A resolution-based calculus for Coalition Logic

CLÁUDIA NALON, *Departament of Computer Science, University of Brasília, C. P. 4466, CEP:70.910-090, Brasília, DF, Brazil.*
E-mail: nalon@unb.br

LAN ZHANG, *Information School, Capital University of Economics and Business, Beijing 100070, China.*
E-mail: lan@cueb.edu.cn

CLARE DIXON and ULLRICH HUSTADT, *Department of Computer Science, University of Liverpool, Liverpool, L69 3BX, UK.*
E-mail: CLDixon@liverpool.ac.uk; U.Hustadt@liverpool.ac.uk

Abstract

We present a resolution-based calculus for Coalition Logic CL, a non-normal modal logic used for reasoning about cooperative agency. We introduce a normal form and a set of inference rules to solve the satisfiability problem in CL. We also show that the calculus presented here is sound, complete, and terminating.

Keywords: Coalition logic, theorem-proving, resolution method.

1 Introduction

Coalition Logic CL was introduced in [16] as a logic for reasoning about cooperative agency, that is, a formalism intended to describe the ability of groups of agents to achieve an outcome in a strategic game. CL has been used for verification of properties of voting procedures [16] and reasoning about strategic games [17].

CL is a multi-modal logic with modal operators of the form $[\mathcal{A}]$, where $\mathcal{A}$ is a set of agents. The formula $[\mathcal{A}]\varphi$, where $\mathcal{A}$ is a set of agents and φ is a formula, reads as *the coalition of agents $\mathcal{A}$ has the ability of bringing about φ* or *the coalition of agents $\mathcal{A}$ has a strategy to achieve φ*. We note that if a set of agents has a strategy for achieving φ and a strategy for achieving ψ, then this does not mean that in general they have a strategy for achieving $\varphi \wedge \psi$. Thus, CL is a non-normal modal logic, that is, the schema that represents the *additivity principle*, $[\mathcal{A}]\varphi \wedge [\mathcal{A}]\psi \Rightarrow [\mathcal{A}](\varphi \wedge \psi)$, is not valid. However, the *monotonicity principle*, given by $[\mathcal{A}](\varphi \wedge \psi) \Rightarrow [\mathcal{A}]\varphi \wedge [\mathcal{A}]\psi$, holds.

Coalition Logic is closely related to *Alternating-Time Temporal Logic*, ATL, given in [1, 2] and revisited in [3]. In fact, CL is equivalent to the next-time fragment of ATL [8], where $[\mathcal{A}]\varphi$ translates into $\langle\langle \mathcal{A} \rangle\rangle \bigcirc \varphi$ (read as *the coalition $\mathcal{A}$ can ensure φ at the next moment in time*). The satisfiability problems for ATL and CL are EXPTIME-complete [20] and PSPACE-complete [17], respectively.

Vol. 24 No. 4, Published by Oxford University Press.

Published online January 21, 2014 doi:10.1093/logcom/ext074

Methods for tackling the satisfiability problem for these logics include, for instance, two tableau-based methods for ATL [9, 20], two automata-based methods [6, 10] for ATL, and one tableau-based method for CL [12]. As to the best of our knowledge, no resolution-based method has been developed for either ATL or CL. Providing a resolution method for CL gives the user a choice of proof methods. Several comparisons of tableau algorithms and resolution methods [11, 13] indicate that there is no overall best approach: for some classes of formulae tableau algorithms perform better whilst on others resolution performs better. So, with a choice of different provers, for the best result the user could run several in parallel or the one most likely to succeed depending on the type of the input formulae.

In this article, we present a resolution-based calculus for CL, $\mathsf{RES}_{\mathsf{CL}}$. The method can be seen as a (syntactic) variation of the resolution calculus for the next-time fragment of ATL introduced in [22], where soundness and termination proofs are given, but where the completeness proof is omitted. We provide the full correctness results here. The completeness proof for $\mathsf{RES}_{\mathsf{CL}}$ is given relative to the tableau calculus in [9]. If a formula is unsatisfiable, the corresponding tableau is closed. We show that deletions that produce the closed tableau correspond to applications of the resolution inference rules given by the method presented here. Establishing the completeness result with respect to the tableau procedure simplifies the proofs. For CL, we could have chosen to prove completeness relatively to the simpler tableau-based method given in [12]. However, the tableau-based method in [9] has a formulation that is closer to that of the resolution method presented here, that is, it works with one-sided sequents whilst the tableau-based method in [12] works with two-sided sequents. We also note that, although [12] presents a method for ATL, neither soundness nor completeness proofs are presented. As it is our intention to extend the method presented here to full ATL, the same technique can be used later, in a modular way, to provide correctness results for a resolution-based calculus for ATL.

This article is organized as follows. In the next section, we present the syntax, axiomatization and semantics of CL. In Section 3, we introduce the resolution-based method for CL and provide a few examples. Correctness results are given in Section 4. Conclusions and future work are given in Section 5. An extended version of this article can be found in [15].

2 Coalition logic

In the following we present the syntax, axiomatization and semantics of CL.

2.1 *Syntax*

As in [9], we define $\Sigma \subset \mathbb{N}$ to be a finite, non-empty set of agents. A **coalition** $\mathcal{A}$ is a subset of Σ. Formulae in CL are constructed from propositional symbols and constants, together with Boolean operators and coalition modalities. A **coalition modality** is either of the form $[\mathcal{A}]\varphi$ or $\langle\mathcal{A}\rangle\varphi$, where φ is a well-formed CL formula. The coalition operator $\langle\mathcal{A}\rangle$ is the dual of $[\mathcal{A}]$, where $\mathcal{A}$ is a coalition, that is, $\langle\mathcal{A}\rangle\varphi$ is an abbreviation for $\neg[\mathcal{A}]\neg\varphi$, for every formula φ.

Definition 2.1
The set of CL well-formed formulae, $\mathsf{WFF}_{\mathsf{CL}}$, is given by:

- constants: $\{\mathbf{true}, \mathbf{false}\}$;
- propositional symbols: $\Pi = \{p, q, r, \ldots, p_1, q_1, r_1, \ldots\}$;

– classical formulae: if $\varphi, \psi \in \mathsf{WFF}_{\mathsf{CL}}$, then so are $\neg\varphi$ (negation), $(\varphi \wedge \psi)$ (conjunction), $(\varphi \vee \psi)$ (disjunction), and $(\varphi \Rightarrow \psi)$ (implication);
– coalition formulae: if $\varphi \in \mathsf{WFF}_{\mathsf{CL}}$, then so are $[\mathcal{A}]\varphi$ and $\langle\mathcal{A}\rangle\varphi$, where $\mathcal{A} \subseteq \Sigma$.

Parentheses will be omitted if the reading is not ambiguous. By convenience, formulae of the form $\bigvee \varphi_i$ (resp. $\bigwedge \varphi_i$), $1 \leq i \leq n$, $n \in \mathbb{N}$, $\varphi_i \in \mathsf{WFF}_{\mathsf{CL}}$, represent arbitrary disjunction (resp. conjunction) of formulae. If $n=0$, $\bigvee \varphi_i$ is called the **empty disjunction**, denoted by **false**, while $\bigwedge \varphi_i$ is called the **empty conjunction** denoted by **true**. Also, when enumerating a specific set of agents, we often omit the curly brackets. For example, we write $[1,2]\varphi$ as an abbreviation for $[\{1,2\}]\varphi$, for a formula φ. In the following, we use 'formula(e)' and 'well-formed formula(e)' interchangeably.

DEFINITION 2.2
Let Π be the set of propositional symbols. A **literal** is either p or $\neg p$, for $p \in \Pi$. For a literal l of the form $\neg p$, where p is a propositional symbol, $\neg l$ denotes p; for a literal l of the form p, $\neg l$ denotes $\neg p$. The literals l and $\neg l$ are called **complementary literals**.

Let $\varphi \in \mathsf{WFF}_{\mathsf{CL}}$, Σ the set of all agents, and $\mathcal{A} \subseteq \Sigma$. As in [9], a **positive coalition formula** (resp. **negative coalition formula**) is a formula of the form $[\mathcal{A}]\varphi$ (resp. $\langle\mathcal{A}\rangle\varphi$). A **coalition formula** is either a positive or a negative coalition formula.

2.2 Axiomatization

As presented in [17], coalition logic can be axiomatised by the following schemata (where $\mathcal{A}, \mathcal{A}'$ are coalitions and $\varphi, \varphi_1, \varphi_2$ are well-formed formulae):

$$
\begin{array}{rl}
\perp & : \neg[\mathcal{A}]\textbf{false} \\
\top & : [\mathcal{A}]\textbf{true} \\
\Sigma & : \neg[\emptyset]\neg\varphi \Rightarrow [\Sigma]\varphi \\
\mathbf{M} & : [\mathcal{A}](\varphi_1 \wedge \varphi_2) \Rightarrow [\mathcal{A}]\varphi_1 \\
\mathbf{S} & : [\mathcal{A}]\varphi_1 \wedge [\mathcal{A}']\varphi_2 \Rightarrow [\mathcal{A} \cup \mathcal{A}'](\varphi_1 \wedge \varphi_2), \text{ if } \mathcal{A} \cap \mathcal{A}' = \emptyset
\end{array}
$$

together with propositional tautologies and the following inference rules: **modus ponens** (from φ_1 and $\varphi_1 \Rightarrow \varphi_2$ infer φ_2) and **equivalence** (from $\varphi_1 \Leftrightarrow \varphi_2$ infer $[\mathcal{A}]\varphi_1 \Leftrightarrow [\mathcal{A}]\varphi_2$). It can be shown that the inference rule **monotonicity** (from $\varphi_1 \Rightarrow \varphi_2$ infer $[\mathcal{A}]\varphi_1 \Rightarrow [\mathcal{A}]\varphi_2$) is a derivable rule in this system.

EXAMPLE 2.3
We show that the formula

$$[\mathcal{A}]\psi_1 \wedge \langle\mathcal{B}\rangle\psi_2 \Rightarrow \langle\mathcal{B}\setminus\mathcal{A}\rangle(\psi_1 \wedge \psi_2)$$

where $\mathcal{A}$ and $\mathcal{B}$ are coalitions, $\mathcal{A} \subseteq \mathcal{B}$, and $\psi_1, \psi_2 \in \mathsf{WFF}_{\mathsf{CL}}$, is valid:

1.	$[\mathcal{A}]\psi_1 \wedge [\mathcal{B}\setminus\mathcal{A}](\psi_1 \Rightarrow \neg\psi_2) \Rightarrow [\mathcal{B}](\psi_1 \wedge (\psi_1 \Rightarrow \neg\psi_2))$	$\mathbf{S}, \mathcal{A}' = \mathcal{B}\setminus\mathcal{A}, \varphi_1 = \psi_1, \varphi_2 = \psi_1 \Rightarrow \neg\psi_2$
2.	$\psi_1 \wedge (\psi_1 \Rightarrow \neg\psi_2) \Rightarrow \neg\psi_2$	*propositional tautology*
3.	$[\mathcal{B}](\psi_1 \wedge (\psi_1 \Rightarrow \neg\psi_2)) \Rightarrow [\mathcal{B}]\neg\psi_2$	2, *monotonicity*
4.	$[\mathcal{A}]\psi_1 \wedge [\mathcal{B}\setminus\mathcal{A}](\psi_1 \Rightarrow \neg\psi_2) \Rightarrow [\mathcal{B}]\neg\psi_2$	1,3, *chaining*
5.	$[\mathcal{A}]\psi_1 \wedge \neg[\mathcal{B}]\neg\psi_2 \Rightarrow \neg[\mathcal{B}\setminus\mathcal{A}](\neg\psi_1 \vee \neg\psi_2)$	4, *rewriting*
6.	$[\mathcal{A}]\psi_1 \wedge \langle\mathcal{B}\rangle\neg\neg\psi_2 \Rightarrow \langle\mathcal{B}\setminus\mathcal{A}\rangle\neg(\neg\psi_1 \vee \neg\psi_2)$	5, *def. dual*
7.	$[\mathcal{A}]\psi_1 \wedge \langle\mathcal{B}\rangle\psi_2 \Rightarrow \langle\mathcal{B}\setminus\mathcal{A}\rangle(\psi_1 \wedge \psi_2)$	6, *rewriting*

2.3 Semantics

Semantics of CL is usually given in terms of *Multiplayer Game Models* (MGMs) [16]. However, we follow the presentation in [3, 9], which uses *Concurrent Game Structures* (CGSs) for describing the semantics of ATL. MGMs yield the same set of validities as CGSs [8].

The semantics of CL is *positional*, that is, agents have no memory of their past decisions and, thus, those decisions are made by taking into account only the current state. Also, the semantics given here is based on pointed-models, as we are interested in the structures together with a distinguished state where the valuation takes place. Restricting the models to pointed ones does not change the class of validities and it is useful in the proofs later presented in this work; for further discussion about pointed-models, see, for instance, [4].

DEFINITION 2.4
A **Concurrent Game Frame** (CGF) is a tuple $\mathcal{F}=(\Sigma,\mathcal{S},s_0,d,\delta)$, where

- Σ is a finite non-empty set of **agents**;
- $\mathcal{S}$ is a non-empty set of **states**, with a distinguished state s_0;
- $d:\Sigma\times\mathcal{S}\longrightarrow\mathbb{N}^+$, where the natural number $d(a,s)\geq 1$ represents the **number of moves** that the agent a has at the state s. Every **move** for agent a at the state s is identified by a number between 0 and $d(a,s)-1$. Let $D(a,s)=\{0,\ldots,d(a,s)-1\}$ be the set of all moves available to agent a at s. For a state s, a **move vector** is a k-tuple $(\sigma_1,\ldots,\sigma_k)$, where $k=|\Sigma|$, such that $0\leq\sigma_a\leq d(a,s)-1$, for all $a\in\Sigma$. Intuitively, σ_a represents an arbitrary move of agent a in s. Let $D(s)=\Pi_{a\in\Sigma}D(a,s)$ be the set of all move vectors at s. We denote by σ an arbitrary member of $D(s)$.
- δ is a **transition function** that assigns to every $s\in\mathcal{S}$ and every $\sigma\in D(s)$ a state $\delta(s,\sigma)\in\mathcal{S}$ that results from s if every agent $a\in\Sigma$ plays move σ_a.

In the following, let $\mathcal{F}=(\Sigma,\mathcal{S},s_0,d,\delta)$ be a CGF with $s,s'\in\mathcal{S}$. We say that s' is a **successor** of s (an s-successor) if $s'=\delta(s,\sigma)$, for some $\sigma\in D(s)$.

Let κ be a tuple. We write κ_n (or $\kappa(n)$) to refer to the n-th element of κ.

DEFINITION 2.5
Let $|\Sigma|=k$ and let $\mathcal{A}\subseteq\Sigma$ be a coalition. An $\mathcal{A}$**-move** $\sigma_{\mathcal{A}}$ at $s\in\mathcal{S}$ is a k-tuple such that $\sigma_{\mathcal{A}}(a)\in D(a,s)$ for every $a\in\mathcal{A}$ and $\sigma_{\mathcal{A}}(a')=*$ (i.e. an arbitrary move) for every $a'\notin\mathcal{A}$. We denote by $D(\mathcal{A},s)$ the set of all $\mathcal{A}$-moves at state s.

DEFINITION 2.6
A move vector σ **extends** an $\mathcal{A}$-move vector $\sigma_{\mathcal{A}}$, denoted by $\sigma_{\mathcal{A}}\sqsubseteq\sigma$ or $\sigma\sqsupseteq\sigma_{\mathcal{A}}$, if $\sigma(a)=\sigma_{\mathcal{A}}(a)$ for every $a\in\mathcal{A}$.

Given a coalition $\mathcal{A}\subseteq\Sigma$, an $\mathcal{A}$-move $\sigma_{\mathcal{A}}\in D(\mathcal{A},s)$, and a $\Sigma\setminus\mathcal{A}$-move $\sigma_{\Sigma\setminus\mathcal{A}}\in D(\Sigma\setminus\mathcal{A},s)$, we denote by $\sigma_{\mathcal{A}}\sqcup\sigma_{\Sigma\setminus\mathcal{A}}$ the unique $\sigma\in D(s)$ such that both $\sigma_{\mathcal{A}}\sqsubseteq\sigma$ and $\sigma_{\Sigma\setminus\mathcal{A}}\sqsubseteq\sigma$.

DEFINITION 2.7
Let $\sigma_{\mathcal{A}}\in D(\mathcal{A},s)$ be an $\mathcal{A}$-move. The **outcome** of $\sigma_{\mathcal{A}}$ at s, denoted by $out(s,\sigma_{\mathcal{A}})$, is the set of all states $s'\in\mathcal{S}$ for which there exists a move vector $\sigma\in D(s)$ such that $\sigma_{\mathcal{A}}\sqsubseteq\sigma$ and $\delta(s,\sigma)=s'$.

DEFINITION 2.8
A **Concurrent Game Model** (CGM) is a tuple $\mathcal{M}=(\mathcal{F},\Pi,\pi)$, where $\mathcal{F}=(\Sigma,\mathcal{S},s_0,d,\delta)$ is a CGF; Π is the set of propositional symbols; and $\pi:\mathcal{S}\longrightarrow 2^{\Pi}$ is a valuation function.

DEFINITION 2.9
Let $\mathcal{M} = (\Sigma, \mathcal{S}, s_0, d, \delta, \Pi, \pi)$ be a CGM with $s \in \mathcal{S}$. The satisfaction relation, denoted by $\models$, is inductively defined as follows.

- $\langle \mathcal{M}, s \rangle \models$ **true**;
- $\langle \mathcal{M}, s \rangle \models p$ iff $p \in \pi(s)$, for all $p \in \Pi$;
- $\langle \mathcal{M}, s \rangle \models \neg\varphi$ iff $\langle \mathcal{M}, s \rangle \not\models \varphi$;
- $\langle \mathcal{M}, s \rangle \models \varphi \wedge \psi$ iff $\langle \mathcal{M}, s \rangle \models \varphi$ and $\langle \mathcal{M}, s \rangle \models \psi$;
- $\langle \mathcal{M}, s \rangle \models [\mathcal{A}]\varphi$ iff there exists a $\mathcal{A}$-move $\sigma_{\mathcal{A}} \in D(\mathcal{A}, s)$ s.t. $\langle \mathcal{M}, s' \rangle \models \varphi$ for all $s' \in out(s, \sigma_{\mathcal{A}})$;
- $\langle \mathcal{M}, s \rangle \models \langle \mathcal{A} \rangle\varphi$ iff for all $\mathcal{A}$-moves $\sigma_{\mathcal{A}} \in D(\mathcal{A}, s)$ exists $s' \in out(s, \sigma_{\mathcal{A}})$ s.t. $\langle \mathcal{M}, s' \rangle \models \varphi$.

Semantics of **false**, disjunctions and implications are given in the usual way. Given a model $\mathcal{M}$, a state s in $\mathcal{M}$, and a formula φ, if $\langle \mathcal{M}, s \rangle \models \varphi$, $s \in \mathcal{S}$, we say that φ is **satisfied at the state** s **in** $\mathcal{M}$. Satisfiability of a formula in a model is defined next.

As discussed in [9, 16, 20] three different notions of satisfiability emerge from the relation between the set of agents occurring in a formula and the set of agents in the language. It turns out that all those notions of satisfiability can be reduced to *tight satisfiability*, that is, when the evaluation of a formula takes into consideration only the agents occurring in such formula [20]. In this work, we will consider this particular notion of satisfiability. We denote by Σ_φ, where $\Sigma_\varphi \subseteq \Sigma$, the set of agents occurring in a well-formed formula φ. If Φ is a set of well-formed formulae, $\Sigma_\Phi \subseteq \Sigma$ denotes $\bigcup_{\varphi \in \Phi} \Sigma_\varphi$. Let $\varphi \in \mathsf{WFF}_{\mathsf{CL}}$ and $\mathcal{M} = (\Sigma_\varphi, \mathcal{S}, s_0, d, \delta, \Pi, \pi)$ be a CGM. Formulae are interpreted with respect to the distinguished world s_0. Thus, a formula φ is said to be **satisfiable in** $\mathcal{M}$, denoted by $\mathcal{M} \models \varphi$, if $\langle \mathcal{M}, s_0 \rangle \models \varphi$; it is said to be **satisfiable** if there is a model $\mathcal{M}$ such that $\langle \mathcal{M}, s_0 \rangle \models \varphi$; and it is said to be **valid** if for all models $\mathcal{M}$ we have $\langle \mathcal{M}, s_0 \rangle \models \varphi$. A finite set $\Gamma \subset \mathsf{WFF}_{\mathsf{CL}}$ is **satisfiable in a state** s **in** $\mathcal{M}$, denoted by $\langle \mathcal{M}, s \rangle \models \Gamma$, if for all $\gamma_i \in \Gamma$, $0 \leq i \leq n$, $\langle \mathcal{M}, s \rangle \models \gamma_i$; Γ is **satisfiable in a model** $\mathcal{M}$, $\mathcal{M} \models \Gamma$, if $\langle \mathcal{M}, s_0 \rangle \models \Gamma$; and Γ is **satisfiable**, if there is a model $\mathcal{M}$ such that $\mathcal{M} \models \Gamma$.

3 Resolution calculus

The resolution calculus for CL, denoted by $\mathsf{RES}_{\mathsf{CL}}$, is based on that given in [22]. A formula to be tested for (un)satisfiability is translated into a *coalition problem* in divided separated normal form which, roughly speaking, separates the different contexts (formulae which are true only at the initial state; formulae which are true in all states without coalition operators, and formulae which are true in all states that include coalition operators) to which a set of resolution-based inference rules are applied. We present the normal form in the next section and the inference rules are given in Section 3.2. Examples are given within those sections.

3.1 Normal form

The resolution-based calculus for CL, $\mathsf{RES}_{\mathsf{CL}}$, operates on sets of clauses. A formula in CL is firstly converted into a coalition problem, which is then transformed into a coalition problem in *Divided Separated Normal Form for Coalition Logic*, $\mathsf{DSNF}_{\mathsf{CL}}$.

DEFINITION 3.1
A **coalition problem** is a tuple $(\mathcal{I}, \mathcal{U}, \mathcal{N})$, where $\mathcal{I}$, the set of initial formulae, is a finite set of propositional formulae; $\mathcal{U}$, the set of global formulae, is a finite set of formulae in $\mathsf{WFF}_{\mathsf{CL}}$; and $\mathcal{N}$, the set of coalition formulae, is a finite set of coalition formulae, i.e. those formulae in which a coalition modality occurs.

The semantics of coalition problems assumes that initial formulae hold at the initial state; and that global and coalition formulae hold at every state of a model. Formally, the semantics of coalition problems is defined as follows.

DEFINITION 3.2
Given a coalition problem $\mathcal{C}=(\mathcal{I},\mathcal{U},\mathcal{N})$, we denote by $\Sigma_{\mathcal{C}}$ the set of agents $\Sigma_{\mathcal{U}\cup\mathcal{N}}$. If $\mathcal{C}=(\mathcal{I},\mathcal{U},\mathcal{N})$ is a coalition problem and $\mathcal{M}=(\Sigma_{\mathcal{C}},\mathcal{S},s_0,d,\delta,\Pi,\pi)$ is a CGM, then $\mathcal{M}\models\mathcal{C}$ if, and only if, $\langle\mathcal{M},s_0\rangle\models\mathcal{I}$ and $\langle\mathcal{M},s\rangle\models\mathcal{U}\cup\mathcal{N}$, for all $s\in\mathcal{S}$. We say that $\mathcal{C}=(\mathcal{I},\mathcal{U},\mathcal{N})$ is **satisfiable**, if there is a model $\mathcal{M}$ such that $\mathcal{M}\models\mathcal{C}$.

In order to apply the resolution method, we further require that formulae within each of those sets are in *clausal forms*. These categories of clauses have the following syntactic form:

initial clauses	$\bigvee_{j=1}^{n} l_j$
global clauses	$\bigvee_{j=1}^{n} l_j$
positive coalition clauses	$\bigwedge_{i=1}^{m} l'_i \Rightarrow [\mathcal{A}]\bigvee_{j=1}^{n} l_j$
negative coalition clauses	$\bigwedge_{i=1}^{m} l'_i \Rightarrow \langle\mathcal{A}\rangle\bigvee_{j=1}^{n} l_j$

where $m,n\geq 0$ and l'_i,l_j, for all $1\leq i\leq m$, $1\leq j\leq n$, are literals or constants. Clauses are kept in the simplest form: literals in conjunctions and disjunctions are always pairwise different; constants **true** and **false** are removed from conjunctions and disjunctions with more than one conjunct/disjunct, respectively; conjunctions (resp. disjunctions) with either complementary literals or **false** (resp. **true**) are simplified to **false** (resp. **true**). Also, the tautologies **true**, **false**$\Rightarrow\varphi$, and $\varphi\Rightarrow$**true** are removed from the sets of clauses.

DEFINITION 3.3
A **coalition problem in** $\mathsf{DSNF}_{\mathsf{CL}}$ is a coalition problem $(\mathcal{I},\mathcal{U},\mathcal{N})$ such that $\mathcal{I}$ is a set of initial clauses, $\mathcal{U}$ is a set of global clauses, and $\mathcal{N}$ is a set of positive and negative coalition clauses.

Transformation rules: the transformation of a coalition logic formula into a coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$ is analogous to the approach taken in [5], where first-order temporal formulae are transformed into a *Divided Separated Normal Form* (DSNF), by means of renaming [18] and rewriting of temporal operators by simulating their fix-point representation. The transformation reduces the number of operators and separates the contexts to which the resolution inference rules are applied.

The transformation into the normal form used here is given by a set of rewrite rules. Let $\varphi\in\mathsf{WFF}_{\mathsf{CL}}$ be a formula and $\tau_0(\varphi)$ be the transformation of φ into the Negation Normal Form (NNF), that is, the formula obtained from φ by pushing negation inwards, so that negation symbols occur only next to propositional symbols. The transformation into NNF uses the following rewrite rules:

$$\begin{array}{rcl@{\qquad\qquad}rcl}
\varphi\Rightarrow\psi & \longrightarrow & \neg\varphi\vee\psi & \neg\neg\varphi & \longrightarrow & \varphi\\
\neg(\varphi\wedge\psi) & \longrightarrow & \neg\varphi\vee\neg\psi & \neg[\mathcal{A}]\varphi & \longrightarrow & \langle\mathcal{A}\rangle\neg\varphi\\
\neg(\varphi\vee\psi) & \longrightarrow & \neg\varphi\wedge\neg\psi & \neg\langle\mathcal{A}\rangle\varphi & \longrightarrow & [\mathcal{A}]\neg\varphi\\
\neg(\varphi\Rightarrow\psi) & \longrightarrow & \varphi\wedge\neg\psi & & &
\end{array}$$

In addition, we want to remove occurrences of the constants **true** and **false** as well as duplicates of formulae in conjunctions and disjunctions. This is achieved by exhaustively applying the following simplification rules (where conjunctions and disjunctions are commutative):

$$\begin{array}{rcl@{\qquad}rcl@{\qquad}rcl}
\varphi\wedge\mathbf{true} & \longrightarrow & \varphi & \neg\mathbf{false} & \longrightarrow & \mathbf{true} & [\mathcal{A}]\mathbf{true} & \longrightarrow & \mathbf{true}\\
\varphi\vee\mathbf{true} & \longrightarrow & \mathbf{true} & \varphi\vee\varphi & \longrightarrow & \varphi & [\mathcal{A}]\mathbf{false} & \longrightarrow & \mathbf{false}\\
\varphi\wedge\mathbf{false} & \longrightarrow & \mathbf{false} & \varphi\wedge\varphi & \longrightarrow & \varphi & \langle\mathcal{A}\rangle\mathbf{true} & \longrightarrow & \mathbf{true}\\
\varphi\vee\mathbf{false} & \longrightarrow & \varphi & \varphi\vee\neg\varphi & \longrightarrow & \mathbf{true} & \langle\mathcal{A}\rangle\mathbf{false} & \longrightarrow & \mathbf{false}\\
\neg\mathbf{true} & \longrightarrow & \mathbf{false} & \varphi\wedge\neg\varphi & \longrightarrow & \mathbf{false} & & &
\end{array}$$

Given a formula φ, we start its transformation into a coalition problem $(\mathcal{I},\mathcal{U},\mathcal{N})$ in $\mathsf{DSNF_{CL}}$ by exhaustively applying the rewriting rules given below, together with simplification, to the tuple $(\{t_0\},\{t_0\Rightarrow\tau_0(\varphi)\},\{\})$, where t_0 is a new propositional symbol and $\tau_0(\varphi)$ is the transformation of φ into NNF. For classical operators, we have the following rewriting rules (where t is a literal; φ_1 and φ_2 are formulae; t_1 is a new propositional symbol; and disjunctions are commutative):

$$\begin{array}{lrcl}
\tau_\wedge & (\mathcal{I},\mathcal{U}\cup\{t\Rightarrow\varphi_1\wedge\varphi_2\},\mathcal{N}) & \longrightarrow & (\mathcal{I},\mathcal{U}\cup\{t\Rightarrow\varphi_1,t\Rightarrow\varphi_2\},\mathcal{N})\\
\tau_\vee & (\mathcal{I},\mathcal{U}\cup\{t\Rightarrow\varphi_1\vee\varphi_2\},\mathcal{N}) & \longrightarrow & (\mathcal{I},\mathcal{U}\cup\{t\Rightarrow\varphi_1\vee t_1,t_1\Rightarrow\varphi_2\},\mathcal{N})\\
 & & & \text{where } \varphi_2 \text{ is not a disjunction of literals}\\
\tau_\Rightarrow & (\mathcal{I},\mathcal{U}\cup\{t\Rightarrow D\},\mathcal{N}) & \longrightarrow & (\mathcal{I},\mathcal{U}\cup\{\neg t\vee D\},\mathcal{N})\\
 & & & \text{where } D \text{ is either a constant or a disjunction of literals}\\
 & (\mathcal{I},\mathcal{U}\cup\{t\Rightarrow D\},\mathcal{N}) & \longrightarrow & (\mathcal{I},\mathcal{U},\mathcal{N}\cup\{t\Rightarrow D\})\\
 & & & \text{where } D \text{ is either of the form } [\mathcal{A}]\varphi_1 \text{ or } \langle\mathcal{A}\rangle\varphi_1
\end{array}$$

Note that, as disjunction is commutative, the rewriting rule $\tau_\vee$ also applies to $(\mathcal{I},\mathcal{U}\cup\{t\Rightarrow\varphi_2\vee\varphi_1\},\mathcal{N})$, where φ_1 is not a disjunction. The rules for renaming complex formulae in the scope of coalition modalities are given below, where $\mathcal{A}$ is a coalition and Σ_φ is the set of agents occurring in the original formula φ.

$$\begin{array}{lrcl}
\tau_{[\mathcal{A}]} & (\mathcal{I},\mathcal{U},\mathcal{N}\cup\{t\Rightarrow[\mathcal{A}]\varphi_1\}) & \longrightarrow & (\mathcal{I},\mathcal{U}\cup\{t_1\Rightarrow\varphi_1\},\mathcal{N}\cup\{t\Rightarrow[\mathcal{A}]t_1\})\\
 & & & \text{where } \varphi \text{ is not a disjunction of literals}\\
\tau_{\langle\mathcal{A}\rangle,\,\mathcal{A}\neq\Sigma_\varphi} & (\mathcal{I},\mathcal{U},\mathcal{N}\cup\{t\Rightarrow\langle\mathcal{A}\rangle\varphi_1\}) & \longrightarrow & (\mathcal{I},\mathcal{U}\cup\{t_1\Rightarrow\varphi_1\},\mathcal{N}\cup\{t\Rightarrow\langle\mathcal{A}\rangle t_1\})\\
 & & & \text{where } \varphi \text{ is not a disjunction of literals}\\
 & & & \text{and } \mathcal{A}\neq\Sigma_\varphi\\
\tau_{\langle\Sigma_\varphi\rangle} & (\mathcal{I},\mathcal{U},\mathcal{N}\cup\{t\Rightarrow\langle\Sigma_\varphi\rangle\varphi_1\}) & \longrightarrow & (\mathcal{I},\mathcal{U},\mathcal{N}\cup\{t\Rightarrow[\emptyset]\varphi_1\})
\end{array}$$

The transformation is linear in the size of the original formula [22]. We now show an example of an application of the transformation rules.

EXAMPLE 3.4
Consider the formula $\neg([1](p\wedge q)\wedge[1](q\wedge r)\Rightarrow[1]p\wedge[1]q\wedge[1]r)$, whose transformation into NNF is $\varphi = [1](p\wedge q)\wedge[1](q\wedge r)\wedge(\langle 1\rangle\neg p\vee\langle 1\rangle\neg q\vee\langle 1\rangle\neg r)$. The transformation into $\mathsf{DSNF_{CL}}$ starts from $\langle\{t_0\},\{t_0\Rightarrow\varphi\},\{\}\rangle$, and proceeds as follows:

1.	t_0	$[\mathcal{I}]$	5.	$t_0\Rightarrow[1](p\wedge q)$	$[\mathcal{U},\tau_\wedge,3]$
2.	$t_0\Rightarrow\varphi$	$[\mathcal{U}]$	6.	$t_0\Rightarrow[1](q\wedge r)$	$[\mathcal{U},\tau_\wedge,3]$
3.	$t_0\Rightarrow[1](p\wedge q)\wedge[1](q\wedge r)$	$[\mathcal{U},\tau_\wedge,2]$	7.	$t_0\Rightarrow[1]t_1$	$[\mathcal{N},\tau_{[\mathcal{A}]},5]$
4.	$t_0\Rightarrow\langle 1\rangle\neg p\vee\langle 1\rangle\neg q\vee\langle 1\rangle\neg r$	$[\mathcal{U},\tau_\wedge,2]$	8.	$t_1\Rightarrow(p\wedge q)$	$[\mathcal{U},\tau_{[\mathcal{A}]},5]$

9.	$t_0 \Rightarrow [1]t_2$	$[\mathcal{N}, \tau_{[\mathcal{A}]}, 6]$
10.	$t_2 \Rightarrow (q \wedge r)$	$[\mathcal{U}, \tau_{[\mathcal{A}]}, 6]$
11.	$t_0 \Rightarrow t_3 \vee \langle 1\rangle\neg q \vee \langle 1\rangle\neg r$	$[\mathcal{U}, \tau_{\vee}, 4]$
12.	$t_3 \Rightarrow \langle 1\rangle\neg p$	$[\mathcal{U}, \tau_{\vee}, 4]$
13.	$t_3 \Rightarrow \langle 1\rangle\neg p$	$[\mathcal{N}, \tau_{\Rightarrow}, 12]$
14.	$t_0 \Rightarrow t_3 \vee t_4 \vee \langle 1\rangle\neg r$	$[\mathcal{U}, \tau_{\vee}, 11]$
15.	$t_4 \Rightarrow \langle 1\rangle\neg q$	$[\mathcal{U}, \tau_{\vee}, 11]$
16.	$t_4 \Rightarrow \langle 1\rangle\neg q$	$[\mathcal{N}, \tau_{\Rightarrow}, 15]$
17.	$t_0 \Rightarrow t_3 \vee t_4 \vee t_5$	$[\mathcal{U}, \tau_{\vee}, 14]$
18.	$t_5 \Rightarrow \langle 1\rangle\neg r$	$[\mathcal{U}, \tau_{\vee}, 14]$
19.	$t_5 \Rightarrow \langle 1\rangle\neg r$	$[\mathcal{N}, \tau_{\Rightarrow}, 18]$
20.	$t_3 \Rightarrow [\emptyset]\neg p$	$[\mathcal{N}, \tau_{\langle\Sigma_\varphi\rangle}, 13]$
21.	$t_3 \Rightarrow [\emptyset]\neg q$	$[\mathcal{N}, \tau_{\langle\Sigma_\varphi\rangle}, 16]$
22.	$t_3 \Rightarrow [\emptyset]\neg r$	$[\mathcal{N}, \tau_{\langle\Sigma_\varphi\rangle}, 19]$
23.	$t_1 \Rightarrow p$	$[\mathcal{U}, \tau_{\wedge}, 8]$
24.	$t_1 \Rightarrow q$	$[\mathcal{U}, \tau_{\wedge}, 8]$
25.	$t_2 \Rightarrow q$	$[\mathcal{U}, \tau_{\wedge}, 10]$
26.	$t_2 \Rightarrow r$	$[\mathcal{U}, \tau_{\wedge}, 10]$
27.	$\neg t_0 \vee t_3 \vee t_4 \vee t_5$	$[\mathcal{U}, \tau_{\Rightarrow}, 17]$
28.	$\neg t_1 \vee p$	$[\mathcal{U}, \tau_{\Rightarrow}, 23]$
29.	$\neg t_1 \vee q$	$[\mathcal{U}, \tau_{\Rightarrow}, 24]$
30.	$\neg t_2 \vee q$	$[\mathcal{U}, \tau_{\Rightarrow}, 25]$
31.	$\neg t_2 \vee r$	$[\mathcal{U}, \tau_{\Rightarrow}, 26]$

The transformation results in the following coalition problem in $\mathsf{DSNF_{CL}}$ $(\mathcal{I}, \mathcal{U}, \mathcal{N})$:

$$\mathcal{I} = \{1.\ t_0\}$$

$$\begin{aligned}\mathcal{U} = \{&27.\ \neg t_0 \vee t_3 \vee t_4 \vee t_5,\\ &28.\ \neg t_1 \vee p,\\ &29.\ \neg t_1 \vee q,\\ &30.\ \neg t_2 \vee q,\\ &31.\ \neg t_2 \vee r\}\end{aligned}$$

$$\begin{aligned}\mathcal{N} = \{\ &7.\ t_0 \Rightarrow [1]t_1,\\ &9.\ t_0 \Rightarrow [1]t_2,\\ &20.\ t_3 \Rightarrow [\emptyset]\neg p,\\ &21.\ t_4 \Rightarrow [\emptyset]\neg q,\\ &22.\ t_5 \Rightarrow [\emptyset]\neg r\}\end{aligned}$$

3.2 Inference rules

Let $(\mathcal{I}, \mathcal{U}, \mathcal{N})$ be a coalition problem in $\mathsf{DSNF_{CL}}$; C, C' be conjunctions of literals; D, D' be disjunctions of literals; l, l_i be literals; and $\mathcal{A}, \mathcal{B} \subseteq \Sigma$ be coalitions (where Σ is the set of all agents).

Classical resolution: the first rule, **IRES1**, is classical resolution applied to clauses which are true at the initial state. The next inference rule, **GRES1**, performs resolution on clauses which are true in all states.

$$\textbf{IRES1}\ \frac{\begin{array}{ll} D \vee l & \in \mathcal{I} \\ D' \vee \neg l & \in \mathcal{I} \cup \mathcal{U} \end{array}}{D \vee D'} \qquad \textbf{GRES1}\ \frac{\begin{array}{ll} D \vee l & \in \mathcal{U} \\ D' \vee \neg l & \in \mathcal{U} \end{array}}{D \vee D'}$$

Coalition resolution: the following rules perform resolution on clauses which are true at the successor states.

$$\begin{array}{l} \textbf{CRES1} \\ \mathcal{A} \cap \mathcal{B} = \emptyset \end{array} \frac{\begin{array}{ll} C \Rightarrow [\mathcal{A}](D \vee l) & \in \mathcal{N} \\ C' \Rightarrow [\mathcal{B}](D' \vee \neg l) & \in \mathcal{N} \end{array}}{C \wedge C' \Rightarrow [\mathcal{A} \cup \mathcal{B}](D \vee D')} \qquad \textbf{CRES2}\ \frac{\begin{array}{ll} D \vee l & \in \mathcal{U} \\ C \Rightarrow [\mathcal{A}](D' \vee \neg l) & \in \mathcal{N} \end{array}}{C \Rightarrow [\mathcal{A}](D \vee D')}$$

$$\begin{array}{l} \textbf{CRES3} \\ \mathcal{A} \subseteq \mathcal{B} \end{array} \frac{\begin{array}{ll} C \Rightarrow [\mathcal{A}](D \vee l) & \in \mathcal{N} \\ C' \Rightarrow \langle\mathcal{B}\rangle(D' \vee \neg l) & \in \mathcal{N} \end{array}}{C \wedge C' \Rightarrow \langle\mathcal{B} \setminus \mathcal{A}\rangle(D \vee D')} \qquad \textbf{CRES4}\ \frac{\begin{array}{ll} D \vee l & \in \mathcal{U} \\ C \Rightarrow \langle\mathcal{A}\rangle(D' \vee \neg l) & \in \mathcal{N} \end{array}}{C \Rightarrow \langle\mathcal{A}\rangle(D \vee D')}$$

Rewriting rules:

$$\textbf{RW1}\ \frac{\bigwedge_{i=1}^{n} l_i \Rightarrow [\mathcal{A}]\textbf{false} \in \mathcal{N}}{\bigvee_{i=1}^{n} \neg l_i} \qquad \textbf{RW2}\ \frac{\bigwedge_{i=1}^{n} l_i \Rightarrow \langle\mathcal{A}\rangle\textbf{false} \in \mathcal{N}}{\bigvee_{i=1}^{n} \neg l_i}$$

Note that the axioms $\perp$ and $\top$, given by $\neg[\mathcal{A}]\textbf{false}$ and $[\mathcal{A}]\textbf{true}$, respectively, imply that the consequent in both rewriting rules cannot be satisfied. Thus, the conclusions from both rewriting rules ensure that $\bigwedge_{i=1}^{n} l_i$ should not be satisfied at any state.

DEFINITION 3.5
A **derivation** from a coalition problem in $\mathsf{DSNF_{CL}}$ $\mathcal{C}=(\mathcal{I},\mathcal{U},\mathcal{N})$ by $\mathsf{RES_{CL}}$ is a sequence $\mathcal{C}_0,\mathcal{C}_1,\mathcal{C}_2,\ldots$ of problems such that $\mathcal{C}_0=\mathcal{C}$, $\mathcal{C}_i=(\mathcal{I}_i,\mathcal{U}_i,\mathcal{N}_i)$, and $\mathcal{C}_{i+1}$ is either

- $(\mathcal{I}_i\cup\{D\},\mathcal{U}_i,\mathcal{N}_i)$, where D is the conclusion of an application of **IRES1**;
- $(\mathcal{I}_i,\mathcal{U}_i\cup\{D\},\mathcal{N}_i)$, where D is the conclusion of an application of **GRES1**, **RW1**, or **RW2**; or
- $(\mathcal{I}_i,\mathcal{U}_i,\mathcal{N}_i\cup\{D\})$, where D is the conclusion of an application of **CRES1**, **CRES2**, **CRES3**, or **CRES4** ;

and $D\notin\{\textbf{true},\textbf{false}\Rightarrow\varphi,\varphi\Rightarrow\textbf{true}\}$, for any formula φ.

We note that the resolvent D is not a tautology and it is always kept in the simplest form: duplicate literals are removed; constants **true** and **false** are removed from conjunctions and disjunctions with more than one conjunct/disjunct, respectively; conjunctions (resp. disjunctions) with either complementary literals or **false** (resp. **true**) are simplified to **false** (resp. **true**).

DEFINITION 3.6
A **refutation** for a coalition problem in $\mathsf{DSNF_{CL}}$ $\mathcal{C}=(\mathcal{I},\mathcal{U},\mathcal{N})$ (by $\mathsf{RES_{CL}}$) is a derivation from $\mathcal{C}$ such that for some $i\geq 0$, $\mathcal{C}_i=(\mathcal{I}_i,\mathcal{U}_i,\mathcal{N}_i)$ contains a contradiction, where a contradiction is given by either $\textbf{false}\in\mathcal{I}_i$ or $\textbf{false}\in\mathcal{U}_i$.

A derivation *terminates* if, and only if, either a contradiction is derived or no new clauses can be derived by further application of resolution rules of $\mathsf{RES_{CL}}$.

EXAMPLE 3.7
In order to verify the validity of the formula

$$[1](p\wedge q)\wedge[1](q\wedge r)\Rightarrow[1]p\wedge[1]q\wedge[1]r$$

we apply the resolution method to the coalition problem in $\mathsf{DSNF_{CL}}$ given in Example 3.4, which shows the transformation of its negation. Note that the original formula is in fact valid. Recall that the monotonicity principle, which holds in CL, is expressed by the schema $[\mathcal{A}](\varphi\wedge\psi)\Rightarrow[\mathcal{A}]\varphi\wedge[\mathcal{A}]\psi$, where φ and ψ are CL formulae and $\mathcal{A}$ is a coalition. Therefore, by monotonicity and by propositional reasoning, we have that $[1](p\wedge q)\wedge[1](q\wedge r)$ implies $([1]p\wedge[1]q)\wedge([1]q\wedge[1]r)$. The proof that the corresponding coalition problem in $\mathsf{DSNF_{CL}}$ is indeed unsatisfiable is presented below. The full proof, where clauses (1)-(11) from Example 3.4 have been renumbered, is given below:

1.	t_0	$[\mathcal{I}]$
2.	$\neg t_0\vee t_3\vee t_4\vee t_5$	$[\mathcal{U}]$
3.	$\neg t_1\vee p$	$[\mathcal{U}]$
4.	$\neg t_1\vee q$	$[\mathcal{U}]$
5.	$\neg t_2\vee q$	$[\mathcal{U}]$
6.	$\neg t_2\vee r$	$[\mathcal{U}]$
7.	$t_0\Rightarrow[1]t_1$	$[\mathcal{N}]$
8.	$t_0\Rightarrow[1]t_2$	$[\mathcal{N}]$
9.	$t_3\Rightarrow[\emptyset]\neg p$	$[\mathcal{N}]$
10.	$t_4\Rightarrow[\emptyset]\neg q$	$[\mathcal{N}]$
11.	$t_5\Rightarrow[\emptyset]\neg r$	$[\mathcal{N}]$
12.	$t_5\Rightarrow[\emptyset]\neg t_2$	$[\mathcal{N}$,**CRES2**,11,6]
13.	$t_4\Rightarrow[\emptyset]\neg t_1$	$[\mathcal{N}$,**CRES2**,10,4]
14.	$t_3\Rightarrow[\emptyset]\neg t_1$	$[\mathcal{N}$,**CRES2**,9,3]
15.	$t_0\wedge t_5\Rightarrow[1]\textbf{false}$	$[\mathcal{N}$,**CRES1**,12,8]
16.	$t_0\wedge t_4\Rightarrow[1]\textbf{false}$	$[\mathcal{N}$,**CRES1**,13,7]
17.	$t_0\wedge t_3\Rightarrow[1]\textbf{false}$	$[\mathcal{N}$,**CRES1**,14,7]
18.	$\neg t_0\vee\neg t_5$	$[\mathcal{U}$,**RW1**,15]
19.	$\neg t_0\vee\neg t_4$	$[\mathcal{U}$,**RW1**,16]
20.	$\neg t_0\vee\neg t_3$	$[\mathcal{U}$,**RW1**,17]
21.	$\neg t_0\vee t_3\vee t_4$	$[\mathcal{U}$,**GRES1**,18,2]
22.	$\neg t_0\vee t_3$	$[\mathcal{U}$,**GRES1**,21,19]
23.	$\neg t_0$	$[\mathcal{U}$,**GRES1**,22,20]
24.	**false**	$[\mathcal{I}$,**IRES1**,23,1]

4 Correctness results

In the previous section, we introduced a resolution-based method for CL. We now provide the correctness results, that is, soundness, termination and completeness results for this method. The soundness proof shows that the transformation into $\mathsf{DSNF}_{\mathsf{CL}}$ as well as the application of the inference rules are satisfiability preserving. Termination is ensured by the fact that a given set of clauses contains only finitely many propositional symbols, from which only finitely many $\mathsf{DSNF}_{\mathsf{CL}}$ clauses can be constructed and therefore only finitely many new $\mathsf{DSNF}_{\mathsf{CL}}$ clauses can be derived. Completeness is proved by showing that if a given set of clauses is unsatisfiable, there is a refutation produced by $\mathsf{RES}_{\mathsf{CL}}$. This corresponds to *refutational completeness*. The resolution calculus presented here, just as the tableau methods for CL [9, 12], is not intended as a *deductively complete* proof method, that is, a calculus which derives all possible consequences from a coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$. For instance, we do not resolve literals in the global set of clauses with literals on the left-hand side of clauses in the coalition set, although this would result in valid consequences of these clauses. Such inferences are not needed for refutational completeness and their absence improves the efficiency of the method in practical applications.

4.1 Correctness of the transformation rules

We show that the transformation rules given in Section 3.1 preserve satisfiability.

LEMMA 4.1
Let $\varphi \in \mathsf{WFF}_{\mathsf{CL}}$ be a formula and let $\mathcal{M} = (\Sigma_\varphi, \mathcal{S}, s_0, d, \delta, \Pi, \pi)$ be a CGM such that $\mathcal{M} \models \varphi$. Let $p \in \Pi$ be an atomic proposition not occurring in φ, and let $\mathcal{M}' = (\Sigma_\varphi, \mathcal{S}, s_0, d, \delta, \Pi, \pi')$ be a CGM identical to $\mathcal{M}$ except for the truth value assigned by π' to p in each state. Then $\mathcal{M}' \models \varphi$.

In the following $\varphi_1, \varphi_2 \in \mathsf{WFF}_{\mathsf{CL}}$, D is a disjunction, t is a literal, and t_0, t_1 are new propositional symbols.

LEMMA 4.2
A formula $\varphi \in \mathsf{WFF}_{\mathsf{CL}}$ is satisfiable if, and only if, the coalition problem $\mathcal{C} = (\{t_0\}, \{t_0 \Rightarrow \varphi\}, \{\})$ is satisfiable, where t_0 does not occur in φ.

PROOF OF LEMMA 4.2. ($\Rightarrow$) Let $\mathcal{M} = (\Sigma_\varphi, \mathcal{S}, s_0, d, \delta, \Pi, \pi)$ be a CGM such that $\mathcal{M} \models \varphi$. Construct a model $\mathcal{M}' = (\Sigma_{\mathcal{C}}, \mathcal{S}, s_0, d, \delta, \Pi, \pi')$, such that $\pi'(s_0) = \pi(s_0) \cup \{t_0\}$, $\pi'(s) = \pi(s) \setminus \{t_0\}$ for all $s \in \mathcal{S}, s \neq s_0$, and $\Sigma_{\mathcal{C}} = \Sigma_\varphi$. The satisfiability of $\mathcal{C}$ follows from Lemma 4.1, semantics of implication, and the definition of satisfiability of a coalition problem.

($\Leftarrow$) Let $\mathcal{M} = (\Sigma_{\mathcal{C}}, \mathcal{S}, s_0, d, \delta, \Pi, \pi)$ be a CGM such that $\mathcal{M} \models \mathcal{C}$. By the definition of satisfiability for a coalition problem and semantics of implication, we have that $\mathcal{M} \models \varphi$. ■

LEMMA 4.3 ($\tau_\wedge$)
Let $\mathcal{C} = (\mathcal{I}, \mathcal{U} \cup \{t \Rightarrow \varphi_1 \wedge \varphi_2\}, \mathcal{N})$ be a coalition problem. $\mathcal{C}$ is satisfiable if, and only if, $(\mathcal{I}, \mathcal{U} \cup \{t \Rightarrow \varphi_1, t \Rightarrow \varphi_2\}, \mathcal{N})$ is satisfiable.

PROOF OF LEMMA 4.3. Immediate from the definition of satisfiability of a coalition problem and semantics of conjunction. ■

LEMMA 4.4 ($\tau_\vee$)
Let $\mathcal{C} = (\mathcal{I}, \mathcal{U} \cup \{t \Rightarrow \varphi_1 \vee \varphi_2\}, \mathcal{N})$ be a coalition problem. $\mathcal{C}$ is satisfiable if, and only if, $(\mathcal{I}, \mathcal{U} \cup \{t \Rightarrow \varphi_1 \vee t_1, t_1 \Rightarrow \varphi_2\}, \mathcal{N})$ is satisfiable, where t_1 does not occur in $\mathcal{C}$.

Proof of Lemma 4.4. Immediate from Lemma 4.1, the definition of satisfiability of a coalition problem, and semantics of disjunction. ■

Lemma 4.5 ($\tau_{\Rightarrow}$)
Let $\mathcal{C}=(\mathcal{I},\mathcal{U}\cup\{t\Rightarrow D\},\mathcal{N})$ be a coalition problem, where t is a literal. $(\mathcal{I},\mathcal{U}\cup\{t\Rightarrow D\},\mathcal{N})$ is satisfiable if, and only if,

(1) $(\mathcal{I},\mathcal{U}\cup\{\neg t\vee D\},\mathcal{N})$ is satisfiable, if D is either a constant or a disjunction of literals;
(2) $(\mathcal{I},\mathcal{U},\mathcal{N}\cup\{t\Rightarrow D\})$ is satisfiable, if D is either of the form $[\mathcal{A}]\varphi_1$ or $\langle\mathcal{A}\rangle\varphi_1$.

Proof of Lemma 4.5. Immediate from the definition of satisfiability of a coalition problem and semantics of implication. ■

Lemma 4.6 ($\tau_{[\mathcal{A}]}$)
Let $\mathcal{C}=(\mathcal{I},\mathcal{U},\mathcal{N}\cup\{t\Rightarrow[\mathcal{A}]\varphi_1\})$ be a coalition problem. $\mathcal{C}$ is satisfiable if, and only if, $(\mathcal{I},\mathcal{U}\cup\{t_1\Rightarrow \varphi_1\},\mathcal{N}\cup\{t\Rightarrow[\mathcal{A}]t_1\})$ is satisfiable, where t_1 does not occur in $\mathcal{C}$.

Proof of Lemma 4.6. ($\Rightarrow$) Let $\mathcal{M}=(\Sigma_{\mathcal{C}},\mathcal{S},s_0,d,\delta,\Pi,\pi)$ be a CGM such that $\mathcal{M}\models\mathcal{C}$. By the definition of satisfiability for coalition problems, $\langle\mathcal{M},s\rangle\models t\Rightarrow[\mathcal{A}]\varphi_1$, for all $s\in\mathcal{S}$. By Lemma 4.1, we can construct a model $\mathcal{M}'=(\Sigma_{\mathcal{C}},\mathcal{S},s_0,d,\delta,\Pi,\pi')$, such that $\pi'(s)=\pi(s)\cup\{t_1\}$ if $\langle\mathcal{M},s\rangle\models\varphi_1$; otherwise, $\pi'(s)=\pi(s)\setminus\{t_1\}$. It follows immediately that for all $s\in\mathcal{S}$, $\langle\mathcal{M}',s\rangle\models t_1\Rightarrow\varphi_1$. If $\langle\mathcal{M}',s\rangle\not\models t$, then $\langle\mathcal{M}',s\rangle\models t\Rightarrow[\mathcal{A}]t_1$. If $\langle\mathcal{M},s\rangle\models t$, then $\langle\mathcal{M},s\rangle\models[\mathcal{A}]\varphi_1$, as formulae in the set of coalition clauses are satisfied at all states. Therefore, there is a $\mathcal{A}$-move $\sigma_{\mathcal{A}}$ such that $\langle\mathcal{M},s'\rangle\models\varphi_1$ for all $s'\in out(s,\sigma_{\mathcal{A}})$. The sets of outcomes of s in $\mathcal{M}$ and in $\mathcal{M}'$ are exactly the same, as those models share the same number of moves (given by d) and the same transition function (given by δ). Thus, for the same $\mathcal{A}$-move $\sigma_{\mathcal{A}}$, for all $s'\in out(s,\sigma_{\mathcal{A}})$, we have $\langle\mathcal{M}',s'\rangle\models\varphi_1$ and, by construction, $\langle\mathcal{M}',s'\rangle\models t_1$. By the semantics of the implication and of the coalition modality, we have that $\langle\mathcal{M}',s\rangle\models t\Rightarrow[\mathcal{A}]t_1$. By the definition of satisfiability for coalition problems, $\mathcal{M}'\models(\mathcal{I},\mathcal{U}\cup\{t_1\Rightarrow\varphi_1\},\mathcal{N}\cup\{t\Rightarrow[\mathcal{A}]t_1\})$.

($\Leftarrow$) Let $\mathcal{M}=(\Sigma_{\mathcal{C}},\mathcal{S},s_0,d,\delta,\Pi,\pi)$ be a CGM such that $\mathcal{M}\models(\mathcal{I},\mathcal{U}\cup\{t_1\Rightarrow\varphi_1\},\mathcal{N}\cup\{t\Rightarrow[\mathcal{A}]t_1\})$. By the definition of satisfiability for coalition problems, $\langle\mathcal{M},s\rangle\models t\Rightarrow[\mathcal{A}]t_1$, for all $s\in\mathcal{S}$. If $\langle\mathcal{M},s\rangle\not\models t$, then $\langle\mathcal{M},s\rangle\models t\Rightarrow[\mathcal{A}]\varphi_1$. If $\langle\mathcal{M},s\rangle\models t$, by the semantics of implication, $\langle\mathcal{M},s\rangle\models[\mathcal{A}]t_1$ and, by the semantics of coalition modalities there is a $\mathcal{A}$-move $\sigma_{\mathcal{A}}$ such that $\langle\mathcal{M},s'\rangle\models t_1$ for all $s'\in out(s,\sigma_{\mathcal{A}})$. As $t_1\Rightarrow\varphi_1$ is satisfiable at all states of the model (by the definition of satisfiability of a coalition problem), for the same $\mathcal{A}$-move $\sigma_{\mathcal{A}}$, for all $s'\in out(s,\sigma_{\mathcal{A}})$, we have $\langle\mathcal{M},s'\rangle\models\varphi_1$. By the semantics of coalition modalities, $\langle\mathcal{M},s\rangle\models[\mathcal{A}]\varphi_1$. Thus, $\langle\mathcal{M},s\rangle\models t\Rightarrow[\mathcal{A}]\varphi_1$. By the definition of satisfiability for coalition problems, $\mathcal{M}\models\mathcal{C}$. ■

Lemma 4.7 ($\tau_{\langle\mathcal{A}\rangle,\mathcal{A}\neq\Sigma_{\varphi}}$)
Let $\mathcal{C}=(\mathcal{I},\mathcal{U},\mathcal{N}\cup\{t\Rightarrow\langle\mathcal{A}\rangle\varphi_1\})$ be a coalition problem. $\mathcal{C}$ is satisfiable if, and only if, $(\mathcal{I},\mathcal{U}\cup\{t_1\Rightarrow \varphi_1\},\mathcal{N}\cup\{t\Rightarrow\langle\mathcal{A}\rangle t_1\})$ is satisfiable, where t_1 does not occur in $\mathcal{C}$.

Proof of Lemma 4.7. ($\Rightarrow$) Let $\mathcal{M}=(\Sigma_{\mathcal{C}},\mathcal{S},s_0,d,\delta,\Pi,\pi)$ be a CGM such that $\mathcal{M}\models\mathcal{C}$. By the definition of satisfiability for coalition problems, $\langle\mathcal{M},s\rangle\models t\Rightarrow\langle\mathcal{A}\rangle\varphi_1$, for all $s\in\mathcal{S}$. By Lemma 4.1, we can construct a model $\mathcal{M}'=(\Sigma_{\mathcal{C}},\mathcal{S},s_0,d,\delta,\Pi,\pi')$, such that $\pi'(s)=\pi(s)\cup\{t_1\}$ if $\langle\mathcal{M},s\rangle\models\varphi_1$; otherwise, $\pi'(s)=\pi(s)\setminus\{t_1\}$. It follows immediately that for all $s\in\mathcal{S}$, $\langle\mathcal{M}',s\rangle\models t_1\Rightarrow\varphi_1$. If $\langle\mathcal{M},s\rangle\not\models t$, then $t\Rightarrow\langle\mathcal{A}\rangle t_1$ is trivially satisfied at $\langle\mathcal{M}',s\rangle$. If $\langle\mathcal{M},s\rangle\models t$, because $\langle\mathcal{M},s\rangle\models t\Rightarrow\langle\mathcal{A}\rangle\varphi_1$, we have that $\langle\mathcal{M},s\rangle\models\langle\mathcal{A}\rangle\varphi_1$. By the semantics of a coalition modality, for all $\mathcal{A}$-moves $\sigma_{\mathcal{A}}$ there is $s'\in out(s,\sigma_{\mathcal{A}})$ such that $\langle\mathcal{M},s'\rangle\models\varphi_1$. The sets of outcomes of s in $\mathcal{M}$ and in $\mathcal{M}'$ are exactly the same, as those models share the same number of moves (given by d) and the same transition function (given by δ). Therefore, for all $\mathcal{A}$-moves $\sigma_{\mathcal{A}}$, there is a $s' \in out(s, \sigma_{\mathcal{A}})$ such that $\langle \mathcal{M}', s' \rangle \models \varphi_1$ and, by construction, $\langle \mathcal{M}', s' \rangle \models t_1$. By the semantics of the coalition modality, we have that $\langle \mathcal{M}', s \rangle \models t \Rightarrow \langle \mathcal{A} \rangle t_1$. By the definition of satisfiability for coalition problems, $\mathcal{M}' \models (\mathcal{I}, \mathcal{U} \cup \{t_1 \Rightarrow \varphi_1\}, \mathcal{N} \cup \{t \Rightarrow \langle \mathcal{A} \rangle t_1\})$.

($\Leftarrow$) Let $\mathcal{M} = (\Sigma_{\mathcal{C}}, \mathcal{S}, s_0, d, \delta, \Pi, \pi)$ be a CGM such that $\mathcal{M} \models (\mathcal{I}, \mathcal{U} \cup \{t_1 \Rightarrow \varphi_1\}, \mathcal{N} \cup \{t \Rightarrow \langle \mathcal{A} \rangle t_1\})$. By the definition of satisfiability for coalition problems, $\langle \mathcal{M}, s \rangle \models t \Rightarrow \langle \mathcal{A} \rangle t_1$, for all $s \in \mathcal{S}$. If $\langle \mathcal{M}, s \rangle \not\models t$, then $\langle \mathcal{M}, s \rangle \models t \Rightarrow \langle \mathcal{A} \rangle \varphi_1$. Now, if $\langle \mathcal{M}, s \rangle \models t$, by the semantics of implication and coalition modalities, then for all $\mathcal{A}$-moves $\sigma_{\mathcal{A}}$ there is $s' \in out(s, \sigma_{\mathcal{A}})$ such that $\langle \mathcal{M}, s' \rangle \models t_1$. By the definition of satisfiability for coalition problems, $\langle \mathcal{M}, s \rangle \models t_1 \Rightarrow \varphi_1$, for all $s \in \mathcal{S}$, thus for all $\mathcal{A}$-moves $\sigma_{\mathcal{A}}$, there is $s' \in out(s, \sigma_{\mathcal{A}})$ such that $\langle \mathcal{M}, s' \rangle \models \varphi_1$. By the semantics of coalition modalities, $\langle \mathcal{M}, s \rangle \models \langle \mathcal{A} \rangle \varphi_1$. Therefore, $\langle \mathcal{M}, s \rangle \models t \Rightarrow \langle \mathcal{A} \rangle \varphi_1$. By the definition of satisfiability for coalition problems, $\mathcal{M} \models \mathcal{C}$. ■

LEMMA 4.8 ($\tau_{\langle \Sigma_\varphi \rangle}$)
Let $\mathcal{C} = (\mathcal{I}, \mathcal{U}, \mathcal{N} \cup \{t \Rightarrow \langle \Sigma_\varphi \rangle \varphi_1\})$ be a coalition problem. $\mathcal{C}$ is satisfiable if, and only if, $(\mathcal{I}, \mathcal{U}, \mathcal{N} \cup \{t \Rightarrow [\emptyset] \varphi_1\})$ is satisfiable.

PROOF OF LEMMA 4.8. The proof follows from the axiomatization of CL, as $[\emptyset]\varphi_1 \Leftrightarrow \langle \Sigma_\varphi \rangle \varphi_1$ is valid. ■

THEOREM 4.9
Let $\varphi \in \mathsf{WFF_{CL}}$. Let $\mathcal{C}_0, \mathcal{C}_1, \ldots$ be a sequence of coalition problems such that $\mathcal{C}_0 = (\{t_0\}, \{t_0 \Rightarrow \tau_0(\varphi)\}, \{\})$ and $\mathcal{C}_{i+1}$ is obtained from $\mathcal{C}_i$ by applying a transformation rule combined with zero or more applications of the simplification rules to a formula in $\mathcal{C}_i$. Then the sequence $\mathcal{C}_0, \mathcal{C}_1, \ldots$ terminates, i.e. there exists an index n, $n \geq 0$, such that no transformation rule can be applied to $\mathcal{C}_n$. Furthermore, $\mathcal{C}_n$ is a coalition problem in $\mathsf{DSNF_{CL}}$ and $\mathcal{C}_n$ is satisfiable if, and only if, φ is satisfiable.

PROOF OF THEOREM 4.9. Termination can be shown by defining a weight function w that maps each coalition problem to a pair of natural numbers and proving that each application of a transformation rule to a coalition problem $\mathcal{C}_i$ results in a coalition problem $\mathcal{C}_{i+1}$ such that $w(\mathcal{C}_i) > w(\mathcal{C}_{i+1})$, where $>$ is the lexicographic combination of the $>$ ordering on natural numbers with itself. To prove that $\mathcal{C}_n$ is a coalition problem in $\mathsf{DSNF_{CL}}$ we show that to any coalition problem $\mathcal{C}_i$ that is not in $\mathsf{DSNF_{CL}}$ we can apply one of the transformation rules. Finally, that $\mathcal{C}_n$ is satisfiable if, and only if, φ is satisfiable follows from Lemmas 4.1 to 4.8, which show that each individual application of a transformation rule preserves satisfiability. ■

4.2 Soundness

We now show that each of the inference rules given in Section 3.2 is sound. In the following, C, C' are conjunctions of literals; D, D' are disjunctions of literals; l, l_i are literals; and $\mathcal{A}, \mathcal{B} \subseteq \Sigma$ are coalitions (where Σ is the set of all agents).

LEMMA 4.10 (**Resolution**)
Let $\mathcal{M} = (\Sigma_\varphi, \mathcal{S}, s_0, d, \delta, \Pi, \pi)$ be a CGM, such that $\langle \mathcal{M}, s \rangle \models D \vee l$ and $\langle \mathcal{M}, s \rangle \models D' \vee \neg l$, for some $s \in \mathcal{S}$. Then $\langle \mathcal{M}, s \rangle \models D \vee D'$.

LEMMA 4.11 (**IRES1**)
Let $\mathcal{C} = (\mathcal{I}, \mathcal{U}, \mathcal{N})$ be a coalition problem in $\mathsf{DSNF_{CL}}$, such that $D \vee l \in \mathcal{I}$ and $D' \vee \neg l \in \mathcal{I} \cup \mathcal{U}$. If $\mathcal{C}$ is satisfiable, then $(\mathcal{I} \cup \{D \vee D'\}, \mathcal{U}, \mathcal{N})$ is satisfiable.

LEMMA 4.12 (**GRES1**)
Let $\mathcal{C}=(\mathcal{I},\mathcal{U},\mathcal{N})$ be a coalition problem in $\mathsf{DSNF_{CL}}$, such that $D\vee l\in\mathcal{U}$ and $D'\vee\neg l\in\mathcal{U}$. If $\mathcal{C}$ is satisfiable, then $(\mathcal{I},\mathcal{U}\cup\{D\vee D'\},\mathcal{N})$ is satisfiable.

The proofs of Lemmas 4.10, 4.11, and 4.12 follow from soundness of the resolution method for propositional logic [19].

LEMMA 4.13 (**CRES1**)
Let $\mathcal{C}=(\mathcal{I},\mathcal{U},\mathcal{N})$ be a coalition problem in $\mathsf{DSNF_{CL}}$, such that $C\Rightarrow[\mathcal{A}](D\vee l)\in\mathcal{N}$ and $C'\Rightarrow[\mathcal{B}](D'\vee\neg l)\in\mathcal{N}$, where $\mathcal{A}\cap\mathcal{B}=\emptyset$. If $\mathcal{C}$ is satisfiable, then $(\mathcal{I},\mathcal{U},\mathcal{N}\cup\{C\wedge C'\Rightarrow[\mathcal{A}\cup\mathcal{B}](D\vee D')\})$ is satisfiable.

PROOF OF LEMMA 4.13. Let $\mathcal{M}=(\Sigma_{\mathcal{C}},\mathcal{S},s_0,d,\delta,\Pi,\pi)$ be a CGM such that $\mathcal{M}\models\mathcal{C}$. By the definition of satisfiability of coalition problems, all formulae in $\mathcal{N}$ are satisfied at all states. For $s\in\mathcal{S}$, we have that $\langle\mathcal{M},s\rangle\models C\Rightarrow[\mathcal{A}](D\vee l)$ and $\langle\mathcal{M},s\rangle\models C'\Rightarrow[\mathcal{B}](D'\vee\neg l)$. If $\langle\mathcal{M},s\rangle\not\models C\wedge C'$, then the implication $C\wedge C'\Rightarrow[\mathcal{A}\cup\mathcal{B}](D\vee D')$ is satisfied at s. Assume that $\langle\mathcal{M},s\rangle\models C\wedge C'$. By the semantics of conjunction and implication, we have that $\langle\mathcal{M},s\rangle\models C\wedge C'\Rightarrow[\mathcal{A}](D\vee l)\wedge[\mathcal{B}](D'\vee\neg l)$. By axiom **S**, we have that $[\mathcal{A}](D\vee l)\wedge[\mathcal{B}](D'\vee\neg l)$ implies $[\mathcal{A}\cup\mathcal{B}]((D\vee l)\wedge(D'\vee\neg l))$. Therefore, $\langle\mathcal{M},s\rangle\models[\mathcal{A}\cup\mathcal{B}]((D\vee l)\wedge(D'\vee\neg l))$. By the definition of satisfiability for coalition modalities, there is a $\mathcal{A}\cup\mathcal{B}$-move $\sigma_{\mathcal{A}\cup\mathcal{B}}$ such that for all $s'\in out(s,\sigma_{\mathcal{A}})\cap out(s,\sigma_{\mathcal{B}})$ we have that $\langle\mathcal{M},s'\rangle\models(D\vee l)$ and $\langle\mathcal{M},s'\rangle\models(D'\vee\neg l)$. By Lemma 4.10 applied at s', we have that $\langle\mathcal{M},s'\rangle\models D\vee D'$. Again, by the definition of satisfiability of the coalition modality, we have that $\langle\mathcal{M},s\rangle\models[\mathcal{A}\cup\mathcal{B}](D\vee D')$. By the definition of satisfiability of sets, $\mathcal{N}\cup\{C\wedge C'\Rightarrow[\mathcal{A}\cup\mathcal{B}](D\vee D')\}$ is satisfiable. By the definition of satisfiability of coalition problems, $(\mathcal{I},\mathcal{U},\mathcal{N}\cup\{C\wedge C'\Rightarrow[\mathcal{A}\cup\mathcal{B}](D\vee D')\})$ is satisfiable. ∎

LEMMA 4.14
Let $\mathcal{C}=(\mathcal{I},\mathcal{U},\mathcal{N})$ be a coalition problem and $\mathcal{M}$ be a model such that $\mathcal{M}\models\mathcal{C}$. If φ is a formula in $\mathcal{U}$, then $\mathcal{M}\models(\mathcal{I},\mathcal{U},\mathcal{N}\cup\{\mathbf{true}\Rightarrow[\emptyset]\varphi\})$.

PROOF OF LEMMA 4.14. Let $\mathcal{M}=(\Sigma_{\mathcal{C}},\mathcal{S},s_0,d,\delta,\Pi,\pi)$ be a CGM such that $\mathcal{M}\models\mathcal{C}$. As $\varphi\in\mathcal{U}$, then by the definition of satisfiability for a coalition problem, for all $s\in\mathcal{S}$, $\langle\mathcal{M},s\rangle\models\varphi$. Therefore, for all σ moves in $D(s)$, for all states $s\in\mathcal{S}$, we have if $s'\in out(s,\sigma)$, then $\langle\mathcal{M},s'\rangle\models\varphi$. By the semantics of a coalition modality, we have that $\langle\mathcal{M},s\rangle\models[\emptyset]\varphi$. By the semantics of implication,$\langle\mathcal{M},s\rangle\models\mathbf{true}\Rightarrow[\emptyset]\varphi$. By the definition of satisfiability of a coalition problem, $\mathcal{M}\models(\mathcal{I},\mathcal{U},\mathcal{N}\cup\{\mathbf{true}\Rightarrow[\emptyset]\varphi\})$. ∎

LEMMA 4.15 (**CRES2**)
Let $\mathcal{C}=(\mathcal{I},\mathcal{U},\mathcal{N})$ be a coalition problem in $\mathsf{DSNF_{CL}}$, such that $(D\vee l)\in\mathcal{U}$ and $C\Rightarrow[\mathcal{A}](D'\vee\neg l)\in\mathcal{N}$. If $\mathcal{C}$ is satisfiable, then $(\mathcal{I},\mathcal{U},\mathcal{N}\cup\{C\Rightarrow[\mathcal{A}](D\vee D')\})$ is satisfiable.

PROOF OF LEMMA 4.15. Let $\mathcal{M}=(\Sigma_{\mathcal{C}},\mathcal{S},s_0,d,\delta,\Pi,\pi)$ be a CGM such that $\mathcal{M}\models\mathcal{C}$. As $(D\vee l)\in\mathcal{U}$, by Lemma 4.14, $\mathbf{true}\Rightarrow[\emptyset](D\vee l)$ is satisfied at all states. From this and from Lemma 4.13, we have that $(\mathcal{I},\mathcal{U},\mathcal{N}\cup\{C\wedge\Rightarrow[\mathcal{A}](D\vee D')\})$ is satisfiable. ∎

LEMMA 4.16 (**CRES3**)
Let $\mathcal{C}=(\mathcal{I},\mathcal{U},\mathcal{N})$ be a coalition problem in $\mathsf{DSNF_{CL}}$, such that $C\Rightarrow[\mathcal{A}](D\vee l)\in\mathcal{N}$ and $C'\Rightarrow\langle\mathcal{B}\rangle(D'\vee\neg l)\in\mathcal{N}$, where $\mathcal{A}\subseteq\mathcal{B}$. If $\mathcal{C}$ is satisfiable, then $(\mathcal{I},\mathcal{U},\mathcal{N}\cup\{C\wedge C'\Rightarrow\langle\mathcal{B}\setminus\mathcal{A}\rangle(D\vee D')\})$ is satisfiable.

PROOF OF LEMMA 4.16. From the axiomatization of CL, we have that (1) $[\mathcal{A}](D\vee l)\wedge\langle\mathcal{B}\rangle(D'\vee\neg l)\Rightarrow\langle\mathcal{B}\setminus\mathcal{A}\rangle((D\vee l)\wedge(D'\vee\neg l))$, with $\mathcal{A}\subseteq\mathcal{B}$, is valid. Let $\mathcal{M}=(\Sigma_{\mathcal{C}},\mathcal{S},s_0,d,\delta,\Pi,\pi)$ be a CGM such that $\mathcal{M}\models\mathcal{C}$. By the semantics of a coalition problem, for all $s\in\mathcal{S}$ we have that $\langle\mathcal{M},s\rangle\models C\Rightarrow[\mathcal{A}](D\vee l)$ and $\langle\mathcal{M},s\rangle\models C'\Rightarrow\langle\mathcal{B}\rangle(D'\vee\neg l)$. By the semantics of conjunction, semantics of implication, and from (1), we have that $\langle\mathcal{M},s\rangle\models C\wedge C'\Rightarrow\langle\mathcal{B}\setminus\mathcal{A}\rangle((D\vee l)\wedge(D'\vee\neg l))$. Assume $\langle\mathcal{M},s\rangle\models C\wedge C'$ (the other case is trivial). Thus, by the semantics of implication $\langle \mathcal{M}, s\rangle \models \langle \mathcal{B}\setminus\mathcal{A}\rangle((D\vee l)\wedge(D'\vee\neg l))$. From the semantics of the coalition modality, we have that for all $\mathcal{B}\setminus\mathcal{A}$-moves $\sigma_{\mathcal{B}\setminus\mathcal{A}}$ there is $s'\in out(s,\sigma_{\mathcal{B}\setminus\mathcal{A}})$ such that $\langle \mathcal{M}, s'\rangle \models ((D\vee l)\wedge(D'\vee\neg l))$. By applying Lemma 4.10 to s', we have that $\langle \mathcal{M}, s'\rangle \models (D\vee D')$. From the semantics of the coalition modality $\langle \mathcal{M}, s\rangle \models \langle \mathcal{B}\setminus\mathcal{A}\rangle(D\vee D')$. By the semantics of implication, $\langle \mathcal{M}, s\rangle \models C\wedge C' \Rightarrow \langle \mathcal{B}\setminus\mathcal{A}\rangle(D\vee D')$. By the definition of satisfiability of sets, $\mathcal{M}\models \mathcal{N}\cup\{C\wedge C'\Rightarrow\langle \mathcal{B}\setminus\mathcal{A}\rangle(D\vee D')\}$. From the definition of satisfiability of coalition problems, $\mathcal{M}\models(\mathcal{I},\mathcal{U},\mathcal{N}\cup\{C\wedge C'\Rightarrow\langle \mathcal{B}\setminus\mathcal{A}\rangle(D\vee D')\})$. ■

LEMMA 4.17 (**CRES4**)
Let $\mathcal{C}=(\mathcal{I},\mathcal{U},\mathcal{N})$ be a coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$, such that $(D\vee l)\in\mathcal{U}$ and $C\Rightarrow\langle\mathcal{A}\rangle(D'\vee\neg l)\in\mathcal{N}$. If $\mathcal{C}$ is satisfiable, then $(\mathcal{I},\mathcal{U},\mathcal{N}\cup\{C\Rightarrow\langle\mathcal{A}\rangle(D\vee D')\})$ is satisfiable.

PROOF OF LEMMA 4.17. From Lemma 4.14, $(D\vee l)\in\mathcal{U}$ implies that $\mathbf{true}\Rightarrow[\emptyset](D\vee l)$ is satisfied at every state of a model. Therefore, the satisfiability of $(\mathcal{I},\mathcal{U},\mathcal{N}\cup\{C\Rightarrow\langle\mathcal{A}\rangle(D\vee D')\})$ follows from the application of Lemma 4.16 to the coalition problem $(\mathcal{I},\mathcal{U},\mathcal{N}\cup\{\mathbf{true}\Rightarrow[\emptyset](D\vee l), C\Rightarrow\langle\mathcal{A}\rangle(D'\vee\neg l)\})$. ■

LEMMA 4.18 (**RW1**)
Let $\mathcal{C}=(\mathcal{I},\mathcal{U},\mathcal{N})$ be a coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$, such that $C\Rightarrow[\mathcal{A}]\mathbf{false}\in\mathcal{N}$. If $\mathcal{C}$ is satisfiable, then $(\mathcal{I},\mathcal{U}\cup\{\neg C\},\mathcal{N})$ is satisfiable.

PROOF OF LEMMA 4.18. From the axiomatization of CL, the schema $[\mathcal{A}]\mathbf{false}$ is unsatisfiable. Therefore, $[\mathcal{A}]\mathbf{false}$ implies $\mathbf{false}$. By classical reasoning, if a state satisfies $C\Rightarrow[\mathcal{A}]\mathbf{false}$, then the state also satisfies $C\Rightarrow\mathbf{false}$ and therefore $\neg C$. ■

LEMMA 4.19 (**RW2**)
Let $\mathcal{C}=(\mathcal{I},\mathcal{U},\mathcal{N})$ be a coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$, such that $C\Rightarrow\langle\mathcal{A}\rangle\mathbf{false}\in\mathcal{N}$. If $\mathcal{C}$ is satisfiable, then $(\mathcal{I},\mathcal{U}\cup\{\neg C\},\mathcal{N})$ is satisfiable.

PROOF OF LEMMA 4.19. From the axiomatization of CL, the schema $\langle\mathcal{A}\rangle\mathbf{false}$ is unsatisfiable. Therefore, if a state in a model satisfies $C\Rightarrow\langle\mathcal{A}\rangle\mathbf{false}$, it also satisfies $\neg C$. ■

The following theorem shows that the application of inference rules in $\mathsf{RES}_{\mathsf{CL}}$ is sound.

THEOREM 4.20 (Soundness of $\mathsf{RES}_{\mathsf{CL}}$)
Let $\mathcal{C}$ be a coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$. Let $\mathcal{C}'$ be the coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$ obtained from $\mathcal{C}$ by applying any of the inference rules **IRES1**, **GRES1**, **CRES1-4** and **RW1-2** to $\mathcal{C}$. If $\mathcal{C}$ is satisfiable, then $\mathcal{C}'$ is satisfiable.

PROOF OF THEOREM 4.20. The proof that the calculus preserves satisfiability follows from the fact that each inference rule preserves satisfiability, as given by Lemmas 4.11 to 4.19. ■

4.3 Termination

The proof that every derivation, as given by Definition 3.5, terminates is trivial and based on the fact that we have a finite number of clauses that can be expressed. As the number of propositional symbols after translation into the normal form is finite and the inference rules do not introduce new propositional symbols, we have that the number of possible literals occurring in clauses is finite and the number of conjunctions (resp. disjunctions) on the left-hand side (resp. right-hand side) of clauses is finite (modulo simplification). As the number of agents is finite, the number of coalition modalities that can be introduced by inference rules is also finite. Thus, only a finite number of clauses can be expressed (modulo simplification), so at some point either we derive a contradiction or no new clauses can be generated.

THEOREM 4.21
Let $\mathcal{C}=(\mathcal{I},\mathcal{U},\mathcal{N})$ be a coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$. Then any derivation from $\mathcal{C}$ by $\mathsf{RES}_{\mathsf{CL}}$ terminates.

4.4 Completeness

The completeness proof for $\mathsf{RES}_{\mathsf{CL}}$ is based on the tableau construction given in [9]. Given an unsatisfiable coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$ $\mathcal{C}$, a closed tableau is obtained by this construction. In this case, we show that there is a refutation by the resolution method presented here, that is, we show that the method is refutational complete. In particular, we show that the application of the resolution inference rules to (sub)sets of clauses in a coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$ correspond to (some) applications of the state deletion procedure in the tableau. We note that, as in [9], this corresponds to *weak completeness*, that is, if a coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$ is satisfiable, then a model can be obtained from the tableau.

In the following, we present the tableau procedure. The presentation will differ slightly from [9], as we adapt the method to the particular normal form presented in this article. The only modification introduced in the method is that we start the construction of a tableau from a set of formulae, instead of starting from a singleton set. This leads to a different (but equivalent) definition for a successful tableau, i.e. instead of checking if the input formula is part of some state of the resulting tableau, we check if the input set of formulae is a subset of some state. We then show how we use this procedure in order to obtain a tableau corresponding to a coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$. Additionally, as well as the set of clauses to be shown (un)satisfiable, the set of formulae, which is the input for the tableau procedure and represents the coalition problem, also contains a set of tautologies, which introduces as many literals as we need in the states of the resulting tableau. This helps to identify which sets of clauses and inference rules used in a derivation by the resolution method correspond to a state deleted from the tableau. This might affect the efficiency of the tableau method, but does not imply any changes in the correctness proof of the method presented in [9].

Graph construction: the procedure consists of three different phases: construction, prestate elimination, and state elimination. During the construction phase, a set of rules is used to build a directed graph called **pretableau**, which contains *states* and *prestates*. States are *downward saturated* sets of formulae, that is, sets of formulae to which all conjunctive (α) and disjunctive (β) rules given in Table 1(a) and (b) have been exhaustively applied. The first column in Table 1(a) (resp. (1b)) shows the premises, that is the α (resp. β) formulae to which an inference rule is applied; and the second column shows the n conclusions that are derived from the premises. The application of those inference rules are formalised below (Def. 4.23) after we precisely define the language to which those rules are applied. We note that the application of the inference rules to conjunctive formulae requires that all conclusions are added to the set of formulae whereas the application of the inference rules to disjunctive formulae requires only one conclusion to be added to the set of formulae. We also note that we have extended the α and β rules to deal with n-ary conjunctions and n-ary disjunctions, respectively. The rules given here can be simulated by several applications of the rules given in [9]. Also note that in a coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$, there is no formulae of the form $\langle\Sigma\rangle\varphi$ (as the application of the transformation rule τ_{Σ_ϕ} rewrites such

TABLE 1. Tableau rules

α	$\alpha_1,\dots,\alpha_n$
$\neg\neg\varphi$	φ
$\varphi_1\wedge\ldots\wedge\varphi_n$	$\varphi_1,\ldots,\varphi_n$
$\neg(\varphi_1\vee\ldots\vee\varphi_n)$	$\neg\varphi_1,\ldots,\neg\varphi_n$
$\langle\langle\emptyset\rangle\rangle\,\Box\varphi$	$\varphi,[\emptyset]\langle\langle\emptyset\rangle\rangle\,\Box\varphi$

(a) α rules.

β	$\beta_1\mid\ldots\mid\beta_n$
$\varphi_1\vee\ldots\vee\varphi_n$	$\varphi_1\mid\ldots\mid\varphi_n$
$\varphi_1\wedge\ldots\wedge\varphi_n\Rightarrow\psi$	$\neg\varphi_1\mid\ldots\mid\neg\varphi_n\mid\psi$

(b) β-rules

formulae) and the corresponding α rule has been suppressed. Prestates are also sets of formulae, but they do not need to be downward saturated; they are used as auxiliary constructs that will be further unwound into states. In the prestate elimination phase, prestates are removed, leaving only states in the graph; also, the edges are rearranged producing a directed graph called an *initial tableau*. The last phase removes from the tableau those states which contain inconsistencies (i.e. the constant **false**, **¬true**, or a formula and its negation) or do not have all the required successors.

We note that in order to fully capture the semantic nature of a coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$ $(\mathcal{I},\mathcal{U},\mathcal{N})$, the clauses in $\mathcal{U}$ and $\mathcal{N}$ must be included in every state of the resulting tableau. Instead of extending the tableau procedure for the next-time fragment of ATL, by explicitly adding those clauses to states, we make use of the existing α rule for the $\langle\langle\emptyset\rangle\rangle\,\Box$ operator given in the tableau procedure for full ATL. We define CL^+ to be the language of CL plus the $\langle\langle\emptyset\rangle\rangle\,\Box$ operator that is only allowed to occur positively in CL^+ formulae. The semantics of the $\langle\langle\emptyset\rangle\rangle\,\Box$ is defined in terms of a run:

DEFINITION 4.22
Let $\mathcal{F}=(\Sigma,\mathcal{S},s_0,d,\delta)$ be a CGF. A **run** in $\mathcal{F}$ is an infinite sequence $\lambda=s'_0,s'_1,\ldots,\ s'_i\in\mathcal{S}$ for all $i\geq 0$, where s'_{i+1} is a successor of s'_i. The indexes i, $i\geq 0$, in a sequence λ are called **positions**. Let $\lambda=s'_0,s'_1,\ldots,s'_i,\ldots,s'_j,\ldots$ be a run. We denote by $\lambda[i]=s'_i$ the i-th state in λ and by $\lambda[i,j]=s'_i,\ldots,s'_j$ the finite sequence that starts at s'_i and ends at s'_j. If $\lambda[0]=s$, then λ is called a ***s*-run**.

Intuitively, $\langle\langle\emptyset\rangle\rangle\,\Box\varphi$ means that, for all runs, φ always holds on them. Formally, a **strategy** $F_\emptyset$ **for** $\emptyset$ (or $\emptyset$-strategy) at a state s is given by $F_\emptyset(\{s\})\in D(\emptyset,s)$, i.e. $F_\emptyset(\{s\})$ is the $\emptyset-move$, $F_\emptyset(\{s\})=\sigma_\emptyset$. The **outcome of** $F_\emptyset$ **at state** $s\in\mathcal{S}$, denoted by $out(s,F_\emptyset)$ is the set of all runs λ such that $\lambda[i+1]\in out(\lambda[i],F_\emptyset(\lambda[i]))$, for all $i\geq 0$. Briefly, the outcome of $F_\emptyset$ at state s is a set consisting of every possible s-run. Finally, given a model $\mathcal{M}$, a state $s\in\mathcal{M}$, and a formula φ, $\langle\mathcal{M},s\rangle\models\langle\langle\emptyset\rangle\rangle\,\Box\varphi$ if, and only if, there exists an $\emptyset$-strategy $F_\emptyset$ such that $\langle\mathcal{M},\lambda[i]\rangle\models\varphi$ for all $\lambda\in out(s,F_\emptyset)$ and all positions $i\geq 0$. The definition of **positive coalition formula** is now extended to a formula of the form $[\mathcal{A}]\varphi$, where φ is a CL^+ formula. Negative coalition formulae and coalition formulae are defined as before. Note that formulae in the form of $\langle\langle\emptyset\rangle\rangle\,\Box$ always occur positively in the set of formulae used in the construction of the tableau for a coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$. Also, as it is clear from the procedure given below, the deletion rule for eventualities (formulae that hold at some future time of a run), which is part of the full tableau procedure, is not applied here and will not contribute to remove nodes from the tableau.

Before presenting the construction rules, we give two definitions that will be used later.

DEFINITION 4.23
Let Δ be a set of CL^+ formulae. We say that Δ is **downward saturated** if Δ satisfies the following two properties:

– If $\alpha \in \Delta$, then $\{\alpha_1, \dots, \alpha_n\} \subseteq \Delta$;
– If $\beta \in \Delta$, then $\beta_1 \in \Delta$, or ..., or $\beta_n \in \Delta$.

DEFINITION 4.24
Let Γ and Δ be sets of CL^+ formulae. We say that Δ is a **minimal downward saturated extension** of Γ if Δ satisfies the following three properties:

– $\Gamma \subseteq \Delta$;
– Δ is downward saturated;
– there is no downward saturated set Δ' such that $\Gamma \subseteq \Delta' \subset \Delta$.

Construction Phase: As mentioned, the construction phase builds a directed graph which contains states and prestates. States are downward saturated sets of formulae. Prestates are sets of formulae used to help the construction of the graph, in a similar fashion to the tableau construction for PTL [21]. There are two construction rules. The first, **SR**, creates states from prestates by saturation and the application of fix-point operations, that is, by applications of α and β rules. We note that the set of α rules also includes a rule for the $\langle\langle\emptyset\rangle\rangle\,\square$ operator. According to the α decomposition rules in [9], $\langle\langle\emptyset\rangle\rangle\,\square\varphi$ should be decomposed into φ and $\langle\langle\emptyset\rangle\rangle\bigcirc\langle\langle\emptyset\rangle\rangle\,\square\varphi$. The ATL formula $\langle\langle\emptyset\rangle\rangle\bigcirc\langle\langle\emptyset\rangle\rangle\,\square\varphi$ corresponds to the CL^+ formula $[\emptyset]\langle\langle\emptyset\rangle\rangle\,\square\varphi$, which explains the decomposition rule we give for $\langle\langle\emptyset\rangle\rangle\,\square\varphi$. The second rule, **Next**, creates prestates from states in order to ensure that coalition formulae are satisfied. There are two types of edges: double edges, from prestates to states; and labelled edges from states to prestates. Intuitively, the last type of edge represents the possible moves for the agents.

The construction starts by creating a prestate, which we call **initial prestate**, with a set of formulae Φ being tested for satisfiability. Then, the two construction rules are applied until no new states or prestates can be created. **SR** is the first of those rules.

SR Given a prestate Γ do:

(1) Create all minimal downward saturated extensions Δ of Γ as states;
(2) For each obtained state Δ, if Δ does not contain any coalition formulae, add $[\Sigma_\Phi]$**true** to Δ;
(3) Let Δ be a state created in steps (1) and (2). If there is already in the pretableau a state Δ' such that $\Delta = \Delta'$, add a double edge from Γ to Δ'; otherwise, add Δ and a double edge from Γ to Δ (i.e. $\Gamma \Longrightarrow \Delta$) to the pretableau.

In the following, we call **initial states** the states created from the first application of the rule **SR** in the construction of the tableau.

The second rule, **Next**, is applied to states in order to build a set of prestates, which correspond intuitively to possible successors of such states. In order to define the moves which are available to agents and coalition of agents in each state, an ordering over the coalition formulae in that state is defined. This ordering results in a list $\mathfrak{L}(\Delta)$, where each positive coalition formula precedes all negative coalition formulae. Intuitively, each index in this ordering refers to a possible move choice for each agent. The number of moves, at a state Δ, for each agent mentioned in a formula $\varphi \in \Delta$, is then given by the number of coalition formulae occurring in Δ, i.e. the size of the list $\mathfrak{L}(\Delta)$. We also note that, from the construction of a tableau, the list $\mathfrak{L}(\Delta)$ is never empty, as the formula $[\Sigma_\varphi]$**true** is included in the state Δ if there are no other coalition formulae in Δ.

Once the moves available to all agents are defined, they are combined into *move vectors*. A move vector labels one or more edges from a state to its successors, which are prestates in the tableau. The decision of which formulae will be included in the successor prestate Γ' of a state Δ by a move σ, is based on the *votes* of the agents. Suppose $[\mathcal{A}]\varphi \in \Delta$ and that $[\mathcal{A}]\varphi$ is the i-th formula in $\mathfrak{L}(\Delta)$. If all $a \in \mathcal{A}$ vote for φ, i.e. the corresponding action for agent a is i in σ, then φ is included in Γ'. For $\langle\mathcal{A}\rangle\varphi \in \Delta$, the decision whether φ is included in Γ' depends on the *collective vote* of the agents which are not in $\mathcal{A}$. We first present the **Next** rule and then show an example of how a collective vote is calculated. We say a state Δ is **consistent** if, and only if, $\{\neg\mathbf{true}, \mathbf{false}\} \cap \Delta = \emptyset$ and for all formulae φ, $\{\varphi, \neg\varphi\} \not\subseteq \Delta$. A state is **inconsistent** if, and only if, it is not consistent.

Next Given a consistent state Δ, do the following:

(1) Order linearly all positive and negative coalition formulae in Δ in such a way that the positive coalition formulae precede the negative coalition formulae. Let $\mathfrak{L}(\Delta)$ be the resulting list:

$$\mathfrak{L}(\Delta) = ([\mathcal{A}_0]\varphi_0, \ldots, [\mathcal{A}_{m-1}]\varphi_{m-1}, \langle\mathcal{A}'_0\rangle\psi_0, \ldots, \langle\mathcal{A}'_{l-1}\rangle\psi_{l-1})$$

and let $r_\Delta = |\mathfrak{L}(\Delta)| = m + l$. Denote by $D(\Delta) = \{0, \ldots, r_\Delta\}^{|\Sigma_\Phi|}$, the set of move vectors available at state Δ. For every $\sigma \in D(\Delta)$, let $N(\sigma) = \{i \mid \sigma_i \geq m\}$ be the set of agents voting for a negative formula in the particular move vector σ. Finally, let $neg(\sigma) = (\Sigma_{i \in N(\sigma)}(\sigma_i - m))$ mod l.

(2) For each $\sigma \in D(\Delta)$:

(a) create a prestate

$$\begin{aligned}\Gamma_\sigma = & \{\varphi_i \mid [\mathcal{A}_i]\varphi_i \in \Delta \text{ and } \sigma_a = i, \forall a \in \mathcal{A}_i\} \\ & \cup \{\psi_j \mid \langle\mathcal{A}'_j\rangle\psi_j \in \Delta, neg(\sigma) = j \text{ and } \Sigma_\Phi \setminus \mathcal{A}'_j \subseteq N(\sigma)\}\end{aligned}$$

If $\Gamma_\sigma = \emptyset$, let Γ_σ be $\{\mathbf{true}\}$.

(b) if Γ_σ is not already a prestate in the pretableau, add Γ_σ to the pretableau and connect Δ and Γ_σ by an edge labelled by σ; otherwise, just add an edge labelled by σ from Δ to the existing prestate Γ_σ (i.e. add $\Delta \xrightarrow{\sigma} \Gamma$).

Let $prestates(\Delta) = \{\Gamma \mid \Delta \xrightarrow{\sigma} \Gamma$ for some $\sigma \in D(\Delta)\}$. Let $\mathfrak{L}(\Delta)$ be the resulting list of ordered coalition formulae in Δ and $\varphi \in \mathfrak{L}(\Delta)$. We denote by $n(\varphi, \mathfrak{L}(\Delta))$ the position of a coalition formula φ in $\mathfrak{L}(\Delta)$; if $\mathfrak{L}(\Delta)$ is clear from the context, we write $n(\varphi)$ for short.

It is easy to see that the **Next** rule is sound with respect to the axiomatization given in Section 2.2. A prestate Γ_σ contains both positive coalition formulae $[\mathcal{A}]\varphi_\mathcal{A}$ and $[\mathcal{B}]\varphi_\mathcal{B}$ only if $\mathcal{A} \cap \mathcal{B} = \emptyset$, because there can be no $i \in \Sigma_\Phi$ such that $\sigma_i = n([\mathcal{A}]\varphi_\mathcal{A})$ and $\sigma_i = n([\mathcal{B}]\varphi_\mathcal{B})$ for $[\mathcal{A}]\varphi_\mathcal{A} \neq [\mathcal{B}]\varphi_\mathcal{B}$. Also, a prestate Γ_σ contains both coalition formulae $[\mathcal{A}]\varphi_\mathcal{A}$ and $\langle\mathcal{B}\rangle\varphi_\mathcal{B}$ only if $\mathcal{A} \subseteq \mathcal{B}$. If $\mathcal{A} \not\subseteq B$, then there is $\mathcal{A}' \subseteq A$ such that $\mathcal{A}' \subseteq \Sigma_\Phi \setminus \mathcal{B} \subseteq N(\sigma)$. However, all agents in $\mathcal{A}$ vote for positive formulae; therefore they cannot be a subset of $N(\sigma)$, which is the set of agents voting for negative formulae.

Let Δ be a state and $\langle\mathcal{A}\rangle\varphi \in \Delta$ be a negative coalition formula. As mentioned above, the decision whether φ is included in a prestate Γ created from Δ depends on the collective votes of the agents. Note that φ might be included in Γ even if the agents $a \in \Sigma_\Phi \setminus \mathcal{A}$ do not vote for $\langle\mathcal{A}\rangle\varphi$. For instance, let $\Sigma_\Phi = \{1,2,3,4\}$ be the set of agents occurring in the set of formulae Φ, Δ be a state, $\mathfrak{L}(\Delta) = ([1]p_1, \langle 2\rangle p_2, \langle 3\rangle p_3, \langle 4\rangle p_4)$ be the list of coalition formulae in Δ, and consider the move vector $(2,0,2,2)$. Agents in $\{1,3,4\}$ all vote for the negative formula $\langle 3\rangle p_3$, whose index is 2. The collective vote is given by $((2-1)+(2-1)+(2-1)) \mod 3 = 0$, that is, the agents collectively vote for the first negative coalition formula, $\langle 2\rangle p_2$. As $\Sigma_\Phi \setminus \{2\} \subseteq \{1,3,4\}$, then p_2 is included in the successor prestate.

Prestate elimination phase: in this phase, the prestates (and edges from and to it) are removed from the pretableau. Let $\mathcal{P}^\Phi$ be the pretableau obtained by applying the construction procedure to the initial prestate containing the set Φ. Let $states(\Gamma) = \{\Delta \mid \Gamma \Longrightarrow \Delta\}$, for any prestate Γ. The deletion rule is given below.

PR For every prestate Γ in $\mathcal{P}^\Phi$:

(1) remove Γ from $\mathcal{P}^\Phi$;
(2) for all states Δ in $\mathcal{P}^\Phi$ such that $\Delta \xrightarrow{\sigma} \Gamma$ and all states $\Delta' \in states(\Gamma)$ put $\Delta \xrightarrow{\sigma} \Delta'$.

The graph obtained from exhaustive application of **PR** to $\mathcal{P}^\Phi$ is the **initial tableau**, denoted by $\mathcal{T}_0^\Phi$.

State elimination phase: in this phase, states that cannot be satisfied in any model are removed from the tableau. There are essentially two reasons to remove a state Δ: Δ is inconsistent (as defined earlier in the text); or for some move $\sigma \in D(\Delta)$, there is no state Δ' such $\Delta \xrightarrow{\sigma} \Delta'$ is in the tableau. The deletion rules are applied non-deterministically, removing one state at every stage. We denote by $\mathcal{T}_{m+1}^\Phi$ the tableau obtained from $\mathcal{T}_m^\Phi$ by an application of one of the state elimination rules given below. Let $\mathcal{S}_m^\Phi$ be the set of states of the tableau $\mathcal{T}_m^\Phi$.

The elimination rules are defined as follows.

–**E1** If Δ is not consistent, obtain $\mathcal{T}_{m+1}^\Phi$ from $\mathcal{T}_m^\Phi$ by eliminating Δ, i.e. let $\mathcal{S}_{m+1}^\Phi = \mathcal{S}_m^\Phi \setminus \{\Delta\}$;

–**E2** If for some $\sigma \in D(\Delta)$, there is no Δ' such that $\Delta \xrightarrow{\sigma} \Delta'$, then obtain $\mathcal{T}_{m+1}^\Phi$ from $\mathcal{T}_m^\Phi$ by eliminating Δ, i.e. let $\mathcal{S}_{m+1}^\Phi = \mathcal{S}_m^\Phi \setminus \{\Delta\}$;

The elimination procedure consists of applying **E1** until all inconsistent states are removed. Then, the rule **E2** is applied until no states can be removed from the tableau. The resulting tableau, called **final tableau**, is denoted by $\mathcal{T}^\Phi$.

Definition 4.25
The final tableau $\mathcal{T}^\Phi$ is **open** if $\Phi \subseteq \Delta$ for some $\Delta \in \mathcal{S}^\Phi$. A tableau $\mathcal{T}_m^\Phi$, $m \geq 0$, is **closed** if $\Phi \not\subseteq \Delta$, for every $\Delta \in \mathcal{S}^\Phi$.

Theorem 4.26
Let Φ be a finite set of formulae in CL^+. The tableau construction for Φ terminates in time exponential in the size of Φ and Φ is unsatisfiable if, and only if, the final tableau for Φ, $\mathcal{T}^\Phi$, is closed.

Proof of Theorem 4.26. Termination and complexity of the tableau construction follows from the results in Section 4 in [9]. Soundness and completeness follow from Theorem 5.15 and Theorem 5.39 of [9], respectively. ■

Tableaux for coalition problems: recall that a derivation, as given in Definition 3.5, is a finite sequence $\mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_n$ of coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$ such that $\mathcal{C}_{i+1}$ is obtained from $\mathcal{C}_i$, $0 \leq i < n$, by an application of a resolution rule to premises in $\mathcal{C}_i$. For each $\mathcal{C}_i$, $0 \leq i \leq n$, we construct an initial tableau $\mathcal{T}_0^{\mathcal{C}_i}$, thereby obtaining a sequence $\mathcal{T}_0^{\mathcal{C}_0}, \mathcal{T}_0^{\mathcal{C}_1}, \mathcal{T}_0^{\mathcal{C}_2}, \ldots, \mathcal{T}_0^{\mathcal{C}_n}$. For each $\mathcal{C}_i$, $0 \leq i \leq n$, we denote by $\mathcal{T}_+^{\mathcal{C}_i}$ the tableau obtained from the initial tableau $\mathcal{T}_0^{\mathcal{C}_i}$ after the deletion rule **E1** has been exhaustively applied. We show that $\mathcal{T}_+^{\mathcal{C}_n}$ is closed if, and only if, $\mathcal{C}_n$ contains a contradiction. The proof is by induction on the number of nodes of the tableaux in the sequence $\mathcal{T}_+^{\mathcal{C}_0}, \mathcal{T}_+^{\mathcal{C}_1}, \mathcal{T}_+^{\mathcal{C}_2}, \ldots, \mathcal{T}_+^{\mathcal{C}_n}$.

Firstly, we define the set of disjunctions that might occur in a coalition problem in $\mathsf{DSNF_{CL}}$ $\mathcal{C} = (\mathcal{I}, \mathcal{U}, \mathcal{N})$. We denote by $\Pi_{\mathcal{C}}$ the set of propositional symbols occurring in $\mathcal{C}$, and by $\Lambda_{\mathcal{C}} = \Pi_{\mathcal{C}} \cup \{\neg p \mid p \in \Pi_{\mathcal{C}}\}$ the set of literals that might occur in $\mathcal{C}$. Let $\mathcal{D}_{\mathcal{C}}$ be $\{simp(\bigvee_{l \in \mathcal{M}} l) \mid \mathcal{M} \in 2^{\Lambda_{\mathcal{C}}}\} \setminus \{\mathbf{true}, \mathbf{false}\}$, where *simp* is defined by $simp(D \vee l \vee \neg l) = \mathbf{true}$ and $simp(D \vee \mathbf{true}) = \mathbf{true}$; in any other case, $simp(D) = D$, for any disjunction D. Thus, $\mathcal{D}_{\mathcal{C}}$ contains any (non trivial) disjunction that can be formed by either propositional symbols or their negations occurring in the coalition problem $\mathcal{C}$. For instance, if $\Pi_{\mathcal{C}} = \{p_1, p_2\}$, then $\mathcal{D}_{\mathcal{C}} = \{p_1, p_2, \neg p_1, \neg p_2, (p_1 \vee p_2), (p_1 \vee \neg p_2), (\neg p_1 \vee p_2), (\neg p_1 \vee \neg p_2)\}$. Let $\Theta_{\mathcal{C}}$ be the set $\{(D \vee \neg D) \mid D \in \mathcal{D}_{\mathcal{C}}\}$. In the following, we refer to $\Theta_{\mathcal{C}}$ as the *set of tautologies*.

The construction of a tableau for a coalition problem in $\mathsf{DSNF_{CL}}$ starts as follows. Let $\mathcal{C}_0 = (\mathcal{I}_0, \mathcal{U}_0, \mathcal{N}_0)$ be a coalition problem in $\mathsf{DSNF_{CL}}$. Let $\mathcal{C}_i = (\mathcal{I}_i, \mathcal{U}_i, \mathcal{N}_i)$ be a coalition problem in $\mathsf{DSNF_{CL}}$ in a derivation from $\mathcal{C}_0$. We construct the initial tableau $\mathcal{T}_0^{\mathcal{C}_i}$ for $\mathcal{C}_i$ from a prestate containing the following set of formulae:

$$\begin{array}{l} \{D \mid D \in \mathcal{I}_i\} \cup \\ \{\langle\langle \emptyset \rangle\rangle \Box D' \mid D' \in \mathcal{U}_i\} \cup \\ \{\langle\langle \emptyset \rangle\rangle \Box D'' \mid D'' \in \mathcal{N}_i\} \cup \\ \{\langle\langle \emptyset \rangle\rangle \Box D''' \mid D''' \in \Theta_{\mathcal{C}_i}\} \end{array}$$

The tautologies in $\Theta_{\mathcal{C}_i}$ are added in order to make available in the tableau all possible disjunctions that might occur in the set of clauses, to identify the premises used in applications of the resolution inference rules, and to deal with subformulae occurring in the scope of a coalition modality. By doing so, we can ensure that tableaux corresponding to coalition problems in a derivation will not grow in size. Also, after the deletion rule **E1** has been applied, every state in the tableau will contain a propositional symbol or its negation, that is, a maximally consistent set of literals. Moreover, every state will contain all disjunctions which are satisfied by that set of literals. Adding the tautologies to the initial set of formulae might increase the size of the resulting tableau and, therefore, affect the efficiency of the tableau procedure. However, we are not concerned with efficiency here, but with making available all information needed to relate the clauses used in a derivation by the resolution method with the states built in the corresponding tableaux. Obviously, as tautologies are satisfiable formulae, the resulting tableau will depend only on the satisfiability of the transformation of the coalition problem.

We note that global and coalition clauses in $\mathsf{DSNF_{CL}}$ are in the scope of the universal modality $\langle\langle \emptyset \rangle\rangle \Box$. This is needed in order to capture the semantics of coalition problems. The next lemma shows that if a clause is in the set of either global clauses, coalition clauses, or in the set of tautologies for a coalition problem $\mathcal{C}$, then it is in every state of the initial tableau $\mathcal{T}_0^{\mathcal{C}}$ for $\mathcal{C}$.

LEMMA 4.27
Let $\mathcal{C} = (\mathcal{I}, \mathcal{U}, \mathcal{N})$ be a coalition problem in $\mathsf{DSNF_{CL}}$. Let $\mathcal{P}^{\mathcal{C}}$ be the pretableau for $\mathcal{C}$, $\mathcal{S}^{\mathcal{C}}$ the set of states in $\mathcal{P}^{\mathcal{C}}$, and $\mathcal{R}^{\mathcal{C}}$ the set of prestates in $\mathcal{P}^{\mathcal{C}}$. If $\varphi \in \mathcal{U} \cup \mathcal{N} \cup \Theta_{\mathcal{C}}$, then the following holds:

1. $\varphi \in \Delta$, for all $\Delta \in \mathcal{S}^{\mathcal{C}}$;
2. $\langle\langle \emptyset \rangle\rangle \Box \varphi \in \Gamma$, for all $\Gamma \in \mathcal{R}^{\mathcal{C}}$.

PROOF OF LEMMA 4.27. The construction of the tableau follows alternate rounds of applications of the rules **SR** and **Next**.

(1) Assume that $\langle\langle\emptyset\rangle\rangle \Box\varphi$ is a formula in a prestate Γ of $\mathcal{P}^{\mathcal{C}}$. By an application of **SR**, the states generated from any prestate are downward saturated. More specifically, as this is a conjunctive formula, every state Δ generated from Γ contains φ and $[\emptyset]\langle\langle\emptyset\rangle\rangle \Box\varphi$. Thus, every state created from Γ contains φ.
(2) Assume that Δ is a state that contains $[\emptyset]\langle\langle\emptyset\rangle\rangle \Box\varphi$. Recall that by applying the **Next** rule, if Γ_σ is a successor prestate generated from a state which contains $[\mathcal{A}_p]\varphi_p$, then $\varphi_p \in \Gamma_\sigma$ if $\sigma_a = p$ for all $a \in \mathcal{A}$. As this condition holds vacuously for the empty coalition, every prestate generated from Δ contains $\langle\langle\emptyset\rangle\rangle \Box\varphi$.

By construction, $\langle\langle\emptyset\rangle\rangle \Box\varphi$, for all $\varphi \in \mathcal{U} \cup \mathcal{N} \cup \Theta_{\mathcal{C}}$, is one of the formulae of the initial prestate. Therefore, from (1) and (2), by induction, all clauses $\varphi \in \mathcal{U} \cup \mathcal{N} \cup \Theta_{\mathcal{C}}$ are in every state created during the construction phase. Also, from (1) and (2), by induction, $\langle\langle\emptyset\rangle\rangle \Box\varphi$ is in every prestate in $\mathcal{P}^{\mathcal{C}}$. ■

LEMMA 4.28
Let $\mathcal{C} = (\mathcal{I}, \mathcal{U}, \mathcal{N})$ be a coalition problem in $\mathsf{DSNF_{CL}}$. Let $\mathcal{T}_0^{\mathcal{C}}$ be the initial tableau for $\mathcal{C}$ and $\mathcal{S}_0^{\mathcal{C}}$ the set of states in $\mathcal{T}_0^{\mathcal{C}}$. If $\varphi \in \mathcal{U} \cup \mathcal{N} \cup \Theta_{\mathcal{C}}$, then $\varphi \in \Delta$, for all $\Delta \in \mathcal{S}_0^{\mathcal{C}}$.

PROOF OF LEMMA 4.28. From Lemma 4.27, if $\varphi \in \mathcal{U} \cup \mathcal{N} \cup \Theta_{\mathcal{C}}$, then φ is in all states in the pretableau $\mathcal{P}^{\mathcal{C}}$. After the construction phase, the rule **PR** only removes prestates. Thus, all the states in the initial tableau contain φ. ■

For technical reasons, we introduce some tautologies in the initial prestate during the construction of a tableau for a coalition problem in $\mathsf{DSNF_{CL}}$. Adding the set of tautologies has the effect that every state in the tableau contains every possible disjunction that can be built from propositional symbols (or their negations) which occur in a coalition problem. In particular, disjunctions in the form of $(l \vee \neg l)$, where l is a literal, are in every state of the tableau.

COROLLARY 4.29
Let $\mathcal{C} = (\mathcal{I}, \mathcal{U}, \mathcal{N})$ be a coalition problem in $\mathsf{DSNF_{CL}}$. Let $\mathcal{P}^{\mathcal{C}}$ be the pretableau for $\mathcal{C}$, $\mathcal{S}^{\mathcal{C}}$ the set of states in $\mathcal{P}^{\mathcal{C}}$, and $\mathcal{R}^{\mathcal{C}}$ the set of prestates in $\mathcal{P}^{\mathcal{C}}$. If $l \in \Lambda_{\mathcal{C}}$, then the following holds:

(1) $(l \vee \neg l) \in \Delta$, for all $\Delta \in \mathcal{S}^{\mathcal{C}}$;
(2) $\langle\langle\emptyset\rangle\rangle \Box(l \vee \neg l) \in \Gamma$, for all $\Gamma \in \mathcal{R}^{\mathcal{C}}$ and $l \in \Lambda_{\mathcal{C}}$.

PROOF OF COROLLARY 4.29. Immediate from Lemma 4.27 and the definitions of $\mathcal{D}_{\mathcal{C}}$ and $\Theta_{\mathcal{C}}$. ■

COROLLARY 4.30
Let $\mathcal{C} = (\mathcal{I}, \mathcal{U}, \mathcal{N})$ be a coalition problem in $\mathsf{DSNF_{CL}}$. Let $\mathcal{T}_0^{\mathcal{C}}$ be the initial tableau for $\mathcal{C}$ and $\mathcal{S}_0^{\mathcal{C}}$ the set of states in $\mathcal{T}_0^{\mathcal{C}}$. If $l \in \Lambda_{\mathcal{C}}$, then $(l \vee \neg l) \in \Delta$, for all $\Delta \in \mathcal{S}_0^{\mathcal{C}}$.

PROOF OF COROLLARY 4.30. Immediate from Lemma 4.28 and the definitions of $\mathcal{D}_{\mathcal{C}}$ and $\Theta_{\mathcal{C}}$. ■

As $\Theta_{\mathcal{C}}$ contains tautologies of the form $\langle\langle\emptyset\rangle\rangle \Box(p \vee \neg p)$, for every propositional symbol p occurring in $\mathcal{C}$, every state of the tableau contains p or its negation.

LEMMA 4.31
Let $\mathcal{C} = (\mathcal{I}, \mathcal{U}, \mathcal{N})$ be a coalition problem in $\mathsf{DSNF_{CL}}$, $\mathcal{T}_0^{\mathcal{C}}$ be the initial tableau for $\mathcal{C}$, and $\mathcal{S}_0^{\mathcal{C}}$ the set of states in $\mathcal{T}_0^{\mathcal{C}}$. If $p \in \Pi_{\mathcal{C}}$, then either $p \in \Delta$ or $\neg p \in \Delta$, for all $\Delta \in \mathcal{S}_0^{\mathcal{C}}$.

Proof of Lemma 4.31. By definition, p and $\neg p$ are both in $\Lambda_{\mathcal{C}}$. By Corollary 4.30, if $l \in \Lambda$, then $(l \vee \neg l) \in \Delta$, for all $\Delta \in \mathcal{S}_0^{\mathcal{C}}$. Because states are downward saturated, either $l \in \Delta$ or $\neg l \in \Delta$, for all $\Delta \in \mathcal{S}_0^{\mathcal{C}}$. ■

Moreover, after the deletion rule **E1** has been applied, every state in the tableau contains a maximal consistent set of literals.

Lemma 4.32 (Tautologies)
Let $\mathcal{T}_+^{\mathcal{C}}$ be the tableau for a coalition problem in $\mathsf{DSNF_{CL}}$ $\mathcal{C} = (\mathcal{I}, \mathcal{U}, \mathcal{N})$ and $\mathcal{S}_+^{\mathcal{C}}$ the set of states in $\mathcal{T}_+^{\mathcal{C}}$. Then every state of $\mathcal{T}_+^{\mathcal{C}}$ contains a maximal consistent set of literals occurring in $\mathcal{C}$.

Proof of Lemma 4.32. By Lemma 4.31, if $l \in \Lambda_{\mathcal{C}}$, then either l or $\neg l$ is in Δ, for all $\Delta \subseteq \mathcal{S}_0^{\mathcal{C}}$, where $\mathcal{S}_0^{\mathcal{C}}$ is the set of states in $\mathcal{T}_0^{\mathcal{C}}$. States containing both l and $\neg l$ for some literal $l \in \Lambda_{\mathcal{C}}$ are deleted by **E1**. Therefore, for all $\Delta \in \mathcal{S}_+^{\mathcal{C}}$, Δ contains a maximal consistent set of literals. ■

Adding the tautologies also helps to show that the tableaux in the sequence corresponding to a derivation do not increase in size. The conclusion of the resolution rules are disjunctions that hold in the initial states (**IRES1**), in all states (**GRES1**, **RW1-2**), or in a particular set of states (**CRES1-4**). The construction of the tableau requires that β rules are applied to those disjunctions. In general, applications of β rules to disjunctions have the effect of multiplying the number of successor states. However, applying β rules to the set of tautologies we introduced in the prestates create all possible states as successors; thus, further applications of β rules to other disjunctions can only have the effect of creating states which do not satisfy those other disjunctions. In the following, we assume that α and β rules are applied in a particular order. This is not important, in general, as the resulting sets of minimal downward saturated formulae is the same independent of which order those rules are applied. However, the assumption of a particular order in the application of α and β rules simplifies the proof that the size of the tableau corresponding to steps in the derivation does not increase, that is, that we have $|\mathcal{T}_+^{\mathcal{C}_0}| \geq |\mathcal{T}_+^{\mathcal{C}_1}| \geq \ldots \geq |\mathcal{T}_+^{\mathcal{C}_n}|$.

Let Γ be a prestate and *states*(Γ) be the set of states created from Γ by an application of the rule **SR**. We denote by *cons*$(\Gamma) \subseteq$ *states*(Γ) the set of consistent states created from Γ, that is, *cons*$(\Gamma) = \{\Delta \mid \Delta \in$ *states*(Γ) and Δ is consistent$\}$.

Lemma 4.33
Let $\mathcal{C}_i = (\mathcal{I}_i, \mathcal{U}_i, \mathcal{N}_i)$ be a coalition problem in $\mathsf{DSNF_{CL}}$. Let $\mathcal{C}_{i+1}$ be the coalition problem in $\mathsf{DSNF_{CL}}$ obtained from $\mathcal{C}_i$ by adding a propositional disjunction φ to the initial set of clauses, that is, $\mathcal{C}_{i+1} = (\mathcal{I}_i \cup \{\varphi\}, \mathcal{U}_i, \mathcal{N}_i)$, where $\Lambda_{\mathcal{C}_i} = \Lambda_{\mathcal{C}_{i+1}}$. Let $\mathcal{S}_+^{\mathcal{C}_i}$ and $\mathcal{S}_+^{\mathcal{C}_{i+1}}$ be the set of states in $\mathcal{T}_+^{\mathcal{C}_i}$ and $\mathcal{T}_+^{\mathcal{C}_{i+1}}$, respectively. Then $|\mathcal{T}_+^{\mathcal{C}_{i+1}}| \leq |\mathcal{T}_+^{\mathcal{C}_i}|$ and for all $\Delta^{\mathcal{C}_{i+1}} \in \mathcal{S}_+^{\mathcal{C}_{i+1}}$ there is $\Delta^{\mathcal{C}_i} \in \mathcal{S}_+^{\mathcal{C}_i}$, such that $\Delta^{\mathcal{C}_i} \subseteq \Delta^{\mathcal{C}_{i+1}}$.

Proof of Lemma 4.33. Construct the pretableau $\mathcal{P}^{\mathcal{C}_i}$ for $\mathcal{C}_i$. Let $\Gamma_0^{\mathcal{C}_i}$ be the initial prestate and let *states*$(\Gamma_0^{\mathcal{C}_i}) = \{\Delta_0^{\mathcal{C}_i}, \ldots, \Delta_n^{\mathcal{C}_i}\}$, for some $n \in \mathbb{N}$, be the set of states created from $\Gamma_0^{\mathcal{C}_i}$ by an application of **SR**. Furthermore, let *cons*$(\Gamma_0^{\mathcal{C}_i}) \subseteq$ *states*$(\Gamma_0^{\mathcal{C}_i})$ be the set of consistent states in *states*$(\Gamma_0^{\mathcal{C}_i})$.

We now construct the pretableau $\mathcal{P}^{\mathcal{C}_{i+1}}$ for $\mathcal{C}_{i+1}$. Let $\Gamma_0^{\mathcal{C}_{i+1}}$ be the initial prestate of $\mathcal{P}^{\mathcal{C}_{i+1}}$. Note that $\Gamma_0^{\mathcal{C}_{i+1}} = \Gamma_0^{\mathcal{C}_i} \cup \{\varphi\}$, because $\mathcal{I}_{i+1} = \mathcal{I}_i \cup \{\varphi\}$, $\mathcal{U}_{i+1} = \mathcal{U}_i$, $\mathcal{N}_{i+1} = \mathcal{N}_i$, and $\Theta_{\mathcal{C}_{i+1}} = \Theta_{\mathcal{C}_i}$. Start the construction by first applying all the α and β rules to those formulae in $\Gamma_0^{\mathcal{C}_{i+1}}$ that are also in $\Gamma_0^{\mathcal{C}_i}$. Because states are downward saturated and $\varphi \in \Gamma_0^{\mathcal{C}_{i+1}}$, we also add φ to the sets created so far. At this point of the construction, we have generated a set $\{\Delta_0^{\mathcal{C}_{i+1}}, \ldots, \Delta_n^{\mathcal{C}_{i+1}}\}$, where every $\Delta_k^{\mathcal{C}_{i+1}} = \Delta_k^{\mathcal{C}_i} \cup \{\varphi\}$, for all $0 \leq k \leq n$. Note that the number of sets of formulae created so far is exactly the same as the number of states created from $\Gamma_0^{\mathcal{C}_i}$, as the same rules were applied in the same order and we only added a formula φ to those sets. Take $\Delta_k^{\mathcal{C}_i}$ in $states(\Gamma_0^{\mathcal{C}_i})$. If $\Delta_k^{\mathcal{C}_i} \notin cons(\Gamma_0^{\mathcal{C}_i})$, then $\Delta_k^{\mathcal{C}_{i+1}}$ is not consistent either and any attempt to expand $\Delta_k^{\mathcal{C}_{i+1}}$ will result in an inconsistent state that will be later removed by **E1**. Assume $\Delta_k^{\mathcal{C}_i} \in cons(\Gamma_0^{\mathcal{C}_i})$. As φ is a disjunction, by Lemma 4.27, $\Delta_k^{\mathcal{C}_i}$ contains $(\varphi \vee \neg\varphi)$, which is a formula in $\Theta_{\mathcal{C}_i}$. Because states are downward saturated, $\Delta_k^{\mathcal{C}_i}$ contains either φ or $\neg\varphi$. Therefore, sets $\Delta_k^{\mathcal{C}_{i+1}} = \Delta_k^{\mathcal{C}_i} \cup \{\varphi\}$ containing both φ and $\neg\varphi$ will be later eliminated by rule **E1**. Assume $\Delta_k^{\mathcal{C}_{i+1}}$, for some $0 \leq k \leq n$, contains φ, but not $\neg\varphi$. We apply the β rule to φ and try to expand $\Delta_k^{\mathcal{C}_{i+1}}$. We note that, in fact, the β rule is applied to all states, not only those which are consistent, but again whatever way we try to expand an inconsistent state will result in inconsistent states that will be later removed by **E1**. Let φ be $l_1 \vee \ldots \vee l_m$, for some $m \in \mathbb{N}$. If $m = 0$, then φ is the empty disjunction (**false**) and no more rules are actually applied. Therefore, no other states are created from $\Gamma_0^{\mathcal{C}_{i+1}}$ (as a matter of fact, the resulting tableau is closed, as every initial state contains **false** and is eliminated by **E1**). If $m > 0$, we apply the β rule to φ. By Corollary 4.29, every state $\Delta_k^{\mathcal{C}_{i+1}}$ contains $l \vee \neg l$, for all literals in $\Lambda_{\mathcal{C}_{i+1}} = \Lambda_{\mathcal{C}_i}$. By construction, every state is downward saturated. Therefore, every state contains l_j or $\neg l_j$, for $1 \leq j \leq m$. Choose any l_j, $0 \leq j \leq m$, and try to expand $\Delta_k^{\mathcal{C}_{i+1}}$. If $\Delta_k^{\mathcal{C}_{i+1}}$ already contains l_j, we do not need to add anything to the state and we have that $\Delta_k^{\mathcal{C}_{i+1}} = \Delta_k^{\mathcal{C}_i}$. If $\Delta_k^{\mathcal{C}_{i+1}}$ does not contain l_j, then it must contain $\neg l_j$; thus, adding l_j results in an inconsistent state which will be later removed by an application of rule **E1**. Therefore, the application of the β rule to φ in a state $\Delta_k^{\mathcal{C}_{i+1}}$ can only contribute to create new states that contain inconsistencies. That is, $cons(\Gamma_0^{\mathcal{C}_{i+1}}) \subseteq cons(\Gamma_0^{\mathcal{C}_i})$. Moreover, for all $\Delta_k^{\mathcal{C}_{i+1}} \in cons(\Gamma_0^{\mathcal{C}_{i+1}})$, there is $\Delta_k^{\mathcal{C}_i} \in cons(\Gamma_0^{\mathcal{C}_i})$, such that $\Delta_k^{\mathcal{C}_i} \subseteq \Delta_k^{\mathcal{C}_{i+1}}$.

Overall, the application of **SR** to $\Gamma_0^{\mathcal{C}_{i+1}}$ results in a set $states(\Gamma_0^{\mathcal{C}_{i+1}})$ with $|states(\Gamma_0^{\mathcal{C}_{i+1}})| \geq |states(\Gamma_0^{\mathcal{C}_i})|$. However, for the set $cons(\Gamma_0^{\mathcal{C}_{i+1}}) \subseteq states(\Gamma_0^{\mathcal{C}_{i+1}})$ of all consistent states, we have that $|cons(\Gamma_0^{\mathcal{C}_{i+1}})| \leq |cons(\Gamma_0^{\mathcal{C}_i})|$.

As φ is in $\mathcal{I}_{i+1}$, then φ is in the initial prestate and in all initial states of the pretableau $\mathcal{P}^{\mathcal{C}_{i+1}}$. However, as φ is a propositional clause, the constructions of $\mathcal{P}^{\mathcal{C}_{i+1}}$ and $\mathcal{P}^{\mathcal{C}_i}$ differs only at the first application of **SR**. The applications of **Next** and **SR** that follow remain the same. Firstly, the application of the **Next** rule depends only on clauses that are in the scope of $[\mathcal{A}]$ for some coalition $\mathcal{A}$. Secondly, further applications of **SR** depend on prestates created by **Next**, which is not affected by the inclusion of φ in the initial states. Therefore, for the remaining of the construction, we have that

$$\bigcup_{\Gamma^{\mathcal{C}_i} \in \mathcal{P}^{\mathcal{C}_i} \setminus \Gamma_0^{\mathcal{C}_i}} states(\Gamma^{\mathcal{C}_i}) = \bigcup_{\Gamma^{\mathcal{C}_{i+1}} \in \mathcal{P}^{\mathcal{C}_{i+1}} \setminus \Gamma_0^{\mathcal{C}_{i+1}}} states(\Gamma^{\mathcal{C}_{i+1}}).$$

Obviously, the sets of consistent states created from prestates in $\Gamma^{\mathcal{C}_i} \in \mathcal{P}^{\mathcal{C}_i} \setminus \Gamma_0^{\mathcal{C}_i}$ and $\Gamma^{\mathcal{C}_{i+1}} \in \mathcal{P}^{\mathcal{C}_{i+1}} \setminus \Gamma_0^{\mathcal{C}_{i+1}}$ are also the same in $\mathcal{T}_+^{\mathcal{C}_i}$ and $\mathcal{T}_+^{\mathcal{C}_{i+1}}$. As the deletion rule **PR** only removes prestates and because the remainder of the construction of $\mathcal{P}^{\mathcal{C}_{i+1}}$ is exactly as in the construction of $\mathcal{P}^{\mathcal{C}_i}$, after exhaustively applying **E1**, the number of states in $\mathcal{T}_+^{\mathcal{C}_{i+1}}$ cannot be greater than the number of states in $\mathcal{T}_+^{\mathcal{C}_i}$. Thus, $|\mathcal{T}_+^{\mathcal{C}_{i+1}}| \leq |\mathcal{T}_+^{\mathcal{C}_i}|$. As $\mathcal{S}_+^{\mathcal{C}_i} = \bigcup_{\Gamma \in \mathcal{P}^{\mathcal{C}_i}} cons(\Gamma)$ and $\mathcal{S}_+^{\mathcal{C}_{i+1}} = \bigcup_{\Gamma^{\mathcal{C}_{i+1}} \in \mathcal{P}^{\mathcal{C}_{i+1}}} cons(\Gamma^{\mathcal{C}_{i+1}})$, we have that for all $\Delta^{\mathcal{C}_{i+1}} \in \mathcal{S}_+^{\mathcal{C}_{i+1}}$ there is $\Delta^{\mathcal{C}_i} \in \mathcal{S}_+^{\mathcal{C}_i}$, such that $\Delta^{\mathcal{C}_i} \subseteq \Delta^{\mathcal{C}_{i+1}}$. ■

LEMMA 4.34
Let $\mathcal{C}_i = (\mathcal{I}_i, \mathcal{U}_i, \mathcal{N}_i)$ be a coalition problem in $\mathsf{DSNF_{CL}}$. Let $\mathcal{C}_{i+1}$ be the coalition problem in $\mathsf{DSNF_{CL}}$ obtained from $\mathcal{C}_i$ by adding a propositional disjunction φ to the global set of clauses,

that is, $\mathcal{C}_{i+1} = (\mathcal{I}_i, \mathcal{U}_i \cup \{\varphi\}, \mathcal{N}_i)$, where $\Lambda_{\mathcal{C}_i} = \Lambda_{\mathcal{C}_{i+1}}$. Let $\mathcal{S}_+^{\mathcal{C}_i}$ and $\mathcal{S}_+^{\mathcal{C}_{i+1}}$ be the set of states in $\mathcal{T}_+^{\mathcal{C}_i}$ and $\mathcal{T}_+^{\mathcal{C}_{i+1}}$, respectively. Then $|\mathcal{T}_+^{\mathcal{C}_{i+1}}| \leq |\mathcal{T}_+^{\mathcal{C}_i}|$ and for all $\Delta^{\mathcal{C}_{i+1}} \in \mathcal{S}_+^{\mathcal{C}_{i+1}}$ there is $\Delta^{\mathcal{C}_i} \in \mathcal{S}_+^{\mathcal{C}_i}$, such that $\Delta^{\mathcal{C}_i} \subseteq \Delta^{\mathcal{C}_{i+1}}$.

Proof of Lemma 4.34. Construct the pretableau $\mathcal{P}^{\mathcal{C}_i}$ for $\mathcal{C}_i$. Let $\Gamma_0^{\mathcal{C}_i}$ be the initial prestate in $\mathcal{P}^{\mathcal{C}_i}$ and let $states(\Gamma_0^{\mathcal{C}_i}) = \{\Delta_0^{\mathcal{C}_i}, \ldots, \Delta_n^{\mathcal{C}_i}\}$, for some $n \in \mathbb{N}$, be the set of states created from $\Gamma_0^{\mathcal{C}_i}$ by an application of **SR**. Furthermore, let $cons(\Gamma_0^{\mathcal{C}_i}) \subseteq states(\Gamma_0^{\mathcal{C}_i})$ be the set of consistent states in $states(\Gamma_0^{\mathcal{C}_i})$.

We now construct the pretableau $\mathcal{P}^{\mathcal{C}_{i+1}}$ for $\mathcal{C}_{i+1}$. Let $\Gamma_0^{\mathcal{C}_{i+1}}$ be the initial prestate of $\mathcal{P}^{\mathcal{C}_{i+1}}$. Note that $\Gamma_0^{\mathcal{C}_{i+1}} = \Gamma_0^{\mathcal{C}_i} \cup \{\langle\langle\emptyset\rangle\rangle \Box\varphi\}$, because $\mathcal{I}_{i+1} = \mathcal{I}_i, \mathcal{U}_{i+1} = \mathcal{U}_i \cup \{\varphi\}, \mathcal{N}_{i+1} = \mathcal{N}_i$, and $\Theta_{\mathcal{C}_{i+1}} = \Theta_{\mathcal{C}_i}$. Start the construction by first applying all the α and β rules to the formulae in $\Gamma_0^{\mathcal{C}_{i+1}}$ which are also in $\Gamma_0^{\mathcal{C}_i}$. Because states are downward saturated and $\langle\langle\emptyset\rangle\rangle \Box\varphi \in \Gamma_0^{\mathcal{C}_{i+1}}$, we also add $\langle\langle\emptyset\rangle\rangle \Box\varphi$, φ, and $[\emptyset]\langle\langle\emptyset\rangle\rangle \Box\varphi$ to the sets of formulae created so far. At this point of the construction, we have generated a set $\{\Delta_0^{\mathcal{C}_{i+1}}, \ldots, \Delta_n^{\mathcal{C}_{i+1}}\}$, where every $\Delta_k^{\mathcal{C}_{i+1}} = \Delta_k^{\mathcal{C}_i} \cup \{\langle\langle\emptyset\rangle\rangle \Box\varphi, \varphi, [\emptyset]\langle\langle\emptyset\rangle\rangle \Box\varphi\}$, for all $0 \leq k \leq n$. Note that the number of sets of formulae created so far is exactly the same as the number of states created from $\Gamma_0^{\mathcal{C}_i}$, as the same rules were applied in the same order and we only added formulae to those sets. Take $\Delta_k^{\mathcal{C}_i}$ in $states(\Gamma_0^{\mathcal{C}_i})$. If $\Delta_k^{\mathcal{C}_i} \notin cons(\Gamma_0^{\mathcal{C}_i})$, then $\Delta_k^{\mathcal{C}_{i+1}}$ is not consistent either and any attempt to expand $\Delta_k^{\mathcal{C}_{i+1}}$ will result in an inconsistent state that will be later removed by **E1**. Assume $\Delta_k^{\mathcal{C}_i} \in cons(\Gamma_0^{\mathcal{C}_i})$. As φ is a disjunction, by Lemma 4.27, $\Delta_k^{\mathcal{C}_i}$ already contains either $\varphi \vee \neg\varphi$, which is a formula in $\Theta_{\mathcal{C}_i}$. By construction, every state is downward saturated. Therefore, $\Delta_k^{\mathcal{C}_i}$ contains φ or $\neg\varphi$. Therefore, sets $\Delta_k^{\mathcal{C}_{i+1}} = \Delta_k^{\mathcal{C}_i} \cup \{\langle\langle\emptyset\rangle\rangle \Box\varphi, \varphi, [\emptyset]\langle\langle\emptyset\rangle\rangle \Box\varphi\}$ containing both φ and $\neg\varphi$ will be later eliminated by rule **E1**. Assume $\Delta_k^{\mathcal{C}_{i+1}}$, for some $0 \leq k \leq n$, contains φ, but not $\neg\varphi$. We apply the β rule to φ and try to expand $\Delta_k^{\mathcal{C}_{i+1}}$. We note that, in fact, the β rule is applied to all states, but whatever way we try to expand an inconsistent state will result in inconsistent states that will be later removed by **E1**. Let φ be $l_1 \vee \ldots \vee l_m$, for some $m \in \mathbb{N}$. If $m = 0$, then φ is the empty disjunction and no more rules are actually applied. Therefore, no other states are created from $\Gamma_0^{\mathcal{C}_{i+1}}$ (as a matter of fact, the resulting tableau is closed, as every initial state contains **false** and is eliminated by **E1**). If $m > 0$, we apply the β rule to φ. By Corollary 4.29, every state $\Delta_k^{\mathcal{C}_{i+1}}$ contains $l \vee \neg l$, for all literals in $\Lambda_{\mathcal{C}_{i+1}} = \Lambda_{\mathcal{C}_i}$. By construction, every state is downward saturated. Therefore, every state contains l_j or $\neg l_j$, for $1 \leq j \leq m$. Choose any l_j, $0 \leq j \leq m$, and try to expand $\Delta_k^{\mathcal{C}_{i+1}}$. If $\Delta_k^{\mathcal{C}_{i+1}}$ contains l_j, we do not need to add l_j. If $\Delta_k^{\mathcal{C}_{i+1}}$ does not contain l_j, then it must contain $\neg l_j$; thus, adding l_j results in an inconsistent state which will be later removed by an application of rule **E1**. Therefore, the application of the β rule to φ at the initial prestate can only contribute to create states that contain inconsistencies. That is, $|cons(\Gamma_0^{\mathcal{C}_{i+1}})| \leq |cons(\Gamma_0^{\mathcal{C}_i})|$. Moreover, for all $\Delta_k^{\mathcal{C}_{i+1}} \in cons(\Gamma_0^{\mathcal{C}_{i+1}})$, there is $\Delta_k^{\mathcal{C}_i} \in cons(\Gamma_0^{\mathcal{C}_i})$, such that $\Delta_k^{\mathcal{C}_i} \subseteq \Delta_k^{\mathcal{C}_{i+1}} = \Delta_k^{\mathcal{C}_i} \cup \{\langle\langle\emptyset\rangle\rangle \Box\varphi, \varphi, [\emptyset]\langle\langle\emptyset\rangle\rangle \Box\varphi\}$.

Overall, the application of **SR** to $\Gamma_0^{\mathcal{C}_{i+1}}$ results in a set $states(\Gamma_0^{\mathcal{C}_{i+1}})$ with $|states(\Gamma_0^{\mathcal{C}_{i+1}})| \geq |states(\Gamma_0^{\mathcal{C}_i})|$. However, for the set $cons(\Gamma_0^{\mathcal{C}_{i+1}}) \subseteq states(\Gamma_0^{\mathcal{C}_{i+1}})$ of all consistent states, we have that $|cons(\Gamma_0^{\mathcal{C}_{i+1}})| \leq |cons(\Gamma_0^{\mathcal{C}_i})|$.

As $\mathcal{N}_{i+1} = \mathcal{N}_i$, the set of prestates created from a state $\Delta^{\mathcal{C}_{i+1}} \in \mathcal{P}^{\mathcal{C}_{i+1}}$ is like the set of prestates created from $\Delta^{\mathcal{C}_i} \in \mathcal{P}^{\mathcal{C}_i}$, except that we add $\langle\langle\emptyset\rangle\rangle \Box\varphi$ to the formulae used in the construction of the set of successor prestates (by an application of the rule **Next** to $[\emptyset]\langle\langle\emptyset\rangle\rangle \Box\varphi \in \Delta^{\mathcal{C}_{i+1}}$). When the rule **SR** is applied to such a prestate, as φ is in the scope of $\langle\langle\emptyset\rangle\rangle \Box$, φ is added to all created states. By reasoning as above, the addition of φ to a state in $\mathcal{P}^{\mathcal{C}_{i+1}}$ can only contribute to create states that contain inconsistencies and that will be later removed by applications of the rule **E1**. Therefore, in this step of the construction we are not adding any consistent states either. Therefore, for the remaining of the construction, for all $\Delta_k^{\mathcal{C}_{i+1}} \in cons(\Gamma^{\mathcal{C}_{i+1}})$, with $\Gamma^{\mathcal{C}_{i+1}} \in \mathcal{P}^{\mathcal{C}_{i+1}} \setminus \Gamma_0^{\mathcal{C}_{i+1}}$, there is $\Delta_k^{\mathcal{C}_i} \in cons(\Gamma^{\mathcal{C}_i})$ with $\Gamma^{\mathcal{C}_i} \in \mathcal{P}^{\mathcal{C}_i} \setminus \Gamma_0^{\mathcal{C}_i}$, such that $\Delta_k^{\mathcal{C}_i} \subseteq \Delta_k^{\mathcal{C}_{i+1}}$.

By induction on the steps of the construction, all added states are inconsistent. As **PR** only removes prestates, after exhaustively applying **E1**, the number of states in $\mathcal{T}_+^{\mathcal{C}_{i+1}}$ cannot be greater than the number of states in $\mathcal{T}_+^{\mathcal{C}_i}$. Thus, $|\mathcal{T}_+^{\mathcal{C}_{i+1}}| \leq |\mathcal{T}_+^{\mathcal{C}_i}|$. As $\mathcal{S}_+^{\mathcal{C}_i} = \bigcup_{\Gamma \in \mathcal{P}^{\mathcal{C}_i}} cons(\Gamma)$ and $\mathcal{S}_+^{\mathcal{C}_{i+1}} = \bigcup_{\Gamma^{\mathcal{C}_{i+1}} \in \mathcal{P}^{\mathcal{C}_{i+1}}} cons(\Gamma^{\mathcal{C}_{i+1}})$, we have that for all $\Delta^{\mathcal{C}_{i+1}} \in \mathcal{S}_+^{\mathcal{C}_{i+1}}$ there is $\Delta^{\mathcal{C}_i} \in \mathcal{S}_+^{\mathcal{C}_i}$, such that $\Delta^{\mathcal{C}_i} \subseteq \Delta^{\mathcal{C}_{i+1}}$. ∎

The next lemma shows that the right-hand side of a coalition formula holds where the left-hand side holds. We need this in order to identify the sets of clauses which contribute to finding a contradiction.

Lemma 4.35

Let $\mathcal{C} = (\mathcal{I}, \mathcal{U}, \mathcal{N})$ be a coalition problem in $\mathsf{DSNF_{CL}}$ and $C \Rightarrow D$ be a clause in $\mathcal{N}$, where $C = l_1 \wedge \ldots \wedge l_n$, for some $n \geq 0$. Let $\mathcal{T}^{\mathcal{C}}$ be the tableau for $\mathcal{C}$ and Δ a state in $\mathcal{T}_+^{\mathcal{C}}$. If $\{l_1, \ldots, l_n\} \subseteq \Delta$, then $D \in \Delta$.

Proof of Lemma 4.35. If $C \Rightarrow D$ is in $\mathcal{N}$, then by Lemma 4.28, $C \Rightarrow D$ is in every state of $\mathcal{T}^{\mathcal{C}}$. If $n = 0$, then C is the empty conjunction (**true**). Because Δ is downward saturated, it must contain either $\neg$**true** or D. As states containing $\neg$**true** are removed by applications of **E1**, Δ must contain D. If $n > 0$, assume $\{l_0, \ldots, l_n\} \subseteq \Delta$. As states are downward saturated, by applications of the β rule to $C \Rightarrow D$, every state contains either a literal in $\{\neg l_1, \ldots, \neg l_n\}$ or D. If for any l_j, $0 \leq j \leq n$, we had that $l_j \in \Delta$, then Δ would be inconsistent and, therefore, Δ would have been removed from the tableau $\mathcal{T}_+^{\mathcal{C}}$. Therefore, as $\Delta \in \mathcal{T}_+^{\mathcal{C}}$, we have that $D \in \Delta$. ∎

Note that if D is the right-hand side of any other coalition clause than $C \Rightarrow D$, then D might also occur in states where none of the literals in C is satisfied. The lemma above shows that applications of the rule **SR** to coalition clauses do not increase the number of states during the construction phase. The next lemma shows that the size of the tableaux in the sequence corresponding to a derivation does not increase by adding implications to the set of coalition clauses.

Lemma 4.36

Let $\mathcal{C}_i = (\mathcal{I}_i, \mathcal{U}_i, \mathcal{N}_i)$ be a coalition problem in $\mathsf{DSNF_{CL}}$. Let $\mathcal{C}_{i+1}$ be the coalition problem in $\mathsf{DSNF_{CL}}$ obtained from $\mathcal{C}_i$ by adding a coalition clause $\varphi \Rightarrow \psi$ to the coalition set of clauses, that is, $\mathcal{C}_{i+1} = (\mathcal{I}_i, \mathcal{U}_i, \mathcal{N}_i \cup \{\varphi \Rightarrow \psi\})$, where $\Lambda_{\mathcal{C}_i} = \Lambda_{\mathcal{C}_{i+1}}$ and where $\Sigma_{\mathcal{C}_i} = \Sigma_{\mathcal{C}_{i+1}}$. Let $\mathcal{S}_+^{\mathcal{C}_i}$ and $\mathcal{S}_+^{\mathcal{C}_{i+1}}$ be the set of states in $\mathcal{T}_+^{\mathcal{C}_i}$ and $\mathcal{T}_0^{\mathcal{C}_{i+1}}$, respectively. Then $|\mathcal{T}_+^{\mathcal{C}_{i+1}}| \leq |\mathcal{T}_+^{\mathcal{C}_i}|$ and for all $\Delta^{\mathcal{C}_{i+1}} \in \mathcal{S}_+^{\mathcal{C}_{i+1}}$ there is $\Delta^{\mathcal{C}_i} \in \mathcal{S}_+^{\mathcal{C}_i}$, such that $\Delta^{\mathcal{C}_i} \subseteq \Delta^{\mathcal{C}_{i+1}}$.

Proof of Lemma 4.36. Construct the pretableau $\mathcal{P}^{\mathcal{C}_i}$ for $\mathcal{C}_i$. Let $\Gamma_0^{\mathcal{C}_i}$ be the initial prestate and let $states(\Gamma_0^{\mathcal{C}_i}) = \{\Delta_0^{\mathcal{C}_i}, \ldots, \Delta_n^{\mathcal{C}_i}\}$, for some $n \in \mathbb{N}$, be the set of states created from $\Gamma_0^{\mathcal{C}_i}$ by an application of **SR**. Let $cons(\Gamma_0^{\mathcal{C}_i}) \subseteq states(\Gamma_0^{\mathcal{C}_i})$ be the set of consistent states in $states(\Gamma_0^{\mathcal{C}_i})$.

We now construct the pretableau $\mathcal{P}^{\mathcal{C}_{i+1}}$ for $\mathcal{C}_{i+1}$. Let $\Gamma_0^{\mathcal{C}_{i+1}}$ be the initial prestate of $\mathcal{P}^{\mathcal{C}_{i+1}}$. Note that $\Gamma_0^{\mathcal{C}_{i+1}} = \Gamma_0^{\mathcal{C}_i} \cup \{\langle\langle\emptyset\rangle\rangle \Box(\varphi \Rightarrow \psi)\}$, because $\mathcal{I}_{i+1} = \mathcal{I}_i$, $\mathcal{U}_{i+1} = \mathcal{U}_i$, $\mathcal{N}_{i+1} = \mathcal{N}_i \cup \{\varphi \Rightarrow \psi\}$, and $\Theta_{\mathcal{C}_{i+1}} = \Theta_{\mathcal{C}_i}$. Start the construction by first applying all the α and β rules to the formulae in $\Gamma_0^{\mathcal{C}_i}$ which are also in $\Gamma_0^{\mathcal{C}_{i+1}}$. Because states are downward saturated and $\langle\langle\emptyset\rangle\rangle \Box(\varphi \Rightarrow \psi) \in \Gamma_0^{\mathcal{C}_{i+1}}$, we also add $\langle\langle\emptyset\rangle\rangle \Box(\varphi \Rightarrow \psi)$, $\varphi \Rightarrow \psi$, and $[\emptyset]\langle\langle\emptyset\rangle\rangle \Box(\varphi \Rightarrow \psi)$ to the sets of formulae created so far. At this point of the construction, we have generated a set $\{\Delta_0^{\mathcal{C}_{i+1}}, \ldots, \Delta_n^{\mathcal{C}_{i+1}}\}$, where every $\Delta_k^{\mathcal{C}_{i+1}} = \Delta_k^{\mathcal{C}_i} \cup \{\langle\langle\emptyset\rangle\rangle \Box(\varphi \Rightarrow$

$\psi), \varphi \Rightarrow \psi, [\emptyset]\langle\langle\emptyset\rangle\rangle \Box(\varphi \Rightarrow \psi)\}$, for all $0 \leq k \leq n$. Note that the number of sets of formulae created so far is exactly the same as the number of states created from $\Gamma_0^{\mathcal{C}_i}$, as the same rules were applied in the same order and we only added formulae to those states. Take $\Delta_k^{\mathcal{C}_i}$ in $states(\Gamma_0^{\mathcal{C}_i})$. If $\Delta_k^{\mathcal{C}_i} \notin cons(\Gamma_0^{\mathcal{C}_i})$, then $\Delta_k^{\mathcal{C}_{i+1}}$ is not consistent either and any attempt to expand $\Delta_k^{\mathcal{C}_{i+1}}$ will result in an inconsistent state that will be later removed by **E1**. Assume $\Delta_k^{\mathcal{C}_i} \in cons(\Gamma_0^{\mathcal{C}_i})$. We now apply the β rule to $\varphi \Rightarrow \psi$ in $\Delta_k^{\mathcal{C}_{i+1}}$. Let φ be $l_1 \wedge \ldots \wedge l_m$, for some $m \in \mathbb{N}$. As states are downward saturated, they contain either one of the literals in $\{\neg l_1, \ldots, \neg l_m\}$ or ψ. By Corollary 4.29, every state $\Delta_k^{\mathcal{C}_{i+1}}$ contains $l \vee \neg l$, for all literals in $\Lambda_{\mathcal{C}_{i+1}} = \Lambda_{\mathcal{C}_i}$. By construction, every state is downward saturated. Therefore, every state contains l_j or $\neg l_j$, for $1 \leq j \leq m$. Choose any $\neg l_j$, $0 \leq j \leq m$, and try to expand $\Delta_k^{\mathcal{C}_{i+1}}$. If $\Delta_k^{\mathcal{C}_{i+1}}$ contains $\neg l_j$, we do not need to add $\neg l_j$. If $\Delta_k^{\mathcal{C}_{i+1}}$ does not contain $\neg l_j$, then it must contain l_j; thus, adding $\neg l_j$ results in an inconsistent state which will be later removed by an application of rule **E1**. Also, by Lemma 4.35, ψ is included in every $\Delta_k^{\mathcal{C}_i}$ that contains all the literals in φ and no new consistent states are created. That is, $|cons(\Gamma_0^{\mathcal{C}_{i+1}})| \leq |cons(\Gamma_0^{\mathcal{C}_i})|$. Moreover, for all $\Delta_k^{\mathcal{C}_{i+1}} \in cons(\Gamma_0^{\mathcal{C}_{i+1}})$, there is $\Delta_k^{\mathcal{C}_i} \in cons(\Gamma_0^{\mathcal{C}_i})$, such that $\Delta_k^{\mathcal{C}_i} \subseteq \Delta_k^{\mathcal{C}_{i+1}} = \Delta_k^{\mathcal{C}_i} \cup \{\langle\langle\emptyset\rangle\rangle \Box(\varphi \Rightarrow \psi), \varphi \Rightarrow \psi, [\emptyset]\langle\langle\emptyset\rangle\rangle \Box(\varphi \Rightarrow \psi)\}$.

The above corresponds to the first application of the rule **SR**. Again, the application of **SR** to $\Gamma_0^{\mathcal{C}_{i+1}}$ results in a set $states(\Gamma_0^{\mathcal{C}_{i+1}})$ with $|states(\Gamma_0^{\mathcal{C}_{i+1}})| \geq |states(\Gamma_0^{\mathcal{C}_i})|$. However, for the set $cons(\Gamma_0^{\mathcal{C}_{i+1}}) \subseteq states(\Gamma_0^{\mathcal{C}_{i+1}})$ of all consistent states, we have that $|cons(\Gamma_0^{\mathcal{C}_{i+1}})| \leq |cons(\Gamma_0^{\mathcal{C}_i})|$. We now apply the **Next** rule to states in $\mathcal{P}^{\mathcal{C}_{i+1}}$ and show that further applications of **SR** will not contribute with new consistent states in $\mathcal{P}^{\mathcal{C}_{i+1}}$.

Let $\Delta_k^{\mathcal{C}_{i+1}}$ be a consistent state that contains ψ. If $\psi \in \Delta_k^{\mathcal{C}_i}$ (for instance, because it is the right-hand side of another coalition clause whose left-hand side is also satisfied in $\Delta_k^{\mathcal{C}_i}$), then the prestates created from $\Delta_k^{\mathcal{C}_{i+1}}$ are exactly as the prestates created from $\Delta_k^{\mathcal{C}_i}$, except for the clause related to $\varphi \Rightarrow \psi$ in $\mathcal{N}_{i+1}$, that is, if Γ is a prestate created from $\Delta_k^{\mathcal{C}_i}$, then $\Gamma' = \Gamma \cup \{\langle\langle\emptyset\rangle\rangle \Box(\varphi \Rightarrow \psi)\}$ is a prestate created from $\Delta_k^{\mathcal{C}_{i+1}}$. Thus, $|prestates(\Delta_k^{\mathcal{C}_{i+1}})| = |prestates(\Delta_k^{\mathcal{C}_i})|$ and for all Γ created from $\Delta_k^{\mathcal{C}_i}$ there is a prestate Γ' created from $\Delta_k^{\mathcal{C}_{i+1}}$ such that $\Gamma \subseteq \Gamma'$. Moreover, as $\langle\langle\emptyset\rangle\rangle \Box(\varphi \Rightarrow \psi)$ is an α formula, if Δ is a state created from a prestate Γ in $prestates(\Delta_k^{\mathcal{C}_i})$, then Δ' created from the prestate $\Gamma' = \Gamma \cup \{\langle\langle\emptyset\rangle\rangle \Box(\varphi \Rightarrow \psi)\}$ is such that $\Delta' = \Delta \cup \{\langle\langle\emptyset\rangle\rangle \Box(\varphi \Rightarrow \psi), \varphi \Rightarrow \psi, [\emptyset]\langle\langle\emptyset\rangle\rangle \Box(\varphi \Rightarrow \psi)\}$. Reasoning as above, no new consistent states are created from further application of **SR** to prestates created from $\Delta_k^{\mathcal{C}_{i+1}} = \Delta_k^{\mathcal{C}_i} \cup \{\langle\langle\emptyset\rangle\rangle \Box(\varphi \Rightarrow \psi), \varphi \Rightarrow \psi, [\emptyset]\langle\langle\emptyset\rangle\rangle \Box(\varphi \Rightarrow \psi)\}$, if $\psi \in \Delta_k^{\mathcal{C}_i}$. That is, for all $\Gamma' \in prestates(\Delta_k^{\mathcal{C}_{i+1}})$, such that $\psi \in \Delta_k^{\mathcal{C}_i}$, we have that $|cons(\Gamma')| \leq |cons(\Gamma)|$, where $\Gamma \in prestates(\Delta_k^{\mathcal{C}_i})$.

If $\psi \notin \Delta_k^{\mathcal{C}_i}$, then let m and l be the number of positive and negative coalition formulae in $\Delta_k^{\mathcal{C}_i}$, respectively. From $\Delta_k^{\mathcal{C}_i}$, a set of prestates $\{\Gamma_1^{\mathcal{C}_i}, \ldots, \Gamma_p^{\mathcal{C}_i}\}$, for some $p \in \mathbb{N}$, is created by an application of the rule **Next**. In particular, there is a prestate, say $\Gamma_1^{\mathcal{C}_i}$, which contains only the clauses in $\{\psi' \mid \varphi' \Rightarrow [\emptyset]\psi' \in \mathcal{N}_i$ and $\Delta_k^{\mathcal{C}_i} \models \varphi'\} \cup \{D, \langle\langle\emptyset\rangle\rangle \Box D \mid D \in \mathcal{U}_i \cup \mathcal{N}_i \cup \Theta_{\mathcal{C}_i}\}$. This particular prestate exists because in the initial set of formulae we have a clause as, for instance, $\langle\langle\emptyset\rangle\rangle \Box(l \vee \neg l)$, for some literal $l \in \Lambda_{\mathcal{C}_i}$, which cannot occur in $\mathcal{N}_i$ since the normal form requires that all disjunctions are kept in their simplest form. As $\langle\langle\emptyset\rangle\rangle \Box(l \vee \neg l)$ is in the initial set of formulae, by Lemma 4.27, $[\emptyset]\langle\langle\emptyset\rangle\rangle \Box(l \vee \neg l)$ is in every state of the pretableau. Say the position of $[\emptyset]\langle\langle\emptyset\rangle\rangle \Box(l \vee \neg l)$ in $\mathfrak{L}(\Delta_k^{\mathcal{C}_i})$ is 0. Then by applying the rule **Next** to $\Delta_k^{\mathcal{C}_i}$, we create a prestate Γ_σ with $\sigma_a = 0$ for all $a \in \Sigma_{\mathcal{C}_i}$ where no other formulae are added, besides the formulae in the scope of $[\emptyset]$ and formulae of the form $\langle\langle\emptyset\rangle\rangle \Box D$, for $D \in \mathcal{U}_i \cup \mathcal{N}_i \cup \Theta_{\mathcal{C}_i}$.

The right-hand side of a coalition clause is a positive or a negative coalition formula. If ψ is of the form $[\mathcal{A}]\chi$ (resp. $\langle\mathcal{A}\rangle\chi$), then the number of positive and negative coalition formulae in $\Delta_k^{\mathcal{C}_{i+1}}$ are $m+1$ and l (resp. m and $l+1$), respectively. From $\Delta_k^{\mathcal{C}_{i+1}}$, a set of prestates $\{\Gamma_1^{\mathcal{C}_{i+1}},\ldots,\Gamma_q^{\mathcal{C}_{i+1}}\}$, for some $q\in\mathbb{N}$, is created. Now, note that there must be a prestate, say $\Gamma_1^{\mathcal{C}_{i+1}}$, which is like $\Gamma_1^{\mathcal{C}_i}$, but where the formulae related to $\varphi\Rightarrow\psi$ in $\mathcal{C}_{i+1}$ are added, that is, $\Gamma_1^{\mathcal{C}_{i+1}}$ contains only the formulae in $\{\psi' \mid \varphi'\Rightarrow[\emptyset]\psi'\in\mathcal{N}_{i+1}$ and $\Delta_k^{\mathcal{C}_i}\models\varphi'\}\cup\{D,\langle\langle\emptyset\rangle\rangle\,\Box D \mid D\in\mathcal{U}_i\cup\mathcal{N}_{i+1}\cup\Theta_{\mathcal{C}_i}\}$.

If $\mathcal{A}=\emptyset$, as formulae in the scope of $[\emptyset]$ are all in $\Gamma_1^{\mathcal{C}_{i+1}}$, we add to the pretableau the edges $\Delta_k^{\mathcal{C}_{i+1}}\xrightarrow{\sigma}\Gamma_1^{\mathcal{C}_{i+1}}$, for all σ. Note that in this case we also have that prestates created from $\Delta_k^{\mathcal{C}_{i+1}}$ are exactly as the prestates created from $\Delta_k^{\mathcal{C}_i}$, except for the formulae related to $\varphi\Rightarrow\psi$ in $\mathcal{N}_{i+1}$, that is, if Γ is a prestate created from $\Delta_k^{\mathcal{C}_i}$, then Γ' created from $\Delta_k^{\mathcal{C}_{i+1}}$ is $\Gamma'=\Gamma\cup\{\langle\langle\emptyset\rangle\rangle\,\Box(\varphi\Rightarrow\psi),\chi\}$. Thus, $|prestates(\Delta_k^{\mathcal{C}_{i+1}})|=|prestates(\Delta_k^{\mathcal{C}_i})|$ and for all Γ created from $\Delta_k^{\mathcal{C}_i}$ there is a prestate Γ' created from $\Delta_k^{\mathcal{C}_{i+1}}$ such that $\Gamma\subseteq\Gamma'$. Reasoning as in Lemma 4.34, the addition of a formula in the scope of $\langle\langle\emptyset\rangle\rangle\,\Box$ has no effect on the number of states created from Γ' compared with the number of states created from Γ, as we only apply an α rule to such a formula; also, as we are adding a propositional disjunction to a prestate, reasoning as above, further application of **SR** to Γ' will not increase the number of states created from Γ' in $\mathcal{T}_+^{\mathcal{C}_{i+1}}$, that is, $|cons(\Gamma')|\leq|cons(\Gamma)|$ and for all $\Delta'\in cons(\Gamma')$, there is $\Delta\in cons(\Gamma)$, such that $\Delta'=\Delta\cup\{\langle\langle\emptyset\rangle\rangle\,\Box(\varphi\Rightarrow\psi),\varphi\Rightarrow\psi,[\emptyset]\langle\langle\emptyset\rangle\rangle\,\Box(\varphi\Rightarrow\psi),\psi,\chi\}$.

Note that there is no coalition clause of the form $\varphi\Rightarrow\langle\mathcal{A}\rangle\chi$, where $\mathcal{A}=\Sigma_{\mathcal{C}_{i+1}}$, because the transformation rule τ_{Σ_φ} rewrites such formulae as $\varphi\Rightarrow[\emptyset]\chi$ and because the applications of **CRES3** cannot produce a resolvent where there is a formulae in the scope of $\langle\Sigma_{\mathcal{C}_{i+1}}\rangle$. So, we do not need to treat this case here.

If $\mathcal{A}\neq\emptyset$ (resp. $\mathcal{A}\neq\Sigma_{\mathcal{C}_{i+1}}$), then a prestate, say $\Gamma_{m+1}^{\mathcal{C}_{i+1}}$ (resp. $\Gamma_{l+1}^{\mathcal{C}_{i+1}}$), containing χ (and possibly other formulae) might be created. We add the prestate and the edges $\Delta_k^{\mathcal{C}_{i+1}}\xrightarrow{\sigma}\Gamma_{m+1}^{\mathcal{C}_{i+1}}$, where $\sigma_{\mathcal{A}}=m+1$ (resp. $\Delta_k^{\mathcal{C}_{i+1}}\xrightarrow{\sigma}\Gamma_{l+1}^{\mathcal{C}_{i+1}}$, where $\Sigma_{\mathcal{C}_{i+1}}\setminus\mathcal{A}\subseteq N(\sigma)$ and $neg(\sigma)=l+1$) to the pretableau. Note, however, that as $\chi\in\mathcal{D}_{\mathcal{C}_i}$, by Lemma 4.27, every state created from $\Gamma_1^{\mathcal{C}_{i+1}}$ contains $(\chi\vee\neg\chi)$; as states are downward saturated, every state contains either χ or $\neg\chi$. Therefore, a state containing χ and all other disjunctions that might be included in $\Gamma_{m+1}^{\mathcal{C}_{i+1}}$ (resp. $\Gamma_{l+1}^{\mathcal{C}_{i+1}}$) has already been created by applications of **SR** to $\Gamma_1^{\mathcal{C}_{i+1}}$ and it is not added to the pretableau. Instead, we add double edges from $\Gamma_{m+1}^{\mathcal{C}_{i+1}}$ (resp. $\Gamma_{l+1}^{\mathcal{C}_{i+1}}$) to the already existing states. If χ is the empty disjunction some new states are created, but all of them contain an inconsistency and will be removed later by the rule **E1**. Again, if Γ is a prestate created from $\Delta_k^{\mathcal{C}_i}$ and Γ' is a prestate created from $\Delta_k^{\mathcal{C}_{i+1}}$ we have that $|cons(\Gamma')|\leq|cons(\Gamma)|$. Also, for all $\Delta'\in cons(\Gamma')$, there is $\Delta\in cons(\Gamma)$, such that either $\Delta'=\Delta\cup\{\langle\langle\emptyset\rangle\rangle\,\Box(\varphi\Rightarrow\psi),\varphi\Rightarrow\psi,[\emptyset]\langle\langle\emptyset\rangle\rangle\,\Box(\varphi\Rightarrow\psi)\}$ (it is as before) or $\Delta'=\Delta\cup\{\langle\langle\emptyset\rangle\rangle\,\Box(\varphi\Rightarrow\psi),\varphi\Rightarrow\psi,[\emptyset]\langle\langle\emptyset\rangle\rangle\,\Box(\varphi\Rightarrow\psi),\chi\}$ (it has the formula in the scope of $[\mathcal{A}]$ or $\langle\mathcal{A}\rangle$ included in the state).

Overall, the inclusion of either positive or negative coalition formulae in a state $\Delta_k^{\mathcal{C}_{i+1}}$ might add to the number of prestates, but not to the number of consistent states which are the successors of $\Delta_k^{\mathcal{C}_i}$, that is, we might have $|prestates(\Delta_k^{\mathcal{C}_{i+1}})|\geq|prestates(\Delta_k^{\mathcal{C}_i})|$, but

$$|\bigcup_{\Gamma'\in prestates(\Delta_k^{\mathcal{C}_{i+1}})} cons(\Gamma')|\leq|\bigcup_{\Gamma\in prestates(\Delta_k^{\mathcal{C}_i})} cons(\Gamma)|.$$

As prestates are removed from rule **PR**, they have no effect on the size of the tableau.

By induction on the steps of the construction, all added states are inconsistent. As **PR** only removes prestates, after exhaustively applying **E1**, the number of states in $\mathcal{T}_+^{\mathcal{C}_{i+1}}$ cannot be greater than the number of states in $\mathcal{T}_+^{\mathcal{C}_i}$. Thus, $|\mathcal{T}_+^{\mathcal{C}_{i+1}}| \leq |\mathcal{T}_+^{\mathcal{C}_i}|$. As $\mathcal{S}_+^{\mathcal{C}_i} = \bigcup_{\Gamma \in \mathcal{P}^{\mathcal{C}_i}} cons(\Gamma)$ and $\mathcal{S}_+^{\mathcal{C}_{i+1}} = \bigcup_{\Gamma^{\mathcal{C}_{i+1}} \in \mathcal{P}^{\mathcal{C}_{i+1}}} cons(\Gamma^{\mathcal{C}_{i+1}})$, we have that for all $\Delta^{\mathcal{C}_{i+1}} \in \mathcal{S}_+^{\mathcal{C}_{i+1}}$ there is $\Delta^{\mathcal{C}_i} \in \mathcal{S}_+^{\mathcal{C}_i}$, such that $\Delta^{\mathcal{C}_i} \subseteq \Delta^{\mathcal{C}_{i+1}}$. ■

From the lemmas above, if a coalition problem in $\mathsf{DSNF_{CL}}$ $\mathcal{C}_{i+1}$ is obtained from $\mathcal{C}_i$ by an application of any of the resolution rules presented in Section 3.2, the size of the tableau for $\mathcal{C}_{i+1}$ is not greater than the size of the tableau for $\mathcal{C}_i$, after the rule **E1** has been applied.

THEOREM 4.37
Let $\mathcal{C}_0, \ldots, \mathcal{C}_n$ be a derivation and $\mathcal{T}_+^{\mathcal{C}_i}$ be the tableau for $\mathcal{C}_i$, $0 \leq i \leq n$, after the **E1** has been exhaustively applied. Let $\mathcal{S}_+^{\mathcal{C}_i}$ and $\mathcal{S}_+^{\mathcal{C}_{i+1}}$ be the set of states in $\mathcal{T}_+^{\mathcal{C}_i}$ and $\mathcal{T}_0^{\mathcal{C}_{i+1}}$, respectively. Then $|\mathcal{T}_+^{\mathcal{C}_0}| \geq \ldots \geq |\mathcal{T}_+^{\mathcal{C}_n}|$ and for all $\Delta^{\mathcal{C}_{i+1}} \in \mathcal{S}_+^{\mathcal{C}_{i+1}}$ there is $\Delta^{\mathcal{C}_i} \in \mathcal{S}_+^{\mathcal{C}_i}$, such that $\Delta^{\mathcal{C}_i} \subseteq \Delta^{\mathcal{C}_{i+1}}$.

PROOF OF THEOREM 4.37. By the definition of derivation, $\mathcal{C}_{i+1}$ is obtained from $\mathcal{C}_i$ by either adding a clause to $\mathcal{I}_i, \mathcal{U}_i$, or $\mathcal{N}_i$. By Lemmas 4.33, 4.34 and 4.36, including a clause in any of those sets does not increase the size of the tableau after the rule **E1** has been exhaustively applied. Thus, $|\mathcal{T}_+^{\mathcal{C}_0}| \geq \ldots \geq |\mathcal{T}_+^{\mathcal{C}_n}|$. By the same lemmas, for all $\Delta^{\mathcal{C}_{i+1}} \in \mathcal{S}_+^{\mathcal{C}_{i+1}}$ there is $\Delta^{\mathcal{C}_i} \in \mathcal{S}_+^{\mathcal{C}_i}$, such that $\Delta^{\mathcal{C}_i} \subseteq \Delta^{\mathcal{C}_{i+1}}$. ■

The next result will be used later in the completeness proof for $\mathsf{RES_{CL}}$.

THEOREM 4.38 (Completeness of classical propositional resolution [19])
If $\mathcal{S}$ is an unsatisfiable set of propositional clauses, then there is a refutation from $\mathcal{S}$ by the resolution method, where the inference rule **RES** is given by $\{(D \vee l), (D' \vee \neg l)\} \vdash (D \vee D')$.

The inference rules **IRES1** and **GRES1** together correspond to classical resolution as given in [19]. The next lemma shows that if the propositional part of a coalition problem in $\mathsf{DSNF_{CL}}$ is unsatisfiable, then there is a refutation using only the inference rules **IRES1** and **GRES1**.

LEMMA 4.39
Let $\mathcal{C} = (\mathcal{I}, \mathcal{U}, \mathcal{N})$ be a coalition problem in $\mathsf{DSNF_{CL}}$. If $\mathcal{I} \cup \mathcal{U}$ is unsatisfiable, there is a refutation for $\mathcal{I} \cup \mathcal{U}$ using only the inference rules **IRES1** and **GRES1**.

PROOF OF LEMMA 4.39. If $\mathcal{I} \cup \mathcal{U}$ is unsatisfiable, by Theorem 4.38, there is a refutation from $\mathcal{I} \cup \mathcal{U}$ by the resolution method. Let $\mathcal{C}'_0, \ldots, \mathcal{C}'_n$, with $n \in \mathbb{N}$, be a sequence of sets of propositional clauses, where $\mathcal{C}'_0 = \mathcal{I} \cup \mathcal{U}$, $\mathbf{false} \in \mathcal{C}'_n$, and, for each $1 \leq i \leq n$, $\mathcal{C}'_{i+1}$ is the set of clauses obtained by adding to $\mathcal{C}'_i$ the resolvent of an application of the classical resolution rule **RES** to clauses in $\mathcal{C}'_i$. We inductively construct a refutation $\mathcal{C}_0, \ldots, \mathcal{C}_n$ for $\mathcal{C} = (\mathcal{I}, \mathcal{U}, \mathcal{N})$ as follows. In the base case, $\mathcal{C}_0 = \mathcal{C}$. For the induction step, let $\mathcal{C}_0, \ldots, \mathcal{C}_i$ be the derivation already constructed. In $\mathcal{C}'_0, \ldots, \mathcal{C}'_i, \mathcal{C}'_{i+1}$, we obtained $(D \vee D')$ by an application of **RES** to $(D \vee l)$ and $(D' \vee \neg l) \in \mathcal{C}'_i$. As clauses in $\mathcal{C}'_i$ are in $\mathcal{I}_i \cup \mathcal{U}_i$, we say that a clause D *originates* from $\mathcal{I}_i$ (resp. $\mathcal{U}_i$), if D is in $\mathcal{I}_i$ (resp. $\mathcal{U}_i$).

- If $(D \vee l) \in \mathcal{C}'_i$ originates from a clause in $\mathcal{I}_i$ and $(D' \vee \neg l) \in \mathcal{C}'_i$ originates from a clause in $\mathcal{I}_i \cup \mathcal{U}_i$, then let $\mathcal{C}_{i+1} = (\mathcal{I}_i \cup \{D \vee D'\}, \mathcal{U}_i, \mathcal{N}_i)$, where $D \vee D'$ is obtained by an application of **IRES1** to $(D \vee l)$ and $(D' \vee \neg l)$ in $\mathcal{C}_i$, and we have $\mathcal{C}'_{i+1} = \mathcal{I}_{i+1} \cup \mathcal{U}_{i+1}$;
- If both $(D \vee l)$ and $(D' \vee \neg l)$ in $\mathcal{C}'_i$ originate from clauses in $\mathcal{U}_i$, then let $\mathcal{C}_{i+1} = (\mathcal{I}_i, \mathcal{U}_i \cup \{D \vee D'\}, \mathcal{N}_i)$, where $D \vee D'$ is obtained by an application of **GRES1** to $(D \vee l)$ and $(D' \vee \neg l)$ in $\mathcal{C}_i$, and we have $\mathcal{C}'_{i+1} = \mathcal{I}_{i+1} \cup \mathcal{U}_{i+1}$.

By construction, **false** $\in \mathcal{C}'_n$, thus there is a refutation in $\mathsf{RES}_{\mathsf{CL}}$ using only the inference rules **IRES1** and **GRES1**. ■

LEMMA 4.40
Let $\mathcal{C}=(\mathcal{I},\mathcal{U},\mathcal{N})$ be a coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$ and $\mathcal{T}_+^{\mathcal{C}}$ be the tableau for $\mathcal{C}$ after the **E1** has been exhaustively applied. If $\mathcal{T}_+^{\mathcal{C}}$ is closed, then $\mathcal{I}\cup\mathcal{U}$ is unsatisfiable. Moreover, there is a refutation from $\mathcal{C}$ that uses only the inference rules **IRES1** and **GRES1**.

PROOF OF LEMMA 4.40. If $\mathcal{T}_+^{\mathcal{C}}$ is closed, all initial states have been eliminated by **E1**, that is, all initial states contain propositional inconsistencies. By Lemma 4.28 if $\varphi \in \mathcal{U}\cup\mathcal{N}\cup\Theta_{\mathcal{C}}$, then $\varphi \in \Delta$, for all $\Delta \in \mathcal{T}_0^{\mathcal{C}}$ and, therefore, φ is in every initial state. By construction, if $\varphi \in \mathcal{I}$, because φ is in the initial prestate and states are downward saturated, then φ is in all initial states. Thus, if all initial states are inconsistent, by Theorem 4.26, we have that

$$\bigwedge_{D\in\mathcal{I}} D \wedge \bigwedge_{D'\in\mathcal{U}} D' \wedge \bigwedge_{(C\Rightarrow D'')\in\mathcal{N}} (\neg C \vee D'') \wedge \bigwedge_{D'''\in\Theta_{\mathcal{C}}} D'''$$

is not satisfiable. As $\bigwedge_{D'''\in\Theta_{\mathcal{C}}} D'''$ is valid, we have that

$$\bigwedge_{D\in\mathcal{I}} D \wedge \bigwedge_{D'\in\mathcal{U}} D' \wedge \bigwedge_{(C\Rightarrow D'')\in\mathcal{N}} (\neg C \vee D'')$$

is unsatisfiable. By Lemma 4.35, D'' on the right-hand side of a coalition clause $C \Rightarrow D''$ holds where C holds. Therefore,

$$(\bigwedge_{D\in\mathcal{I}} D \wedge \bigwedge_{D'\in\mathcal{U}} D' \wedge \bigwedge_{(C\Rightarrow D'')\in\mathcal{N}} \neg C) \vee (\bigwedge_{D\in\mathcal{I}} D \wedge \bigwedge_{D'\in\mathcal{U}} D' \wedge \bigwedge_{(C\Rightarrow D'')\in\mathcal{N}} C \wedge \bigwedge_{(C\Rightarrow D'')\in\mathcal{N}} D'')$$

is not satisfiable. Now, there is no formula in any state which is the negation of a coalition modality because of the particular normal form we use here. Thus, as $\bigwedge_{(C\Rightarrow D'')\in\mathcal{N}} D''$ is not propositional, it cannot contribute directly to deletion of the initial states (by **E1**). Therefore,

$$(\bigwedge_{D\in\mathcal{I}} D \wedge \bigwedge_{D'\in\mathcal{U}} D' \wedge \bigwedge_{(C\Rightarrow D'')\in\mathcal{N}} \neg C) \vee (\bigwedge_{D\in\mathcal{I}} D \wedge \bigwedge_{D'\in\mathcal{U}} D' \wedge \bigwedge_{(C\Rightarrow D'')\in\mathcal{N}} C)$$

is unsatisfiable. By distribution, we have that

$$(\bigwedge_{D\in\mathcal{I}} D \wedge \bigwedge_{D'\in\mathcal{U}} D') \wedge (\bigwedge_{(C\Rightarrow D'')\in\mathcal{N}} \neg C \vee \bigwedge_{(C\Rightarrow D'')\in\mathcal{N}} C)$$

is unsatisfiable. As $(\bigwedge_{(C\Rightarrow D'')\in\mathcal{N}} \neg C) \vee \bigwedge_{(C\Rightarrow D'')\in\mathcal{N}} C))$ is a tautology, by the semantics of conjunction, we have that:

$$\bigwedge_{D\in\mathcal{I}} D \wedge \bigwedge_{D'\in\mathcal{U}} D'$$

is unsatisfiable. By Lemma 4.39, there is a refutation from $\mathcal{C}=(\mathcal{I},\mathcal{U},\mathcal{N})$ that uses only the inference rules **IRES1** and **GRES1**. ■

Next we prove that $\mathsf{RES}_{\mathsf{CL}}$ is complete. That is, given a unsatisfiable coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$, there is a refutation for it.

THEOREM 4.41 (Completeness of $\mathsf{RES}_{\mathsf{CL}}$)
Let $\mathcal{C}=(\mathcal{I},\mathcal{U},\mathcal{N})$ be an unsatisfiable coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$. Then there is a refutation for $\mathcal{C}$ using the inference rules **IRES1**, **GRES1**, **CRES1-4** and **RW1-2**.

PROOF OF THEOREM 4.41. Let $\mathcal{C}=(\mathcal{I},\mathcal{U},\mathcal{N})$ be an unsatisfiable coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$. Firstly, if $\mathcal{C}$ is unsatisfiable and only if $\mathcal{C}$ is unsatisfiable, by Theorem 4.26, we have that $\mathcal{T}^{\mathcal{C}}$ is closed. Obviously, if $\mathcal{C}$ is unsatisfiable, every coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$ $\mathcal{C}_0,\ldots$ in a derivation, is also unsatisfiable. We show that if $\mathcal{C}$ is unsatisfiable, then we can inductively construct a refutation $\mathcal{R}_{\mathcal{C}}=\mathcal{C}_0,\ldots,\mathcal{C}_m$, $m\in\mathbb{N}$. By Theorem 4.37, we have that $|\mathcal{T}_+^{\mathcal{C}_0}|\geq\ldots\geq|\mathcal{T}_+^{\mathcal{C}_m}|$ and we show that $\mathcal{T}_+^{\mathcal{C}_m}$ is closed, that is, that the application of the resolution rules in the derivation $\mathcal{R}_{\mathcal{C}}=\mathcal{C}_0,\ldots,\mathcal{C}_m$ correspond to deletions of states in the corresponding tableaux $\mathcal{T}_+^{\mathcal{C}_0},\ldots,\mathcal{T}_+^{\mathcal{C}_m}$.

For the base case, $\mathcal{C}$ contains either **false** or a propositional symbol and its negation. In the first case, $\mathcal{T}_+^{\mathcal{C}_0}$ is closed, no states are further deleted, and by Lemma 4.40, $\mathcal{R}_{\mathcal{C}}=\mathcal{C}_0$, is a refutation for $\mathcal{C}$. In the second case, if $\{p,\neg p\}\in\mathcal{I}$, then Lemma 4.40 also ensures that there is a refutation for $\mathcal{C}$ which uses only the inference rules **IRES1** and **GRES1**.

Assume $\mathcal{T}_+^{\mathcal{C}_0}$ is not closed. Let $\mathcal{R}_{\mathcal{C}}=\mathcal{C}_0,\ldots,\mathcal{C}_i$ be a derivation and $\mathcal{C}_i$ be the coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$ obtained after the inference rules **IRES1** and **GRES1** have been exhaustively applied. Let $\mathcal{T}_+^{\mathcal{C}_i}$ be the tableau for $\mathcal{C}_i$ after the deletion rule **E1** has been exhaustively applied.

If $\mathcal{T}_+^{\mathcal{C}_i}$ is closed, by Lemma 4.40, $\mathcal{R}_{\mathcal{C}}=\mathcal{C}_0,\ldots,\mathcal{C}_i$, $m=i$, is a refutation for $\mathcal{C}$ which uses only the inference rules **IRES1** and **GRES1**.

If $\mathcal{T}_+^{\mathcal{C}_i}$ is not closed, then, by Theorem 4.26, the final tableau $\mathcal{T}^{\mathcal{C}_i}$ for $\mathcal{C}_i$ must be closed, as the tableau procedure is complete. Therefore, there must be a state in $\mathcal{T}_+^{\mathcal{C}_i}$ that can be deleted by an application of the deletion rule **E2**. Let Δ be the first state to which **E2** is applied. By the definition of **E2**, Δ is deleted if there is a move vector $\sigma\in D(\Delta)$ such that there is no Δ' with $\Delta\xrightarrow{\sigma}\Delta'$. Let $\mathfrak{L}(\Delta)$ be the ordered list of coalition formulae in Δ and let $n(\varphi)$ be the the position of φ in $\mathfrak{L}(\Delta)$. From Lemma 4.28, global clauses and tautologies are in every state. By Lemma 4.35, the right-hand side of coalition formulae are in the states where the left-hand side is satisfied. Therefore, by Lemmas 4.28 and 4.35, and by the definition of the rule **Next** in the tableau construction, which gives the set of prestates that are connected from Δ by an edge labelled by σ, we obtain that Δ' is one of the minimal downward saturated sets built from $\mathcal{U}_i\cup\Theta_{\mathcal{C}_i}\cup\{D'\mid C'\Rightarrow[\mathcal{A}]D'\in\mathcal{N}_i,\Delta\models C'$ and $\sigma_a=n([\mathcal{A}]D')$, for all $a\in\mathcal{A}\}\cup\{D''\mid C''\Rightarrow\langle\mathcal{A}\rangle D''\in\mathcal{N}_i,\Delta\models C'',\Sigma_{\mathcal{C}_i}\setminus\mathcal{A}\subseteq N(\sigma)$ and $neg(\sigma)=n(\langle\mathcal{A}\rangle D'')\}$. If Δ' is not in $\mathcal{T}_+^{\mathcal{C}_i}$, it must have been deleted by an application of **E1**, because Δ is the first state being deleted by **E2**. Therefore, by the definition of **E1**, Δ' contains propositional inconsistencies. Thus, as tautologies are valid formulae,

$$\bigwedge_{D\in\mathcal{U}_i} D \wedge \bigwedge_{\substack{C'\Rightarrow[\mathcal{A}]D'\in\mathcal{N}_i\\ \Delta\models C'\\ \sigma_a=n([\mathcal{A}]D'),\ \text{for all } a\in\mathcal{A}}} D' \wedge \bigwedge_{\substack{C''\Rightarrow\langle\mathcal{A}\rangle D''\in\mathcal{N}_i\\ \Delta\models C''\\ \Sigma_{\mathcal{C}_i}\setminus\mathcal{A}\subseteq N(\sigma)\\ neg(\sigma)=n(\langle\mathcal{A}\rangle D'')}} D''$$

is unsatisfiable. As this corresponds to a propositional set of clauses, by Theorem 4.38 there must be a refutation by the resolution method for this set. Let $\mathcal{C}'_0,\ldots,\mathcal{C}'_n$, with $n\in\mathbb{N}$, be a sequence of sets of propositional clauses, where $\mathcal{C}'_n$ contains the constant **false**, $\mathcal{C}'_0$ is given by the set of clauses above and, for each $1\leq j\leq n$, $\mathcal{C}'_{j+1}$ is the set of clauses obtained by adding to $\mathcal{C}'_j$ the resolvent of an application of the classical resolution rule **RES** to clauses with complementary literals in $\mathcal{C}'_j$. We inductively construct a derivation $\mathcal{C}_i,\ldots,\mathcal{C}_{m'}$, with $m'\in\mathbb{N}$, such that $\mathcal{C}_{m'}$ contains either a clause of the form $C\Rightarrow[\mathcal{A}]\mathbf{false}$ or $C\Rightarrow\langle\mathcal{A}\rangle\mathbf{false}$, where C is a conjunction and $\mathcal{A}$ is a coalition. In the base case, $\mathcal{C}_0=\mathcal{C}$. For the induction step, let $\mathcal{C}_i,\ldots,\mathcal{C}_j$ be the derivation already constructed. In $\mathcal{C}'_0,\ldots,\mathcal{C}'_j,\mathcal{C}'_{j+1}$, we obtained $(D\vee D')$ by an application of **RES** to $(D\vee l)$ and $(D'\vee\neg l)\in\mathcal{C}'_j$. As clauses in $\mathcal{C}'_j$ are in either $\mathcal{U}_i$ or are the right-hand side of a coalition clause in $\mathcal{N}_i$, for $1\leq j\leq n$ and $1\leq i\leq m'$, we say that a clause D *originates* from $\mathcal{U}_i$ (resp. $\mathcal{N}_i$), if D is in $\mathcal{U}_i$ (resp. $C\Rightarrow D$ is in $\mathcal{N}_i$). The possible derivations in $\mathsf{RES}_{\mathsf{CL}}$ are as follows:

1. If $D\vee l$ originates from a clause $C'\Rightarrow[\mathcal{A}](D\vee l)\in\mathcal{N}_{i+j}$ and $D'\vee\neg l$ originates from a clause $C''\Rightarrow[\mathcal{B}](D'\vee\neg l)\in\mathcal{N}_{i+j}$, by soundness of the tableau procedure we have that $\mathcal{A}\cap\mathcal{B}=\emptyset$; let $\mathcal{C}_{i+j+1}=\mathcal{C}_{i+j}\cup\{C'\wedge C''\Rightarrow[\mathcal{A}\cup\mathcal{B}](D\vee D')\}$, where $C'\wedge C''\Rightarrow[\mathcal{A}\cup\mathcal{B}](D\vee D')$ is obtained by an application of **CRES1** to $C'\Rightarrow[\mathcal{A}](D\vee l)$ and $C''\Rightarrow[\mathcal{B}](D\vee\neg l)$;
2. If $(D\vee l)\in\mathcal{U}_{i+j}$ and $(D'\vee\neg l)$ originates from a clause $C'\Rightarrow[\mathcal{A}](D'\vee\neg l)\in\mathcal{N}_{i+j}$, then let $\mathcal{C}_{i+j+1}=\mathcal{C}_{i+j}\cup\{C'\Rightarrow[\mathcal{A}](D\vee D')\}$, where $C'\Rightarrow[\mathcal{A}](D\vee D')$ is obtained by an application of **CRES2** to $D\vee l$ and $C'\Rightarrow[\mathcal{A}](D'\vee\neg l)$;
3. If $D\vee l$ originates from a clause $C'\Rightarrow[\mathcal{A}](D\vee l)\in\mathcal{N}_{i+j}$ and $D'\vee\neg l$ originates from a clause $C''\Rightarrow\langle\mathcal{B}\rangle(D'\vee\neg l)\in\mathcal{N}_{i+j}$, by soundness of the tableaux procedure, we have that $\mathcal{A}\subseteq\mathcal{B}$; let $\mathcal{C}_{i+j+1}=\mathcal{C}_{i+j}\cup\{C'\wedge C''\Rightarrow\langle\mathcal{B}\setminus\mathcal{A}\rangle(D\vee D')\}$, where $C'\wedge C''\Rightarrow\langle\mathcal{B}\setminus\mathcal{A}\rangle(D\vee D')$ is obtained by an application of **CRES3** to $C'\Rightarrow[\mathcal{A}](D\vee l)$ and $C''\Rightarrow\langle\mathcal{B}\rangle(D'\vee\neg l)$;
4. If $D\vee l\in\mathcal{U}_{i+j}$ and $D'\vee\neg l$ originates from a clause $C''\Rightarrow\langle\mathcal{A}\rangle(D'\vee\neg l)\in\mathcal{N}_{i+j}$, then let $\mathcal{C}_{i+j+1}=\mathcal{C}_{i+j}\cup\{C'\Rightarrow\langle\mathcal{A}\rangle(D\vee D')\}$, where $C'\Rightarrow\langle\mathcal{A}\rangle(D\vee D')$ is obtained by an application of **CRES4** to $D\vee l$ and $C'\Rightarrow\langle\mathcal{A}\rangle(D'\vee\neg l)$.

Thus, there is a derivation $\mathcal{C}'_i,\ldots,\mathcal{C}'_{i+n}$, which uses only the inference rules **CRES1-4** and, by construction, either $[\mathcal{A}]\mathbf{false}$ or $\langle\mathcal{A}\rangle\mathbf{false}$ are in $\mathcal{C}_{i+n}$.

If $\Delta\in\mathcal{T}_+^{\mathcal{C}_i}$ has been removed by **E2** during the deletion phase in the construction of $\mathcal{T}_+^{\mathcal{C}_i}$, then there is a derivation $\mathcal{C}_i,\ldots,\mathcal{C}_{i+n}$, using the the inference rules **CRES1-4**, such that either $C\Rightarrow[\mathcal{A}]\mathbf{false}$ or $C\Rightarrow\langle\mathcal{A}\rangle\mathbf{false}$ are in $\mathcal{C}_{i+n}$. Let $\mathcal{C}_{i+n+1}$ be the coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$ obtained from $\mathcal{C}_{i+n}$ by adding the result of **RW1** (resp. **RW2**) applied to $C\Rightarrow[\mathcal{A}]\mathbf{false}$ (resp. $C\Rightarrow\langle\mathcal{A}\rangle\mathbf{false}$) in $\mathcal{C}_{i+n}$, that is, if $C=l_0\wedge\ldots\wedge l_p$, $p\in\mathbb{N}$, we have that $\mathcal{U}_{i+n+1}=\mathcal{U}_{i+n}\cup\{\neg l_0\vee\ldots\vee\neg l_p\}$. Note that, because $\Delta\in\mathcal{T}_+^{\mathcal{C}_i}$, Δ is consistent. Also note that the applications of **CRES1-4** only add coalition formulae to the tableaux $\mathcal{T}_0^{\mathcal{C}_i},\ldots,\mathcal{T}_0^{\mathcal{C}_{i+n}}$. From the proof of Lemma 4.36, the construction rules applied to Δ only affect the states created from (prestates created from) Δ. Note, however, that for all $\mathcal{T}_0^{\mathcal{C}_j}$, for $i<j\leq n+i$, there is a state $\Delta''\in\mathcal{T}_0^{\mathcal{C}_j}$ which is exactly like Δ, but which might contain clauses related to the resolvents from **CRES1-4**. Recall that if the application of **CRES1-4** result in a coalition clause $\varphi\Rightarrow\psi$, then $\Delta''=\Delta\cup\{\langle\langle\emptyset\rangle\rangle\,\square(\varphi\Rightarrow\psi),\varphi\Rightarrow\psi,[\emptyset]\langle\langle\emptyset\rangle\rangle\,\square(\varphi\Rightarrow\psi)\}$. As those clauses do not occur negated in the set of clauses, we have that $\Delta''\in\mathcal{S}_+^{\mathcal{C}_{i+n}}$. As $\Delta\subseteq\Delta''$, if $\Delta\models C$, then $\Delta''\models C$. As **RW1** (resp.**RW2**) adds a disjunction to the set of global clauses, by Lemma 4.34, there is Δ''' in $S_0^{\mathcal{C}_{i+n+1}}$, such that $\Delta''\subseteq\Delta'''$. By Lemma 4.28, as $\neg l_0\vee\ldots\vee\neg l_p\in\mathcal{U}_{i+n+1}$, all states in $\mathcal{T}_+^{\mathcal{C}_{i+n+1}}$ contain $\neg l_0\vee\ldots\vee\neg l_p$. Now, as $\Delta'''\models C$ and Δ''' contains $\neg l_0\vee\ldots\vee\neg l_p$, $\Delta'''\notin\mathcal{T}_+^{\mathcal{C}_{i+n+1}}$, that is, Δ''' is not consistent. Finally, by Theorem 4.37, for all states s' in $\mathcal{T}_+^{\mathcal{C}_{i+n+1}}$ there is a state s in $\mathcal{T}_+^{\mathcal{C}_{i+n}}$, such that $s\subseteq s'$. However, there is at least one state in $\mathcal{T}_+^{\mathcal{C}_{i+n}}$, namely $\Delta''\models C$, for which there is no consistent state $\Delta'''\in\mathcal{T}_+^{\mathcal{C}_{i+n+1}}$ such that $\Delta''\subseteq\Delta'''$, as states that satisfy C are removed by **E1** from $\mathcal{T}_+^{\mathcal{C}_{i+1}}$. Therefore $|\mathcal{S}_+^{\mathcal{C}_{i+n+1}}|<|\mathcal{S}_+^{\mathcal{C}_{i+n}}|$.

Summarising, an application of **RW1** (resp. **RW2**) to $C \Rightarrow [\mathcal{A}]\mathbf{false}$ (resp. $C \Rightarrow \langle\mathcal{A}\rangle\mathbf{false}$) in $\mathcal{C}_{i+n}$ adds $\neg C$ to $\mathcal{U}_{i+n+1}$ in $\mathcal{C}_{i+n+1}$, the next coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$ in the derivation. Thus, states that satisfy the left-hand side of clauses that lead to deletion of Δ' by **E2** in the tableau for $\mathcal{C}_i$ will be removed by **E1** from tableau $\mathcal{T}_{+}^{\mathcal{C}_{i+n+1}}$. This shows that if a state Δ does not have all needed successors, there is some inconsistency at the propositional level of one of its successor, Δ', and applications of the inference rules **RW1-2** correspond, therefore, to the elimination of the states Δ'' such that $\Delta \subseteq \Delta''$ in $\mathcal{T}_{+}^{\mathcal{C}_{i+n+1}}$.

From the above, every application of **E2** can be simulated in $\mathsf{RES}_{\mathsf{CL}}$ by a derivation using **IRES1** and **GRES1**, followed by a derivation using **CRES1-4**, and an application of either **RW1** or **RW2**. As there is no state like Δ in $\mathcal{T}_{+}^{\mathcal{C}_{i+n+1}}$, if $\mathcal{T}_{+}^{\mathcal{C}_{i+n+1}}$ is not closed, we inductively apply the same steps above, removing states which have not all required successors at each time. We note that the number of states that can be deleted by **E2** is in $\mathcal{O}(2^{|\mathcal{C}|})$, where $|\mathcal{C}|$ is the size of the coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$ $|\mathcal{C}|$ [9]. As the number of states being removed by **E2** is finite and, by Theorem 4.37, as the formulae added by the resolution rules do not contribute to increase the size of the tableaux corresponding to steps of a derivation, at some point there is a tableau $\mathcal{T}_{+}^{\mathcal{C}_m}$ which is closed.

By induction on the number of applications of **E2**, if $\mathcal{T}^{\mathcal{C}_0}$ is closed, then there is a derivation $\mathcal{C}_0,\dots,\mathcal{C}_m$, where $\mathcal{C}=\mathcal{C}_0$, $\mathcal{C}_m=(\mathcal{I}_m,\mathcal{U}_m,\mathcal{N}_m)$, and every $\mathcal{C}_{i+1}$ is obtained by an application of rules in $\mathsf{RES}_{\mathsf{CL}}$ to clauses in $\mathcal{C}_i$. Moreover, because $\mathcal{T}_{+}^{\mathcal{C}_m}$ is closed, by Lemma 4.40, we have that $\mathbf{false} \in \mathcal{I}_m \cup \mathcal{U}_m$. Thus, if $\mathcal{C}$ is unsatisfiable, then there is a refutation by $\mathsf{RES}_{\mathsf{CL}}$. ■

4.5 Complexity

The satisfiability problem for CL is PSPACE-complete: the lower bound is due to the fact that KD, which is PSPACE-hard [14], is a sublogic of CL; the upper bound is proved by showing that the size of a satisfiability game is restricted by the modal-depth of a formula and that the construction of each branch of such a game takes polynomial time [17].

The satisfiability problem for ATL is EXPTIME-complete: the lower bound is shown by reducing the global consequence problem in K, which is EXPTIME-hard, to ATL; the upper bound is due to the fact that the existence of a model tree for a formula φ can be checked in time exponential in the length of φ [20].

The reduction of the global consequence problem in K given in [20] can straightforwardly be modified to a reduction of the global consequence problem in K to the satisfiability of coalition problems in $\mathsf{DSNF}_{\mathsf{CL}}$. It follows that the satisfiability problem of coalition problems is EXPTIME-hard. It also follows that the satisfiability problem of CL^{+}, the language of CL plus the $\langle\langle\emptyset\rangle\rangle\,\square$ operator, is EXPTIME-hard.

THEOREM 4.42
The satisfiability problem for coalition problems in $\mathsf{DSNF}_{\mathsf{CL}}$ is EXPTIME-hard.

PROOF OF THEOREM 4.42. Immediate from the translation of a coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$ into CL^{+}, given in Section 4.4, and from [20, Lemma 4.10, page 785]. ■

THEOREM 4.43
The decision procedure based on $\mathsf{RES}_{\mathsf{CL}}$ is in EXPTIME.

PROOF OF THEOREM 4.43. Let $|\mathcal{C}|$ be the size of the coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$ $\mathcal{C}$. The tableau structure for $\mathcal{C}$ has $\mathcal{O}(2^{|\mathcal{C}|})$ states [9]. As it is shown in the completeness proof for $\mathsf{RES}_{\mathsf{CL}}$, every state deletion corresponds to a propositional refutation, whose complexity is in $\mathcal{O}(2^{|\mathcal{C}|})$ [19]. Thus, the overall complexity of $\mathsf{RES}_{\mathsf{CL}}$ is in $\mathcal{O}(2^{|\mathcal{C}|}) \times \mathcal{O}(2^{|\mathcal{C}|})$, that is, $\mathcal{O}(2^{|\mathcal{C}|})$. ■

5 Conclusions

We have presented a sound, complete, and terminating resolution-based calculus for the Coalition Logic CL, which is equivalent to the next-time fragment of ATL. The approach uses a clausal normal form for CL: a formula to be checked for satisfiability is firstly transformed into a coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$, which separates the dimensions to which the resolution rules are applied. The transformation into the normal form is satisfiability preserving and polynomially bounded by the size of the original formula [22]. The calculus consists of six resolution inference rules and two rewriting rules: **IRES1** and **GRES1** are applied to clauses in the propositional language of a coalition problem, that is, to initial and global clauses; **CRES1-4** are applied to coalition and global clauses; and the rewriting rules **RW1-2** ensure that if a set of right-hand sides of coalition and global clauses leads to a contradiction, then the left-hand sides of those coalition clauses should not be satisfied. The resolution-based method for CL is a syntactic variation of the resolution calculus for the next time fragment of ATL given in [22]. Adding to the presentation in [22], we provide full completeness proof for $\mathsf{RES}_{\mathsf{CL}}$. Completeness is proved with respect to the tableau procedure given in [9]. We have shown that deletions in a tableau correspond to applications of the inference rules of $\mathsf{RES}_{\mathsf{CL}}$. Thus, if a tableau for a coalition problem is closed, there is a refutation based on the calculus given here. Moreover, if a tableau for a coalition problem is open, the existence of a model is ensured by soundness of the tableau procedure.

The calculus presented here is very simple in structure, so an implementation can be obtained in a quite straightforward way by extending existing resolution provers for either PTL or CTL, for instance, and it is left as future work.

Future work also includes the extension of this calculus to the full language of ATL, which can be achieved by designing a set of resolution-like inference rules to deal with eventualities, that is, formulae which hold at some future time of a run. Usually, inference rules to deal with eventualities are not trivial, as their application requires the search for so-called loops in the set of clauses. For instance, in [7], the search for loops used in an application of the temporal resolution rule for PTL is the most expensive part of the calculus. Therefore, to extend our calculus to ATL we would need to devise a correct loop-search algorithm for ATL.

Funding

This work was supported by the National Council for the Improvement of Higher Education (CAPES Foundation, BEX 8712/11-5) (to C.N.); EPSRC grant EP/D060451; and (1) National Natural Science Foundation of China (Grant No. 61303018), (2) the Technology Foundation for Selected Overseas Chinese Scholars, Bureau of Human Resources and Social Security of Beijing and (3) The Importation and Development of High-Caliber Talents Project of Beijing Municipal Institutions (Project name: Decision tree generation algorithm and its optimization of incomplete information systems) (to L.Z.).

References

[1] R. Alur, T. A. Henziger, and O. Kupferman. Alternating-time temporal logic. In *Proceedings of the 38th IEEE Symposium on Foundations of Computer Science*, W. P. de Roever, H. Langmaack, and A. Pnueli, eds, *Lecture Notes in Computer Science*, pp. 100–109. Springer, 1997.

[2] R. Alur, T. A. Henziger, and O. Kupferman. Alternating-time temporal logic. *Lecture Notes in Computer Science*, **1536**, 23–60, 1998.

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

[4] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Number 53 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2001.

[5] A. Degtyarev, M. Fisher, and B. Konev. Monodic temporal resolution. *ACM Transactions on Computational Logic*, **7**, 108–150, 2006.

[6] G. van Drimmelen. Satisfiability in alternating-time temporal logic. In *LICS '03: Proceedings of the 18th Annual IEEE Symposium on Logic in Computer Science*, P. G. Kolaitis, ed., pp. 208–217, IEEE Computer Society, 2003.

[7] M. Fisher, C. Dixon, and M. Peim. Clausal Temporal Resolution. *ACM Transactions on Computational Logic*, **2**, 2001.

[8] V. Goranko. Coalition games and alternating temporal logics. In *Proceedings of the 8th Conference on Theoretical Aspects of Rationality and Knowledge*, J. van Benthem, ed., TARK '01, pp. 259–272, Morgan Kaufmann Publishers Inc., 2001.

[9] V. Goranko and D. Shkatov. Tableau-based decision procedures for logics of strategic ability in multiagent systems. *ACM Transactions on Computational Logic*, **11**, 3:1–3:51, 2009.

[10] V. Goranko and G. van Drimmelen. Complete axiomatization and decidability of alternating-time temporal logic. *Theor. Comput. Sci.*, **353**, 93–117, 2006.

[11] R. Goré, J. Thomson, and F. Widmann. An experimental comparison of theorem provers for CTL. In *Eighteenth International Symposium on Temporal Representation and Reasoning, TIME 2011, Lübeck , Germany, September 12-14, 2011*, C. Combi, M. Leucker, and F. Wolter, eds, pp. 49–56. IEEE, 2011.

[12] H. Hansen. *Tableau Games for Coalition Logic and Alternating-time Temporal Logic – Theory and Implementation*. Master's Thesis, University of Amsterdam, 2004.

[13] U. Hustadt and R. A. Schmidt. Scientific benchmarking with temporal logic decision procedures. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Eighth International Conference (KR'2002)*, D. Fensel, F. Giunchiglia, D. McGuinness, and M.-A. Williams, eds, pp. 533–544. Morgan Kaufmann, 2002.

[14] R. E. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM J. Comput.*, **6**, 467–480, 1977.

[15] C. Nalon, L. Zhang, C. Dixon, and U. Hustadt. A resolution-based calculus for coalition logic (extended version). *Technical Report ULCS-13-004*, University of Liverpool, Liverpool, UK, May 2013. Available at http://intranet.csc.liv.ac.uk/research/techreports/?id=ULCS-13-004.

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

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

[18] D. A. Plaisted and S. A. Greenbaum. A structure-preserving clause form translation. *Journal of Symbolic Computation*, **2**, 293–304, 1986.

[19] J. A. Robinson. A Machine–oriented logic based on the resolution principle. *ACM Journal*, **12**, 23–41, 1965.

[20] D. Walther, C. Lutz, F. Wolter, and M. Wooldridge. ATL satisfiability is indeed ExpTime-complete. *Journal of Logic and Computation*, **16**, 765–787, 2006.

[21] P. Wolper. The tableau method for temporal logic: an overview. *Logique et Analyse*, **110–111**, 119–136, 1985.
[22] L. Zhang. *Clausal Reasoning for Branching-time Logics*. PhD Thesis, University of Liverpool, 2010.

Received 22 May 2013