Straightforward from the definition of $\mathcal{CSE}(\Gamma)$ and using Proposition 3.8. ■

Lemma 5.2
Let $\Delta\in S_0^\theta$ be such that $\mathcal{M},s\Vdash\Delta$ for some CMAEM $\mathcal{M}$ and $s\in\mathcal{M}$, and let $\neg\mathbf{D}_A\varphi\in\Delta$. Then, there exists $t\in\mathcal{M}$ such that $(s,t)\in\mathcal{R}_A^D$ and $\mathcal{M},t\Vdash\Gamma$, for a set Γ defined according to the rule **(DR)** applied to Δ and $\neg\mathbf{D}_A\varphi$:

$$\Gamma=\{\neg\varphi\}\cup\{\mathbf{D}_{A'}\psi\in\Delta\mid A'\subseteq A\}\cup\{\neg\mathbf{D}_{A'}\psi\in\Delta\mid A'\subseteq A\text{ and }\neg\mathbf{D}_{A'}\psi\neq\neg\mathbf{D}_A\varphi\}\cup\{\neg\mathbf{C}_{A'}\psi\in\Delta|A'\cap A\neq\emptyset\}$$

Proof. Easily follows from the semantics of the epistemic operators and the definition of CMAEM. ■

Lemma 5.3
Let $\Delta\in\mathcal{S}_0^\theta$, let $\neg\mathbf{C}_A\varphi,\neg\mathbf{D}_a\mathbf{C}_A\varphi\in\Delta$, and let, furthermore, $\Delta\overset{\neg\mathbf{D}_a\mathbf{C}_A\varphi}{\longrightarrow}\Gamma$ for some prestate $\Gamma\in\mathcal{P}^\theta$. Assume that $\mathcal{M},s\Vdash\Delta$ and $(s,s')\in\mathcal{R}_a^D$, for some model $\mathcal{M}$ and a pair of states $s,s'\in\mathcal{M}$; then $\mathcal{M},s'\Vdash\Gamma$.

Proof. Recall from the rule **(DR)** that $\Gamma=\{\neg\mathbf{C}_A\varphi\}\cup\{\mathbf{D}_a\gamma\mid\mathbf{D}_a\gamma\in\Delta\}\cup\{\neg\mathbf{D}_a\gamma\mid\neg\mathbf{D}_a\gamma\in\Delta,\neg\mathbf{D}_a\gamma\neq\neg\mathbf{D}_a\mathbf{C}_A\varphi\}\cup\{\neg\mathbf{C}_a\gamma\mid\neg\mathbf{C}_a\gamma\in\Delta\}$. The claim follows easily, because $\mathcal{R}_a^D$ is an equivalence relation. Indeed, $\mathcal{M},s'\Vdash\neg\mathbf{C}_A\varphi$ because every A-reachable state from s is A-reachable from s', too. Moreover, $(s',s'')\in\mathcal{R}_a^D$ iff $(s,s'')\in\mathcal{R}_a^D$, for all s''. Therefore, $\mathcal{M},s'\Vdash\chi$ for all $\chi\in\Gamma\setminus\{\neg\mathbf{C}_A\varphi\}$. ■

Lemma 5.4
Let $\Delta\in\mathcal{S}_0^\theta$ be such that $\mathcal{M},s\Vdash\Delta$ for some CMAEM $\mathcal{M}$ and $s\in\mathcal{M}$, and let $\neg\mathbf{C}_A\varphi\in\Delta$. Then there is a finite path in $\mathcal{S}_0^\theta$ of satisfiable states that realizes $\neg\mathbf{C}_A\varphi$ at Δ.

PROOF. We start by proving the following:

Let $\neg\mathbf{C}_A\varphi \in \Gamma_1$ for some prestate $\Gamma_1 \in \mathcal{P}^\theta$ such that Γ_1 does not contain any of the β-components of $\neg\mathbf{C}_A\varphi$. Suppose that $\mathcal{M}, s_1 \Vdash \Gamma_1$, and let $s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} \dots \xrightarrow{a_{n-1}} s_n$ be a *shortest* path in $\mathcal{M}$ that satisfies $\neg\mathbf{C}_A\varphi$, i.e. $\mathcal{M}, s_n \Vdash \neg\varphi$, and for all $i < n$, the following hold: $\mathcal{M}, s_i \Vdash \{\neg\mathbf{C}_A\varphi, \varphi\}$, and $(s_i, s_{i+1}) \in \mathcal{R}^D_{a_i}$, for some $a_i \in A$. Then there exists a path

$$\Delta_1 \xrightarrow{\neg\mathbf{D}_{a_1}\mathbf{C}_A\varphi} \Delta_2 \xrightarrow{\neg\mathbf{D}_{a_2}\mathbf{C}_A\varphi} \dots \xrightarrow{\neg\mathbf{D}_{a_{n'-1}}\mathbf{C}_A\varphi} \Delta_{n'},$$

of satisfiable states in $\mathcal{S}^\theta_0$, where $n' \le n$, $\Delta_1 \in \mathbf{states}(\Gamma_1)$ and $\neg\varphi \in \Delta_{n'}$.

We prove the above claim by induction on n.

If $n = 1$, then $\mathcal{M}, s_1 \Vdash \neg\varphi$. Since $\neg\mathbf{C}_A\varphi \in \Gamma_1$ and $\mathcal{M}, s_1 \Vdash \Gamma_1$, Lemma 5.1 implies that there is a $\Delta_1 \in \mathbf{states}(\Gamma_1)$ such that $\mathcal{M}, s_1 \Vdash \Delta_1$ and $\neg\varphi \in \Delta_1$. Thus Δ_1 is the needed path in $\mathcal{S}^\theta_0$ that satisfies the claim above.

Assume now the claim holds for all $m < n$. Let $\neg\mathbf{C}_A\varphi \in \Gamma_1$, let $\mathcal{M}, s_1 \Vdash \Gamma_1$, and assume that none of $\neg\mathbf{C}_A\varphi$'s β-components are in Γ_1. Let the path in $\mathcal{M}$ satisfying the eventuality $\neg\mathbf{C}_A\varphi$ be $s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} \dots \xrightarrow{a_{n-1}} s_n$, where $n > 1$.

Since $\mathcal{M}, s_1 \Vdash \{\neg\mathbf{D}_{a_1}\mathbf{C}_A\varphi, \neg\mathbf{C}_A\varphi\}$, Lemma 5.1 implies the existence of $\Delta_1 \in \mathbf{states}(\Gamma_1)$ in $\mathcal{S}^\theta_0$ with $\mathcal{M}, s_1 \Vdash \Delta_1$, and $\neg\mathbf{D}_{a_1}\mathbf{C}_A\varphi \in \Delta_1$ or $\neg\varphi \in \Delta_1$. In the latter case, Δ_1 is the needed path. In the former case, due to Lemma 5.2, there exists a prestate $\Gamma_2 \in \mathcal{T}^\theta$, with $\Delta_1 \xrightarrow{\neg\mathbf{D}_{a_1}\mathbf{C}_A\varphi} \Gamma_2$; then, $\neg\mathbf{C}_A\varphi \in \Gamma_2$. Note that Γ_2 cannot contain any of $\neg\mathbf{C}_A\varphi$'s β-components, since Δ_1 contains $\neg\mathbf{D}_{a_1}\mathbf{C}_A\varphi$, and thus, it can contain at most one other β-component, namely $\neg\varphi$. But in that case we would have that $\mathcal{M}, s_1 \Vdash \neg\varphi$, which contradicts the assumption. Lemma 5.3 gives us $\mathcal{M}, s_2 \Vdash \Gamma_2$.

Thus, since $s_2 \xrightarrow{a_2} \dots \xrightarrow{a_{n-1}} s_n$ is a path of length $n-1$ that realizes $\neg\mathbf{C}_A\varphi$ at s_2, the induction hypothesis claims that there is a path of satisfiable states in $\mathcal{S}^\theta_0$,

$$\Delta_2 \xrightarrow{\neg\mathbf{D}_{a_2}\mathbf{C}_A\varphi} \dots \xrightarrow{\neg\mathbf{D}_{a_{n'-1}}\mathbf{C}_A\varphi} \Delta_{n'},$$

where $n' \le n-1$, $\Delta_2 \in \mathbf{states}(\Gamma_2)$, $\neg\varphi \in \Delta_{n'}$.

Since $\Gamma_1 \dashrightarrow \Delta_1 \xrightarrow{\neg\mathbf{D}_{a_1}\mathbf{C}_A\varphi} \Gamma_2 \dashrightarrow \Delta_2$, we obtain a path in $\mathcal{S}^\theta_0$ of length atmost n that satisfies the induction hypothesis.

That concludes the induction.

Getting back to the claim of the Lemma, we have that if $\neg\mathbf{C}_A\varphi \in \Delta$, then either $\neg\varphi \in \Delta$ or there exists an $a' \in A$ such that $\neg\mathbf{D}_{a'}\mathbf{C}_A\varphi \in \Delta$, since Δ is fully expanded. In the former case, $\neg\mathbf{C}_A\varphi$ is realized in Δ and the claims follows. In the latter case, there will be a prestate Γ in $\mathcal{T}^\theta$, such that $\Delta \xrightarrow{\neg\mathbf{D}_{a'}\mathbf{C}_A\varphi} \Gamma$. Note that in this case $\Gamma \subseteq \Delta$. Due to **(DR)** and the fact that $\neg\varphi \notin \Delta$, Γ cannot contain any of $\neg\mathbf{C}_A\varphi$'s β-components.

Thus, the statement above gives us that there there is a $\Delta \to \Gamma \to \Delta_1 \to \dots \to \Delta_{n'}$, i.e. there is a path of satisfiable states in $\mathcal{S}^\theta_0$, that realizes $\neg\mathbf{C}_A\varphi \in \Delta$. ∎

THEOREM 5.5
If $\theta \in \mathcal{L}$ is satisfiable in a CMAEM, then $\mathcal{T}^\theta$ is open.

PROOF. Using the preceding Lemma 5.2 and Lemma 5.4, one can show by induction on the number of stages in the state elimination process that no satisfiable state can be eliminated due to **(E1)**–**(E2)**. The claim then follows from Lemma 5.1. ∎

5.2 Completeness

The *completeness* of a tableau procedure means that if the tableau for a formula θ is open, then θ is satisfiable in a CMAEM. In view of Theorem 3.15, it suffices to show that an open tableau for θ can be turned into a CMAEHS for θ. In order to prove that, we need to specify sufficiently strong restrictive conditions C_1 and C_2 governing the application of the cut rules CS1 and CS2, respectively on formulas $\mathbf{D}_A\varphi$ and $\mathbf{C}_A\varphi$ in the Definition 4.1 of cut-saturated sets. We now specify these conditions as follows.

C_1 Cut on $\mathbf{D}_A\varphi \in \mathsf{Sub}(\psi)$ where $\psi \in \Delta$, if either of the following holds:
- C_{11} $\psi = \mathbf{D}_B\delta$ or $\psi = \neg\mathbf{D}_B\delta$, and there is a $\neg\mathbf{D}_E\varepsilon \in \Delta$ such that $A \subseteq E$ and $B \subseteq E$.
- C_{12} $\psi = \neg\mathbf{C}_B\delta$ and there exists a $\neg\mathbf{D}_E\varepsilon \in \Delta$ such that $A \subseteq E$ and $B \cap E \neq \emptyset$.

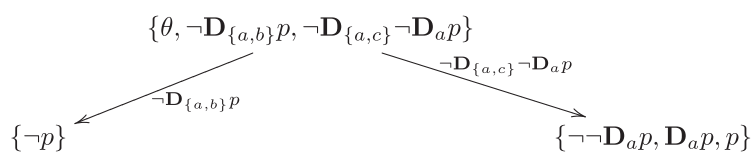
C_2 Cut on $\mathbf{C}_A\varphi \in \mathsf{Sub}(\psi)$ where $\psi \in \Delta$, if either of the following holds:
- C_{21} $\psi = \mathbf{D}_B\delta$ or $\psi = \neg\mathbf{D}_B\delta$, and there exists a $\neg\mathbf{D}_E\epsilon \in \Delta$ such that $B \subseteq E$ and $A \cap E \neq \emptyset$.
- C_{22} $\psi = \neg\mathbf{C}_B\delta$ and there exists a $\neg\mathbf{D}_E\varepsilon \in \Delta$ such that $A \cap E \neq \emptyset$ and $B \cap E \neq \emptyset$.

The intuition: a cut rule only has to be applied to a formula $\mathbf{D}_A\varphi$ or $\mathbf{C}_A\varphi$ if:

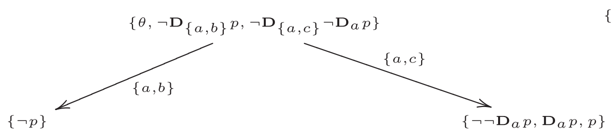
(i) that formula can occur in the label of a descendant state and,
(ii) once it occurs there, it will have an effect spreading back to the current state.

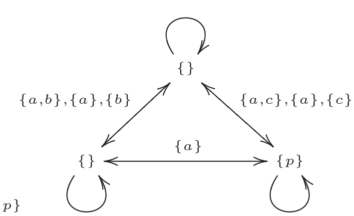
For the former to happen, that formula must occur in a $\mathbf{D}_B$-formula or a $\neg\mathbf{D}_B$-formula or a $\neg\mathbf{C}_B$-formula. For the latter, the path leading from the current state to that descendant must be labelled with relations propagating the effect of the respective box.

Example 5.6
This example illustrates the need for applying cut rules and using cut-saturated sets instead of simply fully expanded sets. First, recall the requirement of the relations in a (pseudo-)**CMAEL**(CD) model to be equivalence relations, reflected in (CH3) of Definition 3.9 for Hintikka structures. Now, consider the tableau constructed for the formula $\theta = \neg\mathbf{D}_{\{a,b\}}p \wedge \neg\mathbf{D}_{\{a,c\}}\neg\mathbf{D}_a p$ *if we would only use fully expanded sets*:



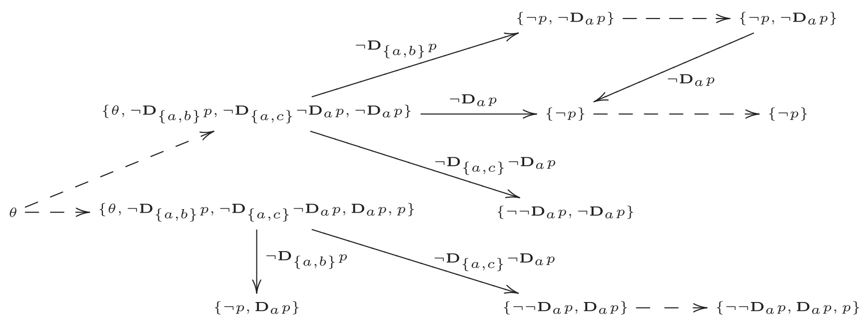
The corresponding claimed Hintikka structure and (pseudo)-model, that this tableau would produce (see the construction in Lemma 5.8) would then be, respectively:





In the 'Hintikka'-structure to the left, we have that $\mathbf{D}_a p$ is in the state in the bottom right corner, but not in the state in the top, though the edge connecting them is labelled with $\{a,c\}$. This on the other hand means, that θ is not satisfied in the 'model' to the right, because $\mathbf{D}_a p$ does not hold at any state, hence $\neg\mathbf{D}_{\{a,c\}}\neg\mathbf{D}_a p$ in not true at any state. In fact, θ is not satisfiable at all.

If we would indeed apply the cut-rules then the tableau for θ would close. The pretableau for θ would look as follows.



Notice that some of the prestates (namely $\{\neg p, \mathbf{D}_a p\}$ and $\{\neg\neg\mathbf{D}_a p, \neg\mathbf{D}_a p\}$) do not have any full expansions since these are patently inconsistent. After the initial tableau has been build, this then causes the two states in **states**(θ) to be deleted by **(E1)** and the final tableau is

$$\{\neg p, \neg\mathbf{D}_a p\} \xrightarrow{\neg\mathbf{D}_a p} \{\neg p\} \qquad\qquad \{\neg\neg\mathbf{D}_a p, \mathbf{D}_a p, p\}$$

which closes.

The following lemma is needed to ensure that the satisfaction of the condition (CH3) from the definition of Hintikka structures for **CMAEL**(CD) is guaranteed in the final tableau.

Lemma 5.7
Suppose $\Delta \xrightarrow{\neg\mathbf{D}_A\varphi} \Delta'$ in the final tableau $\mathcal{T}^\theta$ for some input formula θ and suppose that $\mathbf{D}_B\psi \in \Delta'$ where $B \subseteq A$. Then $\mathbf{D}_B\psi \in \Delta$.

Proof. First, note that if the cut rules CS1 and CS2 are applied unrestrictedly to every subformula $\mathbf{D}_A\varphi$ or $\mathbf{C}_A\varphi$ of a formula in the label of the current state Δ, the proof of the lemma is immediate. We will show that the claim still holds if the restrictions C_1 and C_2, specified above, are imposed.

For a formula α we let $\mathrm{CS}_1(\alpha)$ be the set of all formulae that can occur in any one-step cut-saturated expansion of α according to the procedure described in Definition 4.2. Similarly $\mathrm{CS}_1(\Gamma) = \bigcup_{\alpha\in\Gamma}\mathrm{CS}_1(\alpha)$ for a set Γ of formulae, and recursively we let $\mathrm{CS}_n(\Gamma) = \mathrm{CS}_1(\mathrm{CS}_{n-1}(\Gamma))$. As is easy to see, this construction converges, and the following is true:

- For any formula α and any $n \in \mathbf{N}$, $\mathrm{CS}_n(\alpha) \subseteq \mathsf{ecl}(\alpha)$, i.e.:

$$\mathrm{CS}_n(\alpha) \subseteq \{\beta, \neg\beta \mid \beta \in \mathsf{Sub}(\alpha)\} \cup \{\mathbf{D}_e\mathbf{C}_E\varepsilon, \neg\mathbf{D}_e\mathbf{C}_E\varepsilon \mid \mathbf{C}_E\varepsilon \in \mathsf{Sub}(\alpha), e \in E\}$$

- For any cut-saturated expansion Ω of Γ there is an n such that $\Omega \subseteq \mathrm{CS}_n(\Gamma)$.
- If $\beta \in \mathrm{CS}_n(\Gamma)$, then there is an $\alpha \in \Gamma$, such that $\beta \in \mathrm{CS}_n(\alpha)$.

Now, let Γ be the prestate in the pretableau $\mathcal{P}^\theta$ that gives rise to the relation between Δ and Δ', i.e. $\Delta \xrightarrow{\neg\mathbf{D}_A\varphi} \Gamma \dashrightarrow \Delta'$ in $\mathcal{P}^\theta$. The above gives us that since Δ' is a cut-saturated expansion of Γ and $\mathbf{D}_B\psi \in \Delta'$, there is an $\alpha \in \Gamma$ such that $\mathbf{D}_B\psi \in \mathsf{ecl}(\alpha)$. That is, either $\mathbf{D}_B\psi \in \mathsf{Sub}(\alpha)$, or $\mathbf{D}_B\psi = \mathbf{D}_d\mathbf{C}_D\delta$ for a $\mathbf{C}_D\delta \in \mathsf{Sub}(\alpha)$ and a $d \in D$.

Since $\alpha \in \Gamma$, due to **(DR)**, either $\alpha = \mathbf{D}_C\gamma \in \Delta$ or $\alpha = \neg\mathbf{D}_C\gamma \in \Delta$ for a $C \subseteq A$, or $\alpha = \neg\mathbf{C}_C\gamma \in \Delta$ where $C \cap A \neq \emptyset$, or $\alpha = \neg\varphi$. We notice that it is enough to show that C_1 is applicable to $\mathbf{D}_B\psi$ at Δ, since then either $\mathbf{D}_B\psi \in \Delta$ (which is what we want) or $\neg\mathbf{D}_B\psi \in \Delta$; the latter would, according to **(DR)**, imply that $\neg\mathbf{D}_B\psi \in \Gamma \subseteq \Delta'$, which would cause Δ' to be patently inconsistent, which contradicts Δ' being a cut-saturated set and thus fully expanded (cf. CS0). We split according to cases:

Case 1. $\alpha = \mathbf{D}_C\gamma \in \Delta$ or $\alpha = \neg\mathbf{D}_C\gamma \in \Delta$ for a $C \subseteq A$: $\mathbf{D}_B\psi \in \mathsf{ecl}(\mathbf{D}_C\gamma)$ gives that $\mathbf{D}_B\psi \in \mathsf{Sub}(\mathbf{D}_C\gamma)$, or $\mathbf{D}_B\psi = \mathbf{D}_d\mathbf{C}_D\delta$ for a $\mathbf{C}_D\delta \in \mathsf{Sub}(\mathbf{D}_C\gamma)$, where $d \in D$.

In the first case, $\mathbf{D}_B\psi$ is a subformula of a $\mathbf{D}_C$-formula in Δ, and since $C, B \subseteq A$ and $\neg\mathbf{D}_A\varphi \in \Delta$, C_1 is applicable to $\mathbf{D}_B\psi$ at Δ.

In the second case, $\mathbf{D}_B\psi = \mathbf{D}_d\mathbf{C}_D\delta$ for an $\mathbf{C}_D\delta \in \mathsf{Sub}(\mathbf{D}_C\gamma)$ with $d \in D$. Since $B = \{d\} \subseteq A$, we have $d \in D \cap A$ and hence C_2 is applicable to $\mathbf{C}_D\delta$ at Δ, as we also have $C \subseteq A$ and $\neg\mathbf{D}_A\varphi \in \Delta$. This means that either $\mathbf{C}_D\delta \in \Delta$ or $\neg\mathbf{C}_D\delta \in \Delta$ according to CS2. If $\mathbf{C}_D\delta \in \Delta$, then $\mathbf{D}_B\psi = \mathbf{D}_d\mathbf{C}_D\delta \in \Delta$ according to CS0. If $\neg\mathbf{C}_D\delta \in \Delta$, then according to **(DR)**, $\neg\mathbf{C}_D\delta \in \Gamma$ since $d \in D \cap A$. However, $\mathbf{D}_B\psi = \mathbf{D}_d\mathbf{C}_D\delta \in \Gamma$, and hence $\mathbf{C}_D\delta \in \Delta'$. This gives us a contradiction, as Δ' is fully expanded and, thus, not patently inconsistent.

The case where $\alpha = \neg\mathbf{D}_C\gamma$ is similar.

Case 2. $\alpha = \neg\varphi$: $\mathbf{D}_B\psi \in \mathsf{ecl}(\neg\varphi)$. We have two cases to consider:

Either $\mathbf{D}_B\psi \in \mathsf{Sub}(\neg\varphi)$, in which case $\mathbf{D}_B\psi \in \mathsf{Sub}(\varphi) \subseteq \mathsf{Sub}(\neg\mathbf{D}_A\varphi)$ and thus C_1 is applicable (since $B, A \subseteq A$).

$\mathbf{D}_B\psi = \mathbf{D}_d\mathbf{C}_D\delta$ for an $\mathbf{C}_D\delta \in \mathsf{Sub}(\neg\varphi)$ and $d \in D$ gives that $\mathbf{C}_D\delta \in \mathsf{Sub}(\varphi) \subseteq \mathsf{Sub}(\neg\mathbf{D}_A\varphi)$, and thus C_2 is applicable to $\mathbf{C}_D\delta$ at Δ since, again, $d \in D \cap A$ and $A \subseteq A$. Then, either $\mathbf{C}_D\delta \in \Delta$ or $\neg\mathbf{C}_D\delta \in \Delta$. As before, the former implies that $\mathbf{D}_B\psi \in \Delta$, as desired, while the latter leads to a contradiction.

Case 3. $\alpha = \neg\mathbf{C}_C\gamma$, where $C \cap A \neq \emptyset$:

$\mathbf{D}_B\psi \in \mathsf{Sub}(\neg\mathbf{C}_C\gamma)$ immediately gives that C_1 is applicable to $\mathbf{D}_B\psi$ at Δ.

If $\mathbf{D}_B\psi = \mathbf{D}_d\mathbf{C}_D\delta$, where $\mathbf{C}_D\delta \in \mathsf{Sub}(\neg\mathbf{C}_C\gamma)$ and $d \in D$, then C_2 is applicable to $\mathbf{C}_D\delta$ at Δ, as $d \in D \cap A$ and $\neg\mathbf{D}_A\varphi \in \Delta$. Thus, either $\mathbf{C}_D\delta \in \Delta$ or $\neg\mathbf{C}_D\delta$. The former implies that, due to CS0, $\mathbf{D}_B\psi \in \Delta$, as desired, while the other gives a contradiction, due to **(DR)** and CS0. ∎

Lemma 5.8
If $\mathcal{T}^\theta$ is open, then there exists a CMAEHS for θ.

Proof. The needed Hintikka structure $\mathcal{H}$ for the formula θ is built out of the final tableau $\mathcal{T}^\theta$ by renaming the relations between the states, such that they correspond to a subset of Σ. This is done by labelling the edges from Δ to Δ' with the set A for which $\Delta \stackrel{\neg\mathbf{D}_A\varphi}{\longrightarrow} \Delta'$ in $\mathcal{T}^\theta$.

Now, let Σ be the set of agents occurring in θ, and let $S = \mathcal{S}^\theta$. For any $A \in \mathcal{P}^+(\Sigma)$, let $\mathcal{R}^D_A = \{(\Delta, \Delta') \in S \times S \mid \Delta \stackrel{\neg\mathbf{D}_A\varphi}{\longrightarrow} \Delta' \text{ for some } \varphi\}$, and let $\mathcal{R}^C_A$ be the reflexive, transitive closure of $\bigcup_{B \subseteq A} \mathcal{R}^D_B$. Let $L(\Delta)$ be the labelling of the state in $\mathcal{T}$, i.e. the sets of formulae that has been associated with Δ.

Finally, let $\mathcal{H}_\theta = (\Sigma, \mathcal{S}^\theta, \{\mathcal{R}^D_A\}_{A \in \mathcal{P}^+(\Sigma)}, \{\mathcal{R}^C_A\}_{A \in \mathcal{P}^+(\Sigma)}, \mathbf{AP}, L)$.

We will now show that $\mathcal{H}_\theta$ is a Hintikka structure. To that end, we have to prove $(\Sigma, S, \{\mathcal{R}^D_A\}_{A \in \mathcal{P}^+(\Sigma)}, \{\mathcal{R}^C_A\}_{A \in \mathcal{P}^+(\Sigma)})$ is a CMAES, and that conditions (CH1)-(CH4) of Definition 3.9 hold for $\mathcal{H}$. The former is clear from the construction of $\mathcal{H}$.

(CH1) holds since all states in the final tableau are fully expanded.

(CH2) is satisfied since, otherwise, the state would have been deleted from the tableau due to **(E1)**. Likewise, (CH4) is satisfied since, otherwise, the state would have been removed due to **(E2)**.

It remains to show that (CH3) holds. Let $(\Delta, \Delta') \in \mathcal{R}^D_A$ (i.e. $\Delta \stackrel{\neg\mathbf{D}_A\varphi}{\longrightarrow} \Delta'$ in $\mathcal{T}^\theta$), and $B \subseteq A$. We need to show that $\mathbf{D}_B\psi \in \Delta \Leftrightarrow \mathbf{D}_B\psi \in \Delta'$. If $\mathbf{D}_B\psi \in \Delta$, then due to the propagation rule **(DR)**, $\mathbf{D}_B\psi \in \Gamma$, where Γ is the prestate in the final pretableau, such that $\Delta \xrightarrow{\neg \mathbf{D}_A\varphi} \Gamma \dashrightarrow \Delta'$. Thus $\mathbf{D}_B\psi$ is also in Δ' since Γ is included in all cut-saturated expansions of Γ. The other direction follows from Lemma 5.7 ∎

THEOREM 5.9 (Completeness)
Let $\theta \in \mathcal{L}$ and let $\mathcal{T}^\theta$ be open. Then, θ is satisfiable in a CMAEM.

PROOF. Immediate from Lemma 5.8 and Theorem 3.15. ∎

6 Complexity, efficiency and possible optimizations of the tableau procedure

6.1 Complexity

The termination of the tableau procedure described above is a fairly straightforward consequence of the finiteness of the set of all possible labels of states and prestates and their re-use in the construction phase. In this subsection, we estimate the worst-case running time of all phases of the procedure.

We denote by $|\theta|$ the length of a formula θ and by $|\mathsf{ecl}(\theta)|$ the number of formulae in $\mathsf{ecl}(\theta)$. Let $|\theta| = n$ and the number of agents occurring in θ be k.

By Lemma 4.5, $|\mathsf{ecl}(\theta)| \leq ckn$ for some (small) constant c. Then, the number of prestates and states in the tableau for θ is $\mathcal{O}(2^{ckn})$. Comparing two states or prestates takes $\mathcal{O}(ckn)$ steps (assuming a fixed order of the formulae in $\mathsf{ecl}(\theta)$, and each state being represented as a 0/1 string of length ckn), hence checking whether a prestate or a state has already been created, takes $\mathcal{O}(ckn2^{ckn})$. Therefore, the construction phase takes time $\mathcal{O}(ckn2^{2ckn})$.

The prestate elimination phase takes time $\mathcal{O}(2^{ckn})$. Checking realization of an eventuality in a state takes $\mathcal{O}(2^{ckn})$ steps and the number of eventualities is bounded by n, hence the elimination of a 'bad' state takes at most $\mathcal{O}(n2^{ckn})$ steps. Hence, the elimination state takes $\mathcal{O}(n2^{2ckn})$ steps.

We conclude that the whole tableau procedure terminates in $\mathcal{O}(ckn2^{2ckn})$ steps, hence it is in EXPTIME, which is in compliance with the known EXPTIME(-complete) lower bound (see [9], [10]).

6.2 Efficiency

Some features of the 'diamond-propagating' procedure described above make it sometimes practically sub-efficient.

Firstly, the application of the cut rules CS1 and CS2 can produce many cut-saturated sets, even after imposing the restrictive conditions C_1 and C_2. Potentially, it can create a number of states that is exponential in the number of subformulae of the form $\mathbf{D}_A\psi$ or $\mathbf{C}_A\psi$ occurring in the formulas of the input set Γ.

Secondly, when applying the rule **(DR)** to a state Δ with respect to some $\neg\mathbf{D}_A\varphi$, we propagate to the newly created prestate all the diamond-formulae of the form $\neg\mathbf{D}_B\psi$, where $B \subseteq A$, except $\neg\mathbf{D}_A\varphi$ itself. Likewise, all formulae $\neg\mathbf{C}_A\psi$ where A and B are not disjoint, get propagated. Thus, the presence of a 'diamond' in a prestate Γ is then passed on to all states in $\mathbf{states}(\Gamma)$, resulting in the need to apply the rule **(DR)** to every state in $\mathbf{states}(\Gamma)$ with respect to this diamond; this, again, implies the creation of a large number of states (even though, as we have shown, the maximal number of states is still no more than exponential in the size of the input formula). However, we re-iterate that this 'diamond-propagation' is necessary for the procedure developed here, because if a diamond-formula is not propagated forward, then its negation, which is a box-formula, may be added to a successor state and thus clash with that diamond-formula in the current state.

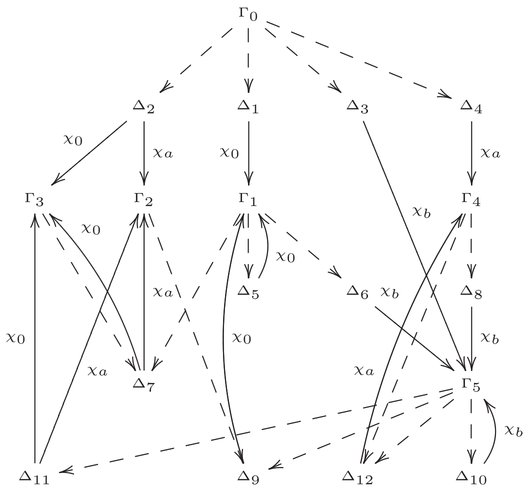
On the other hand, the restrictive conditions C_1 and C_2 for the application of cut-saturation in the production of CS-expansions can have a very significant effect on the size of the tableau, as illustrated by the next example.

EXAMPLE 6.1
Suppose we want to build a tableau for the formula $\theta = \mathbf{C}_{\{a,b\}}\mathbf{D}_a p \to \neg\mathbf{C}_{\{b,c\}}\mathbf{D}_b p \equiv \neg(\mathbf{C}_{\{a,b\}}\mathbf{D}_a p \wedge \mathbf{C}_{\{b,c\}}\mathbf{D}_b p)$ and suppose that $\Sigma = \{a,b,c\}$. We start off with creating a single prestate $\{\theta\}$. Using only the unrestricted conditions C_1 and C_2 to cut, applying the rule **(SR)** to this prestate produces an overwhelming number of 35 states:

1. $\{\theta, \neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \mathbf{D}_a p, p, \mathbf{C}_{\{b,c\}}\mathbf{D}_b p\}$;
2. $\{\theta, \neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \mathbf{D}_a p, p, \neg\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_b\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \mathbf{D}_b p\}$;
3. $\{\theta, \neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \mathbf{D}_a p, p, \neg\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_b\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_b p\}$;
4. $\{\theta, \neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \mathbf{D}_a p, p, \neg\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_c\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \mathbf{D}_b p\}$;
5. $\{\theta, \neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \mathbf{D}_a p, p, \neg\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_c\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_b p\}$;
6. $\{\theta, \neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \mathbf{D}_a p, p, \neg\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_b p\}$;
7. $\{\theta, \neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a p, \mathbf{C}_{\{b,c\}}\mathbf{D}_b p, p\}$;
8. $\{\theta, \neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a p, \neg\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_b\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \mathbf{D}_b p, p\}$;
9. $\{\theta, \neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a p, \neg\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_b\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_b p\}$;
10. $\{\theta, \neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a p, \neg\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_c\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \mathbf{D}_b p, p\}$;
11. $\{\theta, \neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a p, \neg\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_c\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_b p\}$;
12. $\{\theta, \neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a p, \neg\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_b p\}$;
13. $\{\theta, \neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_b\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \mathbf{D}_a p, p, \mathbf{C}_{\{b,c\}}\mathbf{D}_b p\}$;
14. $\{\theta, \neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_b\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \mathbf{D}_a p, p, \neg\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_b\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \mathbf{D}_b p\}$;
15. $\{\theta, \neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_b\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \mathbf{D}_a p, p, \neg\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_b\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_b p\}$;
16. $\{\theta, \neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_b\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \mathbf{D}_a p, p, \neg\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_c\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \mathbf{D}_b p\}$;
17. $\{\theta, \neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_b\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \mathbf{D}_a p, p, \neg\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_c\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_b p\}$;
18. $\{\theta, \neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_b\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \mathbf{D}_a p, p, \neg\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_b p\}$;
19. $\{\theta, \neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_b\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a p, \mathbf{C}_{\{b,c\}}\mathbf{D}_b p, p\}$;
20. $\{\theta, \neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_b\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a p, \neg\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_b\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \mathbf{D}_b p, p\}$;
21. $\{\theta, \neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_b\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a p, \neg\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_b\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_b p\}$;
22. $\{\theta, \neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_b\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a p, \neg\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_c\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \mathbf{D}_b p, p\}$;
23. $\{\theta, \neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_b\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a p, \neg\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_c\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_b p\}$;
24. $\{\theta, \neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_b\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a p, \neg\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_b p\}$;
25. $\{\theta, \neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a p, \mathbf{C}_{\{b,c\}}\mathbf{D}_b p, p\}$;
26. $\{\theta, \neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a p, \neg\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_b\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \mathbf{D}_b p, p\}$;
27. $\{\theta, \neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a p, \neg\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_b\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_b p\}$;
28. $\{\theta, \neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a p, \neg\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_c\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \mathbf{D}_b p, p\}$;
29. $\{\theta, \neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a p, \neg\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_c\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_b p\}$;
30. $\{\theta, \neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a p, \neg\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_b p\}$;
31. $\{\theta, \mathbf{C}_{\{a,b\}}\mathbf{D}_a p, p, \neg\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_b\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \mathbf{D}_b p\}$;
32. $\{\theta, \mathbf{C}_{\{a,b\}}\mathbf{D}_a p, p, \neg\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_b\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_b p\}$;
33. $\{\theta, \mathbf{C}_{\{a,b\}}\mathbf{D}_a p, p, \neg\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_c\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \mathbf{D}_b p\}$;
34. $\{\theta, \mathbf{C}_{\{a,b\}}\mathbf{D}_a p, p, \neg\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_c\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_b p\}$;
35. $\{\theta, \mathbf{C}_{\{a,b\}}\mathbf{D}_a p, p, \neg\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_b p\}$;

If we instead use the restricted C_1 and C_2, we will produce 8 states:

1. $\{\theta, \neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \mathbf{D}_a p, p\}$
2. $\{\theta, \neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a p\}$
3. $\{\theta, \neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_b\mathbf{C}_{\{a,b\}}\mathbf{D}_a p\}$
4. $\{\theta, \neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a p\}$
5. $\{\theta, \neg\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_b\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \mathbf{D}_b p, p\}$
6. $\{\theta, \neg\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_b\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_b p\}$
7. $\{\theta, \neg\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_c\mathbf{C}_{\{b,c\}}\mathbf{D}_b p\}$
8. $\{\theta, \neg\mathbf{C}_{\{b,c\}}\mathbf{D}_b p, \neg\mathbf{D}_b p\}$



$\chi_0 = \neg\mathbf{D}_a p,\ \chi_a = \neg\mathbf{D}_a\mathbf{C}_{\{a,b\}}\mathbf{D}_a p,\ \chi_b = \neg\mathbf{D}_b\mathbf{C}_{\{a,b\}}\mathbf{D}_a p$

$\Gamma_0 = \{\neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p\}$
$\Delta_1 = \{\neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a p\}$
$\Delta_2 = \{\neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a p\}$
$\Delta_3 = \{\neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_b\mathbf{C}_{\{a,b\}}\mathbf{D}_a p\}$
$\Delta_4 = \{\neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \mathbf{D}_a p, p\}$
$\Gamma_1 = \{\neg p, \mathbf{D}_a\neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p\}$
$\Gamma_2 = \{\neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a p\}$
$\Gamma_3 = \{\neg p, \mathbf{D}_a\neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a\mathbf{C}_{\{a,b\}}\mathbf{D}_a p\}$
$\Gamma_4 = \{\neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \mathbf{D}_a p\}$
$\Delta_5 = \{\neg p, \neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a p\}$
$\Delta_6 = \{\neg p, \neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_b\mathbf{C}_{\{a,b\}}\mathbf{D}_a p\}$
$\Delta_7 = \{\neg p, \neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a p\}$
$\Delta_8 = \{\neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \mathbf{D}_a p, p, \neg\mathbf{D}_b\mathbf{C}_{\{a,b\}}\mathbf{D}_a p\}$
$\Gamma_5 = \{\neg\mathbf{C}_{\{ab\}}\mathbf{D}_a p\}$
$\Delta_9 = \{\neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a p\}$
$\Delta_{10} = \{\neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_b\mathbf{C}_{\{a,b\}}\mathbf{D}_a p\}$
$\Delta_{11} = \{\neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a p\}$
$\Delta_{12} = \{\neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \neg\mathbf{D}_a\mathbf{C}_{\{a,b\}}\mathbf{D}_a p, \mathbf{D}_a p, p\}$

FIG. 5. Pretableau for $\neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p$.

Figure 5 shows the pretableau for one part of θ, i.e. $\neg\mathbf{C}_{\{a,b\}}\mathbf{D}_a p$. The tableau for the other part of θ will be similar and disjoint from this tableau.

As seen here, the backtracking procedure is rather inefficient when applied to formulae of the type of $\mathbf{C}_{\{a,b\}}\mathbf{D}_a p \rightarrow \neg\mathbf{C}_{\{b,c\}}\mathbf{D}_b p$. ■

Both causes of potential inefficiencies discussed above, viz. the forward diamond-propagation and the (restricted) analytic cut rules on box-formulae, are needed to ensure that every ‘successful’ tableau can be turned into a Hintikka structure. More precisely, they together ensure that the right-to-left implication in the statement of property (CH3) of Hintikka structures (recall Definition 3.9) holds.

A possible way of eliminating these causes for inefficiencies is to change the strategy in the tableau-building, by implementing a mechanism for *backward propagation of boxes*: if $\mathbf{D}_A\varphi$ occurs in a state Δ, then ensure that this box is propagated backwards to all predecessor states where it must occur. The main disadvantage of this approach is that it requires an elaborated mechanism of repeated updating the hitherto constructed part of the tableau. We leave the realization of this idea for future work.

6.3 Improvements

As stated earlier, the main emphasize of our tableau construction is the ease of presentation, comprehension and implementation, rather than technical sophistication and optimality of the procedure. While being worst-case time optimal, it is amenable to various improvements and further optimizations, some of which we will mention briefly here.

To begin with, for methodological reasons, our procedure is divided into three phases, where the different components of the tableau-building procedure are dealt with separately. That separation of the procedure into phases makes it less optimal compared to the approach whereby the three phases are carried out simultaneously and the prestate and state elimination is done 'on-the-fly'.

Also, as briefly mentioned in Section 4.1, it is possible to make the procedure cut-free by using a mechanism for 'backwards propagation' of **D**-formulas, which, when well designed can lead to more optimal performance in some cases. This approach is taken e.g. in [22], where the authors construct a cut-free tableau-based algorithm for the logic PDL with converse, while the algorithm presented in [29] builds on this work by constructing a cut-free tableaux-based algorithm for the description logic SHI, which contains inverse roles. Both methods account for the case where a (number of) formula(s) turns up in a node, which will be required to be in the already created predecessor node of the node in question. The former algorithm deals with eventualities, too. Adopting this approach to our procedure while optimizing it for the logic **CMAEL(CD)** would result in a procedure sketched below.

6.3.1 State elimination 'on-the-fly'

Here we make use of the concept of 'potential rescuers' used in [22] and [29], though in a slightly different way, adjusted to our needs. We likewise take on board the techniques of updating and propagating statuses of nodes in the tableaux.

Firstly, we maintain a status for (pre)states, which can either be `unexplored`, `open` or `closed`. The status of a (pre)state is initially set to `unexplored` when the (pre)state is created, and then updated during the procedure. When a prestate is expanded or a state expanded for all diamond-formulas in it, its status changes to `open`. Later on the status of a state Δ can then change to `closed` in the following cases:

- there is an epistemic prestate Γ such that $\Delta \xrightarrow{\delta} \Gamma$ for a formula δ and the status of Γ is `closed`.
- Δ contains an eventuality $\neg\mathbf{C}_A\varphi$ that it neither realized in the current tableau under construction nor has a 'potential rescuer'. A potential rescuer is a (pre)state, which is A-reachable from Δ, contains $\neg\mathbf{C}_A\varphi$, and has not been expanded yet, i.e. it has status `unexplored`. Here we use a modified definition of A-reachability, where $\dashrightarrow$-arrows are allowed too.

The status of a prestate Γ is set to `closed` if:

- all states in **states**(Γ) are closed, including the case where **states**$(\Gamma)=\emptyset$, or
- Γ contains an eventuality that it neither realized nor has a potential rescuer.

Additionally, we make sure, that unsatisfiable (pre)states are removed on-the-fly and that the procedure stops and the tableau closes as soon as unsatisfiability of the input prestate is detected during the procedure, i.e.:

- We close a prestate when it is expanded and does not have any cut-saturated expansions.
- When a (pre)state Σ closes, we propagate updates of statuses to the relevant (pre)states, whose status depend on the status of Σ. These are (pre)states that have outgoing arrows pointing to Σ.
- We keep an eye on the initial prestate, labelled with the input formula whose satisfiability we are checking. When/if this prestate closes, we stop the whole procedure and return 'unsat'.

Finally, we also want to avoid the unnecessary checking of unrealized eventualities, since this step is one of the more expensive checks. Thus, when updating the status of a (pre)state we only check containment of unrealized eventualities, when this is really necessary. E.g. we do not check that if a potential rescuer is known to be reachable. This of course requires some bookkeeping.

6.3.2 Making the procedure cut-free

The procedure above takes care of doing the satisfiability checking 'on the fly', however it is not cut-free. Though, the procedure can be made cut-free by incorporating the following:

Firstly, we use full expansions instead of cut-saturated expansions. Secondly, we now need to account for a further reason why a state Δ can close, namely that Δ contains a diamond formula $\neg\mathbf{D}_A\varphi$, such that $\Delta \overset{\neg\mathbf{D}_A\varphi}{\longrightarrow} \Gamma$ and all states in $\mathbf{states}(\Gamma)$ are *incompatible* with Δ with respect to $\neg\mathbf{D}_A\varphi$. Here, $\Delta' \in \mathbf{states}(\Gamma)$ is incompatible with Δ if $\{\mathbf{D}_{A'}\psi \in \Delta' \mid A' \subseteq A\} \nsubseteq \Delta$, i.e. condition (CH3) will not be fulfilled in the resulting Hintikka structure. This, however, does not neccessarily mean, that the state Δ needs to close. After all, since we are not proactively looking ahead for box-formulas which could possibly occur in a future descendent state of Δ and include these in Δ (as is done when using cut-saturated expansions), it is possible that Δ could become satisfiable if the box-formulas in question were added to Δ.

Therefore, when it happens that $\Delta \overset{\neg\mathbf{D}_A\varphi}{\longrightarrow} \Gamma$ and none of the states in $\mathbf{states}(\Gamma)$ are compatible with Δ with respect to $\neg\mathbf{D}_A\varphi$, we construct so-called 'alternatives' for the state Δ. These are states labelled with the fully expanded sets $\Delta \cup S'$ for each $S' \in \bigcup_{\Delta' \in \mathbf{states}(\Gamma)} \mathcal{FE}(\{\mathbf{D}_{A'}\psi \in \Delta' \mid A' \subseteq A \text{ and } \mathbf{D}_{A'}\psi \notin \Delta\})$. Then $\dashrightarrow$-arrows pointing to these alternatives are added from each prestates pointing to Δ, and finally we close the original state Δ (and propagate the change of status that hereby occurs, as described previously).

In this procedure, we need to keep track of when such incompatibilities occur, which requires some further bookkeeping.

7 Concluding remarks

We have developed a sound and complete tableau-based decision procedure for the full coalitional multiagent epistemic logic **CMAEL(CD)**. The incremental tableau style adopted here is intuitive, practically more efficient, and more flexible than the maximal tableau style, developed e.g. for the fragment **MAEL(C)** of **CMAEL(CD)** in [25], and therefore it is more suitable both for manual and automated execution. In fact, an earlier, less optimal version, of this procedure has been implemented and reported in [42]. On the other hand, as discussed in the previous section, various further optimizations of the procedure are possible and desirable, and some such optimizations have been developed for logics related to **CMAEL(CD)**, see Section 1.2. Furthermore, our tableau procedure is also amenable to various extensions, subject to current and future work:

- to temporal epistemic logics of linear and branching time, preliminary reports on which have appeared respectively in [15] and [16].
- with the strategic abilities operators of the Alternating-time temporal logic **ATL**, a tableau-based decision procedure for which were developed in [17]. Merging tableaux for these two logical systems will produce, inter alia, a feasible decision procedure for the Alternating-time temporal epistemic logic **ATEL** [40].
- a cut-free, 'on the fly' version, as described in Section 6.3.

Acknowledgments

We gratefully acknowledge the financial support from: the FIRST Research School and Roskilde University, funding the PhD study of the first author, part of which is her contribution to the present research; the National Research Foundation of South Africa for the second author through several research grants, and the Claude Harris Leon Foundation, funding the third author's post-doctoral fellowship at the University of the Witwatersrand, during which the initial work of this research was done. We also thank the anonymous referees for their valuable comments and constructive criticism.

References

[1] P. Abate, R. Gorè, and F. Widmann. Cut-free single-pass tableaux for the logic of common knowledge. In *Proceedings of TABLEAUX 2007 Workshop on Agents and Deduction*, 2007.

[2] R. Alur, T. A. Henzinger, and O. Kuperman. Alternating-time temporal logic. *Journal of the ACM*, **49**, 672–713, 2002.

[3] M. Ben-Ari, A. Pnueli, and Z. Manna. The temporal logic of branching time. *Acta Informatica*, **20**, 207–226, 1983.

[4] K. Brünnler and M. Lange. Cut-free sequent systems for temporal logic. *Journal of Logic and Algebraic Programming*, **76**, 216–225, 2008.

[5] B. Dunin-Keplicz, L. Anh Nguyen, and A. Szalas. Converse-PDL with regular inclusion axioms: a framework for MAS logics. *Journal of Applied Non-Classical Logics*, **21**, 61–91, 2011.

[6] M. Dziubinski, R. Verbrugge, and B. Dunin-Keplicz. Complexity issues in multiagent logics. *Fundamenta Infomaticae*, **75**, 239–262, 2007.

[7] E. Allen Emerson. Temporal and modal logics. In *Handbook of Theoretical Computer Science*, vol. B, J. van Leeuwen, ed., pp. 995–1072. MIT Press, 1990.

[8] E. Allen Emerson and J. Halpern. Decision procedures and expressiveness in the temporal logic of branching time. *Journal of Computer and System Sciences*, **30**, 1–24, 1985.

[9] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning about Knowledge*. MIT Press, 1995.

[10] R. Fagin, J. Y. Halpern, and M. Y. Vardi. What can machines know? On the properties of knowledge in distributed systems. *Journal of the ACM*, **39**, 328–376, 1992.

[11] M. C. Fitting. *Proof Methods for Modal and Intuitionistic Logic*. D. Reidel Publ., 1983.

[12] M. C. Fitting. Modal proof theory. In *Handbook of Modal Logic*, P. Blackburn, J. van Bentem, and F. Wolter, eds, pp. 85–138. Elsevier, 2007.

[13] V. Goranko and W. Jamroga. Comparing semantics of logics for multi-agent systems. *Synthese*, **139**, 241–280, 2004.

[14] V. Goranko and D. Shkatov. Tableau-based decision procedure for the multi-agent epistemic logic with operators of common and distributed knowledge. In *Proceedings of the sixth IEEE Conference on Sofware Engineering and Formal Methods (SEFM 2008)*, A. Cerone and S. Gunter, eds, pp. 237–246. IEEE Computer Society Press, 2008. Corrected version available at http://arxiv.org/abs/0808.4133.

[15] V. Goranko and D. Shkatov. Tableau-based decision procedure for full coalitional multiagent temporal-epistemic logic of linear time. In *Proceedings of AAMAS'2009*, Decker, Sichman, Sierra, and Castelfranchi, eds, pp. 969–976. IFAAMAS publications, 2009.

[16] V. Goranko and D. Shkatov. Tableau-based decision procedure for the full coalitional multiagent logic of branching time. In *MALLOW'2009*, M. Baldoni *et al.* eds. 2009.

[17] V. Goranko and D. Shkatov. Tableau-based decision procedures for logics of strategic ability in multiagent systems. *ACM Transactions on Computational Logic*, **11**, 1–48, 2009.

[18] V. Goranko and D. Shkatov. Tableau-based procedure for deciding satisfiability in the full coalitional multiagent epistemic logic. In *Proceedings of the Symposium on Logical Foundations of Computer Science (LFCS 2009)*, Vol. 5407 of *LNCS*, S. Artemov and A. Nerode, eds, pp. 197–213. Springer-Verlag, 2009.

[19] R. Goré. Tableau methods for modal and temporal logics. In *Handbook of Tableau Methods*, M. D'Agostino, D. Gabbay, R. Hänhle, and J. Posegga, eds, pp. 297–396. Kluwer, 1988.

[20] R. Goré and L. A. Nguyen. Analytic cut-free tableaux for regular modal logics of agent beliefs. In *CLIMA*, pp. 268–287, 2007.

[21] R. Goré and L. A. Nguyen. EXPTIME tableaux with global caching for description logics with transitive roles, inverse roles and role hierarchies. In *TABLEAUX'2007*, pp. 133–148, 2007.

[22] R. Goré and F. Widmann. Optimal and cut-free tableaux for propositional dynamic logic with converse. In *IJCAR 2010*, Vol. 6173 of *LNCS*, J. Giesl and R. Hähnle, eds, pp. 225–239. Springer, 2010.

[23] J. Y. Halpern. Using reasoning about knowledge to analyze distributed systems. *Annual Review of Computer Science*, **2**, 37–68, 1987.

[24] J. Y. Halpern and Y. Moses. Knowledge and common knowledge in a distributed environment. *Journal of ACM*, **37**, 549–587, 1990.

[25] J. Y. Halpern and Y. Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, **54**, 319–379, 1992.

[26] J. Jäger, M. Kretz, and T. Studer. Cut-free common knowledge. *Journal of Applied Logic*, **5**, 681–689, 2007.

[27] J.-J. Ch. Meyer and W. van der Hoek. *Epistemic Logic for Computer Science and Artificial Intelligence*. CUP, 1995.

[28] L. A. Nguyen. Sequent-like tableau systems with the analytic superformula property for the modal logics KB, KBD, K5, KD5*. In *Proceedings of TABLEAUX'2000*, Vol. 1847 of *LNCS*, R. Dyckhoff, ed., pp. 341–351. Springer, 2000.

[29] L. A. Nguyen. A cut-free exptime tableau decision procedure for the description logic SHI. In *ICCCI'2011*, Vol. 6922 of *LNCS*, P. Jedrzejowicz, N. T. Nguyen, and K. Hoang, eds, pp. 572–581. Springer, 2011.

[30] M. Pauly. *Logic for Social Software*. PhD Thesis, University of Amsterdam, 2001. ILLC Dissertation Series 2001-10.

[31] M. Pauly. A logical framework for coalitional effectivity in dynamic procedures. *Bulletin of Economic Research*, **53**, 305–324, 2001.

[32] M. Pauly. A modal logic for coalitional power in games. *Journal of Logic and Computation*, **12**, 149–166, 2002.

[33] V. R. Pratt. A practical decision method for propositional dynamic logic. In *Proceedings of the 10th Annual ACM Symposium on the Theory of Computing*, pp. 326–227, San Diego, California, May 1979.

[34] V. R. Pratt. A near optimal method for reasoning about action. *Journal of Computer and System Sciences*, **20**, 231–254, 1980.

[35] S. Schwendimann. A new one-pass tableau calculus for PLTL. In *Proceedings of TABLEAUX'98*, Vol. 1397 of *Lecture Notes in Artificial Intelligence*, H. de Swart, ed., pp. 277–291. Springer-Verlag, 1998.

[36] Y. Shoham and K. Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic and Logical Foundations*. CUP, 2008.

[37] R. M. Smullyan. *First-order Logic*. Springer-Verlag, 1968.
[38] W. van der Hoek and J.-J. Ch. Meyer. Making some issues of implicit knowledge explicit. *International Journal of Foundations of Computer Science*, **3**, 193–224, 1992.
[39] W. van der Hoek and J.-J. Ch. Meyer. A complete epistemic logic for multiple agents–combining distributed and common knowledge. In *Epistemic Logic and the Theory of Games and Decisions*, M. O. L. Bacharach *et al*., eds, pp. 35–68. Kluwer Academic Publishers, 1997.
[40] W. van der Hoek and M. Wooldridge. Cooperation, knowledge, and time: alternating-time temporal epistemic logic and its applications. *Studia Logica*, **75**, 125–157, 2004.
[41] M. Vardi. Automata-theoretic techniques for temporal reasoning. In *Handbook of Modal Logic*, P. Blackburn, J. van Benthem, and F. Wolter, eds, pp. 971–990. Elsevier, 2007.
[42] T. Vestergaard. Algorithmic design and implementation of tableau-based decision procedure for multi-agent epistemic logics. Master's thesis, Technical University of Denmark, Master thesis IMM-MSC-2010-64, 2010.
[43] D. Walther. ATEL with common and distributed knowledge is ExpTime-complete. In *Proceedings of Methods for Modalities 4*, Berlin, 2005.
[44] D. Walther, C. Lutz, F. Wolter, and M. Wooldridge. ATL satisfiability is indeed ExpTime-complete. *Journal of Logic and Computation*, **16**, 765–787, 2006.
[45] G. Weiss, ed. *Multiagent Systems*. MIT Press, 1999.
[46] P. Wolper. The tableau method for temporal logic: an overview. *Logique et Analyse*, **28**, 119–136, 1985.
[47] M. Wooldridge. *An Introduction to Multiagent Systems*. John Willey and Sons, 2002.

Received 18 February 2011