Ordered Resolution for Coalition Logic

Ullrich Hustadt[1], Paul Gainer[1], Clare Dixon[1], Cláudia Nalon[2], and Lan Zhang[3]

[1] Department of Computer Science, University of Liverpool, UK
{uhustadt,sgpgaine,cldixon}@liverpool.ac.uk

[2] Department of Computer Science, University of Brasília, Brazil
nalon@unb.br

[3] Information School, Capital University of Economics and Business, China
lan@cueb.edu.cn

21st May 2015

1 Introduction

Coalition Logic CL was introduced by Pauly [19] as a logic for reasoning about what groups of agents can bring about by collective action. 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, can be read as *the coalition of agents* $\mathcal{A}$ *can bring about* φ or *the coalition of agents* $\mathcal{A}$ *is effective for* φ or *the coalition of agents* $\mathcal{A}$ *has a strategy to achieve* φ.

Coalition Logic is closely related to *Alternating-Time Temporal Logic*, ATL [1, 2, 3] a multi-modal logic with coalition quantifiers $\langle\langle\mathcal{A}\rangle\rangle$, where $\mathcal{A}$ is again a set of agents, and temporal operators $\bigcirc$ ("next"), $\square$ ("always") and $\mathcal{U}$ ("until"), that extends propositional logic with formulae of the form $\langle\langle\mathcal{A}\rangle\rangle\bigcirc\varphi$, $\langle\langle\mathcal{A}\rangle\rangle\square\varphi$ and $\langle\langle\mathcal{A}\rangle\rangle\varphi\,\mathcal{U}\,\psi$. 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 [28] and PSPACE-complete [20], respectively.

Methods for tackling the satisfiability problem for these logics include two tableau-based methods for ATL [28, 9], two automata-based methods [26, 10] for ATL, and one tableau-based method for CL [11]. An implementation of the two-phase tableau calculus by Goranko and Shkatov for ATL [9, 6] exists in the form of TATL [7]. A first resolution-based method for CL, $\mathsf{RES}_{\mathsf{CL}}$, consisting of a normal form transform and a resolution calculus, was presented in [17], and shown to be sound, complete and terminating. In particular, the completeness of $\mathsf{RES}_{\mathsf{CL}}$ is shown relative to the tableau calculus for ATL in [9]. If a CL formula φ is unsatisfiable, the corresponding tableau is closed. In the completeness proof for $\mathsf{RES}_{\mathsf{CL}}$ it is shown that deletions that produce the closed tableau correspond to applications of the resolution inference rules of $\mathsf{RES}_{\mathsf{CL}}$ that in turn produce a refutation of φ. A prototype implementation of $\mathsf{RES}_{\mathsf{CL}}$ in the programming language Prolog exists in the form of CLProver [18].

In this paper we revisit the resolution-based method for CL. First, we discuss variants of the normal form and normal form transformation for Coalition Logic. Second, we correct the completeness result for the calculus $\mathsf{RES}_{\mathsf{CL}}$ presented in [18] and present a revised completeness proof for the calculus. Third, we introduce *Vector Coalition Logic (VCL)* and a novel normal form, coalition problems in $\mathsf{DSNF}_{\mathsf{VCL}}$, for Coalition Logic. This novel normal form allows us to define $\mathsf{RES}^{\succ}_{\mathsf{CL}}$, an ordering refinement of the resolution calculus $\mathsf{RES}_{\mathsf{CL}}$ for CL. Finally, we prove soundness, completeness, termination and complexity results for $\mathsf{RES}^{\succ}_{\mathsf{CL}}$.

The paper is organised as follows. In the next section, we present the syntax, axiomatisation, and semantics of CL.

In Section 3, we introduce various normal forms and normal transformations for CL, including coalition problems in $\mathsf{DSNF}_{\mathsf{CL}}$ and coalition problems in unit $\mathsf{DSNF}_{\mathsf{CL}}$, state again the rules of the resolution calculus $\mathsf{RES}_{\mathsf{CL}}$, show that the calculus is not incomplete on coalition problems in $\mathsf{DSNF}_{\mathsf{CL}}$, and prove that $\mathsf{RES}_{\mathsf{CL}}$ is complete for coalition problems in unit $\mathsf{DSNF}_{\mathsf{CL}}$.

In Section 4, we show that a naive imposition of an ordering refinement on resolution leads to incompleteness even for coalition problems in unit $\mathsf{DSNF}_{\mathsf{CL}}$. This motivates the introduction of a new logic, Vector Coalition Logic, and a new normal form, coalition problems in $\mathsf{DSNF}_{\mathsf{VCL}}$. We define a normal form transformation for Coalition Logic formulae and coalition problems in $\mathsf{DSNF}_{\mathsf{CL}}$ into coalition problems in $\mathsf{DSNF}_{\mathsf{VCL}}$. We prove that this normal form transformation preserves satisfiability. We then introduce an ordered resolution calculus $\mathsf{RES}^{\succ}_{\mathsf{CL}}$ with inference rules that operate on coalition problems in $\mathsf{DSNF}_{\mathsf{VCL}}$. We prove that $\mathsf{RES}^{\succ}_{\mathsf{CL}}$ is sound and complete. We also show that derivation by $\mathsf{RES}^{\succ}_{\mathsf{CL}}$ always terminate and discuss the complexity of a decision procedure based on $\mathsf{RES}^{\succ}_{\mathsf{CL}}$.

Conclusions and future work are given in Section 5.

2 Coalition Logic

2.1 Syntax

Let $\Sigma \subset \mathbb{N}$ to be a non-empty, finite set of agents and $\Pi = \{p, q, r, \ldots, p_1, q_1, r_1, \ldots\}$ be a non-empty, finite or countably infinite set of propositional symbols. A *coalition* $\mathcal{A}$ is a subset of Σ. Formulae in CL are constructed from propositional symbols using Boolean operators and the *coalition modalities* $[\mathcal{A}]$ and $\langle\mathcal{A}\rangle$.

Definition 1. The set $\mathsf{WFF}_{\mathsf{CL}}$ of CL *formulae* is inductively defined as follows.

- all propositional symbols in Π are CL formulae;
- if φ and ψ are CL formulae, then so are $\neg\varphi$ (negation) and $(\varphi \to \psi)$ (implication);
- if φ_i, $1 \leq i \leq n$, $n \in \mathbb{N}_0$, are CL formula, then so are $(\varphi_1 \wedge \ldots \wedge \varphi_n)$ (conjunction), also written $\bigwedge_{i=1}^n \varphi_i$, and $(\varphi_1 \vee \ldots \vee \varphi_n)$ (disjunction), also written $\bigvee_{i=1}^n \varphi_i$; and
- if $\mathcal{A} \subseteq \Sigma$ is a (finite) set of agents and φ is a CL formula, then so are $[\mathcal{A}]\varphi$ (*positive coalition formula*) and $\langle\mathcal{A}\rangle\varphi$ (*negative coalition formula*).

Parentheses will be omitted if the reading is not ambiguous. We consider the conjunction and disjunction operator to be associative and commutative, that is, we do not distinguish between, for example, $(p \vee (q \vee r))$, $((r \vee p) \vee q)$ and $(q \vee r \vee p)$. The formula $\bigvee_{i=1}^0 \varphi_i$ is called the *empty disjunction*, also denoted by **false**, while $\bigwedge_{i=1}^0 \varphi_i$ is called the *empty conjunction*, also denoted by **true**. When enumerating a specific set of agents, we often omit the curly brackets. For example, we write $[1, 2]\varphi$ instead of $[\{1, 2\}]\varphi$, for a formula φ. A *coalition formula* is either a positive or a negative coalition formula. In the following, we use "formula(e)" and "well-formed formula(e)" interchangeably.

Definition 2. 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*.

Definition 3. We call a CL formula *atomic* if it is a propositional symbol and *non-atomic* otherwise. For a non-atomic CL formula φ we denote by $\mathrm{op}(\varphi)$ its *principal operator* and by $\mathrm{args}(\varphi)$ the list of its direct subformulae. In more detail:

$$
\begin{aligned}
\mathrm{op}(\neg\psi) &= \neg & \mathrm{args}(\neg\psi) &= (\psi)\\
\mathrm{op}(\psi_1 \to \psi_2) &= \to & \mathrm{args}(\psi_1 \to \psi_2) &= (\psi_1, \psi_2)\\
\mathrm{op}(\textstyle\bigwedge_{i=1}^n \psi_i) &= \wedge & \mathrm{args}(\textstyle\bigwedge_{i=1}^n \psi_i) &= (\psi_1, \ldots, \psi_n)\\
\mathrm{op}(\textstyle\bigvee_{i=1}^n \psi_i) &= \vee & \mathrm{args}(\textstyle\bigvee_{i=1}^n \psi_i) &= (\psi_1, \ldots, \psi_n)\\
\mathrm{op}([\mathcal{A}]\psi) &= [\mathcal{A}] & \mathrm{args}([\mathcal{A}]\psi) &= (\psi)\\
\mathrm{op}(\langle\mathcal{A}\rangle\psi) &= \langle\mathcal{A}\rangle & \mathrm{args}(\langle\mathcal{A}\rangle\psi) &= (\psi)
\end{aligned}
$$

We denote the length of a list L by $|L|$ and denote the i-the element of a list L by $L[i]$.

We use sequences of positive numbers, called *positions*, to refer to specific subformulae in a formula. The empty sequence ϵ is a position in any formula. A sequence $i \cdot \lambda$ is a *position in a formula* φ if φ is a non-atomic formula, $1 \leq i \leq |\text{args}(\varphi)|$, and λ is a position in $\text{args}(\varphi)[i]$. If λ is a position in a formula φ, then the *subformula* $\varphi_{|\lambda}$ *of* φ *at position* λ is φ, if $\lambda = \epsilon$, and $\varphi_{i|\lambda'}$ if $\text{args}(\varphi) = (\varphi_1, \ldots, \varphi_n)$ and $\lambda = i \cdot \lambda'$, for some i, $1 \leq i \leq n$. By $\text{Pos}(\varphi)$ we denote the set of all positions in a formula φ.

The *size* $|\varphi|$of a formula φ to be the size of the set $\text{Pos}(\varphi)$. For a finite set Φ of formulae, the size $|\Phi|$ of Φ is $|\Phi| = \Sigma_{\varphi \in \Phi} |\varphi|$.

Definition 4. The *modal depth* $\text{mdepth}(\varphi)$ of a CL formula φ is inductively defined as follows:

$$\text{mdepth}(p) = 0 \text{ for every propositional symbol } p$$
$$\text{mdepth}(\neg\psi) = \text{mdepth}(\psi) \qquad \text{mdepth}(\psi_1 \to \psi_2) = \max\{\text{mdepth}(\psi_i) \mid i \in \{1,2\}\}$$
$$\text{mdepth}(\textstyle\bigwedge_{i=1}^n \psi_i) = \max\{\text{mdepth}(\psi_i) \mid 1 \leq i \leq n\} \qquad \text{mdepth}([\mathcal{A}]\psi) = 1 + \text{mdepth}(\psi)$$
$$\text{mdepth}(\textstyle\bigvee_{i=1}^n \psi_i) = \max\{\text{mdepth}(\psi_i) \mid 1 \leq i \leq n\} \qquad \text{mdepth}(\langle\mathcal{A}\rangle\psi) = 1 + \text{mdepth}(\psi)$$

The *modal layer* $\text{mlayer}(\varphi, \lambda)$ of a position λ in φ is inductively defined as follows:

$$\text{mlayer}(\psi, \epsilon) = 0$$
$$\text{mlayer}(\neg\psi, 1 \cdot \lambda') = \text{mlayer}(\psi, \lambda') \qquad \text{mlayer}(\psi_1 \to \psi_2, i \cdot \lambda') = \text{mlayer}(\psi_i) \text{ for } i \in \{1,2\}$$
$$\text{mlayer}(\textstyle\bigwedge_{i=1}^n \psi_i, j \cdot \lambda') = \text{mlayer}(\psi_j, \lambda') \qquad \text{mlayer}([\mathcal{A}]\psi, 1 \cdot \lambda') = 1 + \text{mlayer}(\psi, \lambda')$$
$$\text{mlayer}(\textstyle\bigvee_{i=1}^n \psi_i, j \cdot \lambda') = \text{mlayer}(\psi_j, \lambda') \qquad \text{mlayer}(\langle\mathcal{A}\rangle\psi, 1 \cdot \lambda') = 1 + \text{mlayer}(\psi, \lambda')$$

If ψ is a subformula of φ at position λ and $\text{mlayer}(\varphi, \lambda) = l$, then we say the subformula occurrence of ψ occurs at *modal layer* l of φ.

2.2 Axiomatisation

Coalition logic can be axiomatised by the following axiom schemata [20], where $\mathcal{A}, \mathcal{A}'$ are coalitions and φ, φ_1, φ_2 are well-formed formulae:

$$\begin{array}{lcl}
\bot & : & \neg[\mathcal{A}]\mathbf{false} \\
\top & : & [\mathcal{A}]\mathbf{true} \\
\mathbf{\Sigma} & : & \neg[\emptyset]\neg\varphi \to [\Sigma]\varphi \\
\mathbf{M} & : & [\mathcal{A}](\varphi_1 \wedge \varphi_2) \to [\mathcal{A}]\varphi_1 \\
\mathbf{S} & : & [\mathcal{A}]\varphi_1 \wedge [\mathcal{A}']\varphi_2 \to [\mathcal{A} \cup \mathcal{A}'](\varphi_1 \wedge \varphi_2), \text{ if } \mathcal{A} \cap \mathcal{A}' = \emptyset \\
\langle\mathcal{A}\rangle & : & \langle\mathcal{A}\rangle\varphi \leftrightarrow \neg[\mathcal{A}]\neg\varphi
\end{array}$$

together with axiom schemata for propositional logic and the inference rules **modus ponens** (from φ_1 and $\varphi_1 \to \varphi_2$ 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 *monotonicity principle* $[\mathcal{A}](\varphi \wedge \psi) \to [\mathcal{A}]\varphi \wedge [\mathcal{A}]\psi$, follows from axiom $\mathbf{M}$.

CL is a non-normal modal logic, that is, the schema that represents the *additivity principle*, $[\mathcal{A}]\varphi \wedge [\mathcal{A}]\psi \to [\mathcal{A}](\varphi \wedge \psi)$, does not hold, instead axiom $\mathbf{S}$ reflects the weaker form of additivity that holds for two positive coalition formulae in CL. Using the axiomatisation above it is possible to show that the schema

$$\mathbf{S'} \ : \ [\mathcal{A}]\psi_1 \wedge \langle\mathcal{B}\rangle\psi_2 \to \langle\mathcal{B} \setminus \mathcal{A}\rangle(\psi_1 \wedge \psi_2), \text{ if } \mathcal{A} \subseteq \mathcal{B}$$

holds. Schema **S'** indicates under what conditions and how a negative and a positive coalition formula can be 'combined'. Just as in the case of basic modal logic there is no corresponding schema for two negative coalition formulae.

2.3 Semantics

We use *Concurrent Game Structures* (CGSs) [3, 9] for describing the semantics of ATL. Also, the semantics given here uses *rooted models*, that is, models with a distinguished state where a formula has to be satisfied.

Definition 5. 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;
- $d : \Sigma \times \mathcal{S} \longrightarrow \mathbb{N}_0^+$, 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 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.

Given a CGF $\mathcal{F} = (\Sigma, \mathcal{S}, s_0, d, \delta)$ 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 $\kappa[n]$ to refer to the n-th element of κ.

Definition 6. 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 7. A move vector σ *extends* an $\mathcal{A}$-move $\sigma_{\mathcal{A}}$, 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 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 9. Let $\mathcal{M} = (\mathcal{F}, \Pi, \pi)$ with $\mathcal{F} = (\Sigma, \mathcal{S}, s_0, d, \delta)$ be a CGM with $s \in \mathcal{S}$. The satisfaction relation, denoted by $\models$, is inductively defined as follows.

- $\langle \mathcal{M}, s \rangle \models p$ iff $p \in \pi(s)$, for all $p \in \Pi$;
- $\langle \mathcal{M}, s \rangle \models \neg\varphi$ iff $\langle \mathcal{M}, s \rangle \not\models \varphi$;
- $\langle \mathcal{M}, s \rangle \models \varphi \to \psi$ iff $\langle \mathcal{M}, s \rangle \models \varphi$ implies $\langle \mathcal{M}, s \rangle \models \psi$;
- $\langle \mathcal{M}, s \rangle \models \bigwedge_{i=1}^{n} \varphi_i$ iff $\langle \mathcal{M}, s \rangle \models \varphi_i$ for all i, $1 \leq i \leq n$;
- $\langle \mathcal{M}, s \rangle \models \bigvee_{i=1}^{n} \varphi_i$ iff $\langle \mathcal{M}, s \rangle \models \varphi_i$ for some i, $1 \leq i \leq n$;
- $\langle \mathcal{M}, s \rangle \models [\mathcal{A}]\varphi$ iff there exists an $\mathcal{A}$-move $\sigma_{\mathcal{A}} \in D(\mathcal{A}, s)$ s.t.
  for all $\sigma \in D(s)$ $\sigma_{\mathcal{A}} \sqsubseteq \sigma$ implies $\langle \mathcal{M}, \delta(s, \sigma) \rangle \models \varphi$;
- $\langle \mathcal{M}, s \rangle \models \langle \mathcal{A} \rangle \varphi$ iff for all $\mathcal{A}$-moves $\sigma_{\mathcal{A}} \in D(\mathcal{A}, s)$
  exists $\sigma \in D(s)$ s.t. $\sigma_{\mathcal{A}} \sqsubseteq \sigma$ and $\langle \mathcal{M}, \delta(s, \sigma) \rangle \models \varphi$.

Definition 10. Let $\mathcal{M}$ be a CGM. A CL formula φ is *satisfied at the state s in* $\mathcal{M}$ if $\langle \mathcal{M}, s \rangle \models \varphi$ and φ is *satisfiable in* $\mathcal{M}$, denoted by $\mathcal{M} \models \varphi$, if $\langle \mathcal{M}, s_0 \rangle \models \varphi$. A finite set $\Gamma \subset \mathsf{WFF_{CL}}$ is *satisfiable in a state s in* $\mathcal{M}$, denoted by $\langle \mathcal{M}, s \rangle \models \Gamma$, if for all $\varphi_i \in \Gamma$, $0 \leq i \leq n$, $\langle \mathcal{M}, s \rangle \models \varphi_i$, and Γ is *satisfiable in* $\mathcal{M}$, denoted by $\mathcal{M} \models \Gamma$, if $\langle \mathcal{M}, s_0 \rangle \models \Gamma$.

As discussed in [19, 28, 9] 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 [28]. In this work, we will consider this particular notion of satisfiability. We denote by $\Sigma_\varphi \subseteq \Sigma$, the set of agents occurring in a well-formed formula φ or the set $\{a\}$ for some arbitrary agent $a \in \Sigma$ if the set of agents occurring in φ is empty (as the set of agents in a CGF and CGM has to be non-empty). If Φ is a set of well-formed formulae, $\Sigma_\Phi \subseteq \Sigma$ denotes $\bigcup_{\varphi \in \Phi} \Sigma_\varphi$.

Definition 11. A CL formula φ is *satisfiable* if there is a model $\mathcal{M} = (\Sigma_\varphi, \mathcal{S}, s_0, d, \delta, \Pi, \pi)$ such that $\langle \mathcal{M}, s_0 \rangle \models \varphi$. A CL formula φ is *valid* if for all models $\mathcal{M}$ we have $\langle \mathcal{M}, s_0 \rangle \models \varphi$. A finite set Γ of CL formulae is *satisfiable*, if there is a model $\mathcal{M}$ such that $\langle \mathcal{M}, s_0 \rangle \models \Gamma$. A finite set Γ of CL formulae is *valid*, if for all models $\mathcal{M}$ we have $\langle \mathcal{M}, s_0 \rangle \models \Gamma$.

Coalition logic shares a number of properties with basic modal logic, for instance, the *tree model property* [10], that is, if a formula φ is satisfiable, then there exists a CGM $\mathcal{M} = (\Sigma_\varphi, \mathcal{S}, s_0, d, \delta, \Pi, \pi)$ such that $\langle \mathcal{M}, s_0 \rangle \models \varphi$ and the successor relation on $\mathcal{S}$ is a tree with root s_0. We call such a model a *tree model* of φ.

Also the following property holds for Coalition Logic:

Lemma 1. Let φ be a CL formula and $\mathcal{M}$ be a tree model of φ. Let λ be position in φ and the subformula occurrence $\psi = \varphi|_\lambda$ occurs at modal layer l and ψ has modal depth k. Then the satisfiability of ψ in $\mathcal{M}$ only depends on states of $\mathcal{M}$ at tree depth i where $l \leq i \leq l + k$.

In particular, for propositional symbols occurring at modal layer l in a CL formula, only states of $\mathcal{M}$ at tree depth l are relevant.

This motivates the following transformation 'mlt' (*modal layer transformation*) of CL formula. With every propositional symbol p occurring in a formula φ and natural number n we uniquely associate a propositional symbol p_n not occurring in φ. Then 'mlt :' is inductively defined as follows.

$$
\begin{aligned}
\mathrm{mlt}(p, n) &= p_n \\
\mathrm{mlt}(\neg\psi, n) &= \neg\mathrm{mlt}(\psi, n) & \mathrm{mlt}(\psi_1 \to \psi_2) &= \mathrm{mlt}(\psi_1, n) \to \mathrm{mlt}(\psi_2, n) \\
\mathrm{mlt}(\textstyle\bigwedge_{i=1}^n \psi_i, n) &= \textstyle\bigwedge_{i=1}^n \mathrm{mlt}(\psi_i, n) & \mathrm{mlt}([\mathcal{A}]\psi, n) &= [\mathcal{A}]\mathrm{mlt}(\psi, n+1) \\
\mathrm{mlt}(\textstyle\bigvee_{i=1}^n \psi_i, n) &= \textstyle\bigvee_{i=1}^n \mathrm{mlt}(\psi_i, n) & \mathrm{mlt}(\langle\mathcal{A}\rangle\psi, n) &= \langle\mathcal{A}\rangle\mathrm{mlt}(\psi, n+1)
\end{aligned}
$$

The transformation mlt preserves satisfiability:

Lemma 2. Let φ be a CL formula. Then φ is satisfiable iff $\mathrm{mlt}(\varphi, 0)$ is satisfiable.

Proof. Note that $\Sigma_\varphi = \Sigma_{\mathrm{mlt}(\varphi,0)}$. Let $\mathcal{M}' = (\Sigma_\varphi, \mathcal{S}', s_0, d', \delta', \Pi', \pi')$ be a tree model of $\mathrm{mlt}(\varphi, 0)$. Let $\mathcal{M}$ be the model $(\Sigma_\varphi, \mathcal{S}', s_0, d', \delta', \Pi', \pi)$ such that for each state s at tree depth l, $\pi(s) = \{p \mid p_l \in \pi'(s)\}$. It follows from Lemma 1 that $\mathcal{M}$ is a model of φ.

Analogously, let $\mathcal{M} = (\Sigma_\varphi, \mathcal{S}, s_0, d, \delta, \Pi, \pi)$ be a tree model of φ. We defined a model $\mathcal{M}'$ as $(\Sigma_\varphi, \mathcal{S}, s_0, d, \delta, \Pi, \pi')$ such that for each state s at tree depth l, $\pi'(s) = \{p_l \mid p \in \pi(s)\}$. Again, it follows from Lemma 1 that $\mathcal{M}'$ is a model of $\mathrm{mlt}(\varphi, 0)$. □

3 Unrefined Resolution for CL

The resolution method presented in [17] proceeds by translating a CL formula φ that is to be tested for (un)satisfiability into a clausal normal form $\mathcal{C}$, a *coalition problem in Divided Separated Normal Form for Coalition Logic*, to which then resolution-based inference rules are applied. The application of these rules always terminates, either resulting in a coalition problem $\mathcal{C}'$ that is evidently contradictory or, otherwise, satisfiable. The formula φ is satisfiable iff $\mathcal{C}'$ is satisfiable.

In the following we first present the normal form transformation before introducing the inference rules.

3.1 Normal Form Transformation

The resolution-based calculus for CL, $\mathsf{RES_{CL}}$, operates on *coalition problems in Divided Separated Normal Form for Coalition Logic*, $\mathsf{DSNF_{CL}}$. Any CL formula in is firstly converted into a *coalition problem*, which is then transformed into a coalition problem in $\mathsf{DSNF_{CL}}$.

Definition 12. 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 $\mathsf{WFF_{CL}}$; 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 *size* $|\mathcal{C}|$ of a coalition problem $(\mathcal{I},\mathcal{U},\mathcal{N})$ is $|\mathcal{I}| + |\mathcal{U}| + |\mathcal{N}|$.

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 13. 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 form.* 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 such that within every conjunction and every disjunction literals are pairwise different.

Definition 14. A *coalition problem in $\mathsf{DSNF_{CL}}$* 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.

Definition 15. A *coalition problem in unit $\mathsf{DSNF_{CL}}$* 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 such that coalition clauses have the following form:

positive coalition clauses	$\bigwedge_{i=1}^{m} l'_i$	$\rightarrow$	$[\mathcal{A}]p$
negative coalition clauses	$\bigwedge_{i=1} l'_i$	$\rightarrow$	$\langle\mathcal{A}\rangle p$
	where p is a propositional symbol		

The transformation into the normal form is given by a set of rewriting rules. Let $\varphi \in \mathsf{WFF_{CL}}$ 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 rewriting rules:

$$
\begin{array}{rcl@{\qquad}rcl}
\varphi \rightarrow \psi & \Rightarrow & \neg\varphi \vee \psi & \neg\neg\varphi & \Rightarrow & \varphi \\
\neg(\bigwedge_{i=1}^{n} \varphi_i) & \Rightarrow & \bigvee_{i=1}^{n} \neg\varphi_i & \neg[\mathcal{A}]\varphi & \Rightarrow & \langle\mathcal{A}\rangle\neg\varphi \\
\neg(\bigvee_{i=1}^{n} \varphi_i) & \Rightarrow & \bigwedge_{i=1}^{n} \neg\varphi_i & \neg\langle\mathcal{A}\rangle\varphi & \Rightarrow & [\mathcal{A}]\neg\varphi \\
\neg(\varphi \rightarrow \psi) & \Rightarrow & \varphi \wedge \neg\psi & & &
\end{array}
$$

In addition, we want to remove occurrences of **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):

$\varphi \wedge \mathbf{true}$	$\Rightarrow$	φ	$\neg\mathbf{false}$	$\Rightarrow$	$\mathbf{true}$	$[\mathcal{A}]\mathbf{true}$	$\Rightarrow$	$\mathbf{true}$
$\varphi \vee \mathbf{true}$	$\Rightarrow$	$\mathbf{true}$	$\varphi \vee \varphi$	$\Rightarrow$	φ	$[\mathcal{A}]\mathbf{false}$	$\Rightarrow$	$\mathbf{false}$
$\varphi \wedge \mathbf{false}$	$\Rightarrow$	$\mathbf{false}$	$\varphi \wedge \varphi$	$\Rightarrow$	φ	$\langle\mathcal{A}\rangle\mathbf{true}$	$\Rightarrow$	$\mathbf{true}$
$\varphi \vee \mathbf{false}$	$\Rightarrow$	φ	$\varphi \vee \neg\varphi$	$\Rightarrow$	$\mathbf{true}$	$\langle\mathcal{A}\rangle\mathbf{false}$	$\Rightarrow$	$\mathbf{false}$
$\neg\mathbf{true}$	$\Rightarrow$	$\mathbf{false}$	$\varphi \wedge \neg\varphi$	$\Rightarrow$	$\mathbf{false}$			

In [17], given a formula φ, the transformation of φ into a coalition problem $(\mathcal{I},\mathcal{U},\mathcal{N})$ in $\mathsf{DSNF}_{\mathsf{CL}}$ is performed by exhaustively applying the rules of the rewriting system $R_1 = \{\Rightarrow^1_\wedge, \ldots, \Rightarrow^1_{\langle\Sigma_\varphi\rangle}\}$ given below, together with simplification, to the tuple $(\{t_0\}, \{t_0 \to \tau_0(\varphi)\}, \{\})$, where t_0 is a new propositional symbol and $\tau_0(\varphi)$ is the transformation of φ into NNF (where t is a literal; φ_i, $i \geq 0$, are CL formulae; $\mathcal{A}$ is a coalition; Σ_φ is the set of agents occurring in the original formula φ t_1 is a new propositional symbol; and disjunctions are associative and commutative):

$$
\begin{array}{rcl}
(\mathcal{I},\mathcal{U} \cup \{t \to \bigwedge_{i=1}^n \varphi_i\},\mathcal{N}) & \Rightarrow^1_\wedge & (\mathcal{I},\mathcal{U} \cup \bigcup_{i=1}^n \{t \to \varphi_i\},\mathcal{N}) \\
(\mathcal{I},\mathcal{U} \cup \{t \to \psi \vee \bigvee_{i=1}^n \varphi_i\},\mathcal{N}) & \Rightarrow^1_\vee & (\mathcal{I},\mathcal{U} \cup \{t \to t_1 \vee \bigvee_{i=1}^n \varphi_i, t_1 \to \psi\},\mathcal{N}) \\
& & \text{where } \psi \text{ is not a disjunction of literals} \\
(\mathcal{I},\mathcal{U} \cup \{t \to D\},\mathcal{N}) & \Rightarrow^1_{\to,\mathcal{U}} & (\mathcal{I},\mathcal{U} \cup \{\neg t \vee D\},\mathcal{N}) \\
& & \text{where } D \text{ is a disjunction of literals} \\
(\mathcal{I},\mathcal{U} \cup \{t \to D\},\mathcal{N}) & \Rightarrow^1_{\to,\mathcal{N}} & (\mathcal{I},\mathcal{U},\mathcal{N} \cup \{t \to D\}) \\
& & \text{where } D \text{ is either of the form } [\mathcal{A}]\varphi_1 \text{ or } \langle\mathcal{A}\rangle\varphi_1 \\
(\mathcal{I},\mathcal{U},\mathcal{N} \cup \{t \to [\mathcal{A}]\psi\}) & \Rightarrow^1_{[\_]} & (\mathcal{I},\mathcal{U} \cup \{t_1 \to \psi\},\mathcal{N} \cup \{t \to [\mathcal{A}]t_1\}) \\
& & \text{where } \psi \text{ is not a disjunction of literals} \\
(\mathcal{I},\mathcal{U},\mathcal{N} \cup \{t \to \langle\mathcal{A}\rangle\psi\}) & \Rightarrow^1_{\langle\_\rangle} & (\mathcal{I},\mathcal{U} \cup \{t_1 \to \psi\},\mathcal{N} \cup \{t \to \langle\mathcal{A}\rangle t_1\}) \\
& & \text{where } \psi \text{ is not a disjunction of literals} \\
& & \text{and } \mathcal{A} \neq \Sigma_\varphi \\
(\mathcal{I},\mathcal{U},\mathcal{N} \cup \{t \to \langle\Sigma_\varphi\rangle\varphi_1\}) & \Rightarrow^1_{\langle\Sigma_\varphi\rangle} & (\mathcal{I},\mathcal{U},\mathcal{N} \cup \{t \to [\emptyset]\varphi_1\})
\end{array}
$$

The rules $\Rightarrow^1_\vee$, $\Rightarrow^1_{[\_]}$, and $\Rightarrow^1_{\langle\_\rangle}$ use renaming [21] in order to bring coalition problems closer to divided separated normal form $\mathsf{DSNF}_{\mathsf{CL}}$. The implication $t_1 \to \psi$ introduced by these rules defines the new propositional symbol t_1 [21].

Note that if the subformula ψ occurs n times in φ, then the rewriting rules of R_1 will introduce n definitions $t_1 \to \psi$, ..., $t_n \to \psi$ during the normal form transformation. Obviously, it would suffice to introduce just one definition $t_1 \to \psi$ and to replace all occurrences of ψ by t_1 during the normal form transformation.

To achieve this we uniquely associate with every subformula ψ of φ a propositional symbol t_ψ that does not occur in φ. To ensure that a definition involving a subformula ψ is introduced only once during the normal form transformation we use the function $\mathrm{Def}(\psi, \Gamma)$ where ψ is a CL formula and Γ is a set of CL formulae, more precisely, a set of implications $t \to \theta$:

$$
\mathrm{Def}(\psi,\Gamma) = \begin{cases} \{t_\psi \to \psi\}, & \text{if } t_\psi \text{ does not occur in } \Gamma \\ \emptyset, & \text{otherwise} \end{cases}
$$

Given a formula φ, we then start its transformation with the coalition problem $(\{t_0\}, \{t_\varphi \to \tau_0(\varphi)\}, \{\})$ and exhaustively apply the rules of the rewriting system $R_2 = \{\Rightarrow^1_\wedge, \Rightarrow^1_{\to,\mathcal{U}}, \Rightarrow^1_{\to,\mathcal{N}}, \Rightarrow^1_{\langle\Sigma_\varphi\rangle}, \Rightarrow^2_\vee, \Rightarrow^2_{[\_]}, \Rightarrow^2_{\langle\_\rangle}\}$ together with simplification.

$$
\begin{array}{rcl}
(\mathcal{I},\mathcal{U} \cup \{t \to \psi \vee \bigvee_{i=1}^n \varphi_i\},\mathcal{N}) & \Rightarrow^2_\vee & (\mathcal{I},\mathcal{U} \cup \{t \to t_\psi \vee \bigvee_{i=1}^n \varphi_i\} \cup \mathrm{Def}(\psi,\mathcal{U} \cup \mathcal{N}),\mathcal{N}) \\
& & \text{where } \psi \text{ is not a disjunction of literals} \\
(\mathcal{I},\mathcal{U},\mathcal{N} \cup \{t \to [\mathcal{A}]\psi\}) & \Rightarrow^2_{[\_]} & (\mathcal{I},\mathcal{U} \cup \mathrm{Def}(\psi,\mathcal{U} \cup \mathcal{N}),\mathcal{N} \cup \{t \to [\mathcal{A}]t_\psi\}) \\
& & \text{where } \psi \text{ is not a disjunction of literals} \\
(\mathcal{I},\mathcal{U},\mathcal{N} \cup \{t \to \langle\mathcal{A}\rangle\psi\}) & \Rightarrow^2_{\langle\_\rangle} & (\mathcal{I},\mathcal{U} \cup \mathrm{Def}(\psi,\mathcal{U} \cup \mathcal{N}),\mathcal{N} \cup \{t \to \langle\mathcal{A}\rangle t_\psi\}) \\
& & \text{where } \psi \text{ is not a disjunction of literals} \\
& & \text{and } \mathcal{A} \neq \Sigma_\varphi
\end{array}
$$

For a discussion of the interdependencies between the normal form transformation and the resolution calculus, we will need a third variation of the rewriting system. R_3 consists of the rewriting rules $\Rightarrow^1_{\wedge}$, $\Rightarrow^1_{\rightarrow,\mathcal{U}}$, $\Rightarrow^1_{\rightarrow,\mathcal{N}}$, $\Rightarrow^1_{\langle\Sigma_\varphi\rangle}$, $\Rightarrow^2_{\vee}$, $\Rightarrow^3_{[\_]}$, and $\Rightarrow^3_{\langle\_\rangle}\}$ plus the simplification rules. The rewriting rules $\Rightarrow^3_{[\_]}$ and $\Rightarrow^3_{\langle\_\rangle}\}$ are defined as follows:

$$
\begin{array}{lll}
(\mathcal{I},\mathcal{U},\mathcal{N}\cup\{t\rightarrow[\mathcal{A}]\psi\}) & \Rightarrow^3_{[\_]} & (\mathcal{I},\mathcal{U}\cup\mathrm{Def}(\psi,\mathcal{U}\cup\mathcal{N}),\mathcal{N}\cup\{t\rightarrow[\mathcal{A}]t_\psi\}) \\
 & & \text{where } \psi \text{ is not a literal} \\
(\mathcal{I},\mathcal{U},\mathcal{N}\cup\{t\rightarrow\langle\mathcal{A}\rangle\psi\}) & \Rightarrow^3_{\langle\_\rangle} & (\mathcal{I},\mathcal{U}\cup\mathrm{Def}(\psi,\mathcal{U}\cup\mathcal{N}),\mathcal{N}\cup\{t\rightarrow\langle\mathcal{A}\rangle t_\psi\}) \\
 & & \text{where } \psi \text{ is not a literal and } \mathcal{A}\neq\Sigma_\varphi
\end{array}
$$

The rewriting rules will ensure that for any formula ψ in a coalition problem in $\mathsf{DSNF_{CL}}$ with $\mathrm{op}(\psi)=[\mathcal{A}]$ or $\mathrm{op}(\psi)=\langle\mathcal{A}\rangle$ we have that $\mathrm{args}(\psi)=(l)$ for some literal l.

Theorem 1. *Let $\varphi\in \mathit{WFF_{CL}}$ and let R be one of the rewriting systems R_1, R_2, or R_3. Let $\mathcal{C}_0,\mathcal{C}_1,\ldots$ be a sequence of coalition problems such that $\mathcal{C}_0=(\{t_\varphi\},\{t_\varphi\rightarrow\tau_0(\varphi)\},\emptyset)$ and $\mathcal{C}_{i+1}$ is obtained from $\mathcal{C}_i$ by applying a rewriting rule in R 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 rewriting rule can be applied to $\mathcal{C}_n$. Furthermore, $\mathcal{C}_n$ is a coalition problem in $\mathsf{DSNF_{CL}}$, the size of $\mathcal{C}_n$ is linear in the size of φ, and $\mathcal{C}_n$ is satisfiable if, and only if, φ is satisfiable.*

For the proof of Theorem 1 for the rewriting system R_1 see [17]. The proofs for the rewriting systems R_2 and R_3 are analogous. Note that while the size of coalition problem $\mathcal{C}_n$ in $\mathsf{DSNF_{CL}}$ that we obtain by exhaustively applying the rules of one of the rewriting systems is linear in the size of the given CL formula φ, the time required to compute $\mathcal{C}_n$ is also linear for R_1 but quadratic for R_2 and R_1, unless we assume that the formula φ is not given as sequence of symbols or as a tree structure but instead given as a formula DAG in which multiple occurrences of the same subformula are stored only once.

3.2 Resolution Calculus $\mathsf{RES_{CL}}$

Let $(\mathcal{I},\mathcal{U},\mathcal{N})$ be a coalition problem in $\mathsf{DSNF_{CL}}$; P, Q be conjunctions of literals; C, 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).

The resolution calculus $\mathsf{RES_{CL}}$, introduced in [17], consists of the following rules:

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

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

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

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

Note the connection between the axiomatisation of Coalition Logic and the inference rules of $\mathsf{RES_{CL}}$: **IRES1** and **GRES1** are consequences of the axioms for propositional logic, **CRES1** and, in particular, its side condition, follow from axiom **S**, **CRES3** and its side condition follow from schema **S'**, **CRES2** and **CRES4** follow from monotonicity, while **RW1** and **RW2** are based on axioms $\bot$ and $\top$ (together with $\langle\mathcal{A}\rangle$), respectively.

Definition 16. A *derivation* from a coalition problem in $\mathsf{DSNF_{CL}}$ $\mathcal{C}$ by a calculus $\mathcal{R}$ is a sequence $\mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_2, \ldots$ of coalition problems such that $\mathcal{C}_0 = \mathcal{C}$ and $\mathcal{C}_{i+1}$ is obtained from $\mathcal{C}_i$ by an application of a rule of $\mathcal{R}$ to $\mathcal{C}$.

In particular, 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

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

and conjunctions and disjunctions in R are always kept in the simplest form, that is, duplicate literals are removed, conjunctions (resp. disjunctions) with complementary literals are simplified to **false** (resp. **true**), and $R \notin \{\mathbf{true}, \mathbf{false} \to \varphi, \varphi \to \mathbf{true}, \varphi \to [\mathcal{A}]\mathbf{true}, \varphi \to \langle\mathcal{A}\rangle\mathbf{true}\}$, for any formula φ.

Definition 17. A *refutation* for a coalition problem in $\mathsf{DSNF_{CL}}$ $\mathcal{C} = (\mathcal{I}, \mathcal{U}, \mathcal{N})$ (by a calculus $\mathcal{R}$) 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 $\mathbf{false} \in \mathcal{I}_i$ or $\mathbf{false} \in \mathcal{U}_i$.

Definition 18. A derivation for a coalition problem in $\mathsf{DSNF_{CL}}$ $\mathcal{C}$ by calculus $\mathcal{R}$ *terminates* if, and only if, either a contradiction is derived or no new clauses can be derived by further application of the rules of $\mathcal{R}$.

3.3 Completeness Revisited

Definition 19. A calculus $\mathcal{R}$ is *complete* for a class $\mathfrak{C}$ of coalition problems in $\mathsf{DSNF_{CL}}$ iff for every unsatisfiable coalition problem $\mathcal{C} \in \mathfrak{C}$ there exists a refutation of $\mathcal{C}$ by $\mathcal{R}$. A calculus $\mathcal{R}$ is *sound* for a class $\mathfrak{C}$ of coalition problems in $\mathsf{DSNF_{CL}}$ iff no satisfiable coalition problem $\mathcal{C} \in \mathfrak{C}$ has a refutation of $\mathcal{C}$ by $\mathcal{R}$.

Definition 20. Let $\mathfrak{C}_i$, $1 \leq i \leq 3$ be the class of all coalition problems $\mathcal{C}$ in $\mathsf{DSNF_{CL}}$ such that $\mathcal{C}$ results from an application of rewriting rules of the rewriting system R_i.

It turns out that $\mathsf{RES_{CL}}$ is not complete for the classes $\mathfrak{C}_1$ and $\mathfrak{C}_2$. To show this we use a formalisation of the pigeon hole problem with three pigeons and two holes in Coalition Logic, given by the following formula φ^3_2:

$$\begin{array}{ll} & [\emptyset](\neg x^1_1 \vee \neg x^2_1) \wedge [\emptyset](\neg x^1_1 \vee \neg x^3_1) \wedge [\emptyset](\neg x^2_1 \vee \neg x^3_1) \\ \wedge & [\emptyset](\neg x^1_2 \vee \neg x^2_2) \wedge [\emptyset](\neg x^1_2 \vee \neg x^3_2) \wedge [\emptyset](\neg x^2_2 \vee \neg x^3_2) \\ \wedge & [1](x^1_1 \vee x^1_2) \wedge [2](x^2_1 \vee x^2_2) \wedge [3](x^3_1 \vee x^3_2) \end{array}$$

Here the proposition symbol x^i_j, $1 \leq i \leq 3, 1 \leq j \leq 2$ denotes that dove i is in hole j. Each conjunct $[i](x^i_1 \vee x^i_2)$, $1 \leq i \leq 3$, states that agent i can bring about that dove i is either in hole 1 or in hole 2, i.e., agent i controls into which hole dove i is placed. Each conjunct $[\emptyset](\neg x^i_j \vee \neg^{i'}_j)$, $1 \leq i < i' \leq 3, 1 \leq j \leq 2$, expresses that in every state that the agents can bring about no two doves can be in the same hole. The formula φ^3_2 is unsatisfiable. The transformation of φ^3_2 into a coalition problem in $\mathsf{DSNF_{CL}}$ using either rewriting system R_1 or R_2 is straightforward and results in Clauses (1) to (10) shown in Figure 1. We denote the corresponding coalition problem by $\mathcal{C}^3_2$. Clauses (11) to (70) in Figure 1 are all the non-tautological clauses that can be derived from $\mathcal{C}^3_2$ using $\mathsf{RES_{CL}}$ and there is no contradiction among these clauses. To understand better why this is the case, we focus on the following clauses from that derivation:

1.	t_0	$[\mathcal{I}]$
2.	$t_0 \rightarrow [\emptyset]\neg x_1^1 \vee \neg x_1^2$	$[\mathcal{N}]$
3.	$t_0 \rightarrow [\emptyset]\neg x_1^1 \vee \neg x_1^3$	$[\mathcal{N}]$
4.	$t_0 \rightarrow [\emptyset]\neg x_1^2 \vee \neg x_1^3$	$[\mathcal{N}]$
5.	$t_0 \rightarrow [\emptyset]\neg x_2^1 \vee \neg x_2^2$	$[\mathcal{N}]$
6.	$t_0 \rightarrow [\emptyset]\neg x_2^1 \vee \neg x_2^3$	$[\mathcal{N}]$
7.	$t_0 \rightarrow [\emptyset]\neg x_2^2 \vee \neg x_2^3$	$[\mathcal{N}]$
8.	$t_0 \rightarrow [1]x_1^1 \vee x_2^1$	$[\mathcal{N}]$
9.	$t_0 \rightarrow [2]x_1^2 \vee x_2^2$	$[\mathcal{N}]$
10.	$t_0 \rightarrow [3]x_1^3 \vee x_2^3$	$[\mathcal{N}]$

11.	$t_0 \rightarrow [1]x_2^1 \vee \neg x_1^2$	[$\mathcal{N}$, **CRES1**, $2, 8, \neg x_1^1$]
12.	$t_0 \rightarrow [2]x_2^2 \vee \neg x_1^1$	[$\mathcal{N}$, **CRES1**, $2, 9, \neg x_1^2$]
13.	$t_0 \rightarrow [1]x_2^1 \vee \neg x_1^3$	[$\mathcal{N}$, **CRES1**, $3, 8, \neg x_1^1$]
14.	$t_0 \rightarrow [3]x_2^3 \vee \neg x_1^1$	[$\mathcal{N}$, **CRES1**, $3, 10, \neg x_1^3$]
15.	$t_0 \rightarrow [2]x_2^2 \vee \neg x_1^3$	[$\mathcal{N}$, **CRES1**, $4, 9, \neg x_1^2$]
16.	$t_0 \rightarrow [3]x_2^3 \vee \neg x_1^2$	[$\mathcal{N}$, **CRES1**, $4, 10, \neg x_1^3$]
17.	$t_0 \rightarrow [1]x_1^1 \vee \neg x_2^2$	[$\mathcal{N}$, **CRES1**, $5, 8, \neg x_2^1$]
18.	$t_0 \rightarrow [2]x_1^2 \vee \neg x_2^1$	[$\mathcal{N}$, **CRES1**, $5, 9, \neg x_2^2$]
19.	$t_0 \rightarrow [1]x_1^1 \vee \neg x_2^3$	[$\mathcal{N}$, **CRES1**, $6, 8, \neg x_2^1$]
20.	$t_0 \rightarrow [3]x_1^3 \vee \neg x_2^1$	[$\mathcal{N}$, **CRES1**, $6, 10, \neg x_2^3$]
21.	$t_0 \rightarrow [2]x_1^2 \vee \neg x_2^3$	[$\mathcal{N}$, **CRES1**, $7, 9, \neg x_2^2$]
22.	$t_0 \rightarrow [3]x_1^3 \vee \neg x_2^2$	[$\mathcal{N}$, **CRES1**, $7, 10, \neg x_2^3$]
23.	$t_0 \rightarrow [1,3]x_1^3 \vee \neg x_1^2$	[$\mathcal{N}$, **CRES1**, $11, 20, x_2^1$]
24.	$t_0 \rightarrow [1,2]x_2^1 \vee \neg x_2^3$	[$\mathcal{N}$, **CRES1**, $11, 21, \neg x_1^2$]
25.	$t_0 \rightarrow [1]\neg x_1^2 \vee \neg x_2^2$	[$\mathcal{N}$, **CRES1**, $11, 5, x_2^1$]
26.	$t_0 \rightarrow [1]\neg x_1^2 \vee \neg x_2^3$	[$\mathcal{N}$, **CRES1**, $11, 6, x_2^1$]
27.	$t_0 \rightarrow [1,2]x_2^1 \vee x_2^2$	[$\mathcal{N}$, **CRES1**, $11, 9, \neg x_1^2$]
28.	$t_0 \rightarrow [1,2]x_2^2 \vee \neg x_2^3$	[$\mathcal{N}$, **CRES1**, $12, 19, \neg x_1^1$]
29.	$t_0 \rightarrow [2,3]x_1^3 \vee \neg x_1^1$	[$\mathcal{N}$, **CRES1**, $12, 22, x_2^2$]
30.	$t_0 \rightarrow [2]\neg x_1^1 \vee \neg x_2^1$	[$\mathcal{N}$, **CRES1**, $12, 5, x_2^2$]
31.	$t_0 \rightarrow [2]\neg x_1^1 \vee \neg x_2^3$	[$\mathcal{N}$, **CRES1**, $12, 7, x_2^2$]
32.	$t_0 \rightarrow [1,2]x_1^2 \vee \neg x_1^3$	[$\mathcal{N}$, **CRES1**, $13, 18, x_2^1$]
33.	$t_0 \rightarrow [1,3]x_2^1 \vee \neg x_2^2$	[$\mathcal{N}$, **CRES1**, $13, 22, \neg x_1^3$]
34.	$t_0 \rightarrow [1]\neg x_2^2 \vee \neg x_1^3$	[$\mathcal{N}$, **CRES1**, $13, 5, x_2^1$]
35.	$t_0 \rightarrow [1]\neg x_1^3 \vee \neg x_2^3$	[$\mathcal{N}$, **CRES1**, $13, 6, x_2^1$]
36.	$t_0 \rightarrow [1,3]x_2^1 \vee x_2^3$	[$\mathcal{N}$, **CRES1**, $13, 10, \neg x_1^3$]
37.	$t_0 \rightarrow [1,3]x_2^3 \vee \neg x_2^2$	[$\mathcal{N}$, **CRES1**, $14, 17, \neg x_1^1$]
38.	$t_0 \rightarrow [2,3]x_1^2 \vee \neg x_1^1$	[$\mathcal{N}$, **CRES1**, $14, 21, x_2^3$]
39.	$t_0 \rightarrow [3]\neg x_1^1 \vee \neg x_2^1$	[$\mathcal{N}$, **CRES1**, $14, 6, x_2^3$]
40.	$t_0 \rightarrow [3]\neg x_1^1 \vee \neg x_2^2$	[$\mathcal{N}$, **CRES1**, $14, 7, x_2^3$]
41.	$t_0 \rightarrow [1,2]x_1^1 \vee \neg x_1^3$	[$\mathcal{N}$, **CRES1**, $15, 17, x_2^2$]
42.	$t_0 \rightarrow [2,3]x_2^2 \vee \neg x_2^1$	[$\mathcal{N}$, **CRES1**, $15, 20, \neg x_1^3$]
43.	$t_0 \rightarrow [2]\neg x_2^1 \vee \neg x_1^3$	[$\mathcal{N}$, **CRES1**, $15, 5, x_2^2$]
44.	$t_0 \rightarrow [2]\neg x_1^3 \vee \neg x_2^3$	[$\mathcal{N}$, **CRES1**, $15, 7, x_2^2$]
45.	$t_0 \rightarrow [2,3]x_2^2 \vee x_2^3$	[$\mathcal{N}$, **CRES1**, $15, 10, \neg x_1^3$]
46.	$t_0 \rightarrow [2,3]x_2^3 \vee \neg x_2^1$	[$\mathcal{N}$, **CRES1**, $16, 18, \neg x_1^2$]
47.	$t_0 \rightarrow [1,3]x_1^1 \vee \neg x_1^2$	[$\mathcal{N}$, **CRES1**, $16, 19, x_2^3$]
48.	$t_0 \rightarrow [3]\neg x_2^1 \vee \neg x_1^2$	[$\mathcal{N}$, **CRES1**, $16, 6, x_2^3$]
49.	$t_0 \rightarrow [3]\neg x_1^2 \vee \neg x_2^2$	[$\mathcal{N}$, **CRES1**, $16, 7, x_2^3$]
50.	$t_0 \rightarrow [1,2]x_1^1 \vee x_1^2$	[$\mathcal{N}$, **CRES1**, $17, 9, \neg x_2^2$]
51.	$t_0 \rightarrow [1,3]x_1^1 \vee x_1^3$	[$\mathcal{N}$, **CRES1**, $19, 10, \neg x_2^3$]
52.	$t_0 \rightarrow [2,3]x_1^2 \vee x_1^3$	[$\mathcal{N}$, **CRES1**, $21, 10, \neg x_2^3$]
53.	$t_0 \rightarrow [1,3]\neg x_1^2$	[$\mathcal{N}$, **CRES1**, $23, 4, x_1^3$]
54.	$t_0 \rightarrow [1,2,3]x_2^2 \vee x_1^3$	[$\mathcal{N}$, **CRES1**, $23, 9, \neg x_1^2$]
55.	$t_0 \rightarrow [1,2]\neg x_2^3$	[$\mathcal{N}$, **CRES1**, $24, 6, x_2^1$]
56.	$t_0 \rightarrow [1,2,3]x_2^1 \vee x_1^3$	[$\mathcal{N}$, **CRES1**, $24, 10, \neg x_2^3$]
57.	$t_0 \rightarrow [2,3]\neg x_1^1$	[$\mathcal{N}$, **CRES1**, $29, 3, x_1^3$]
58.	$t_0 \rightarrow [1,2]\neg x_1^3$	[$\mathcal{N}$, **CRES1**, $32, 4, x_1^2$]
59.	$t_0 \rightarrow [1,2,3]x_1^2 \vee x_2^3$	[$\mathcal{N}$, **CRES1**, $32, 10, \neg x_1^3$]
60.	$t_0 \rightarrow [1,3]\neg x_2^2$	[$\mathcal{N}$, **CRES1**, $33, 5, x_2^1$]
61.	$t_0 \rightarrow [1,2,3]x_2^1 \vee x_1^2$	[$\mathcal{N}$, **CRES1**, $33, 9, \neg x_2^2$]
62.	$t_0 \rightarrow [1,2,3]x_1^1 \vee x_2^3$	[$\mathcal{N}$, **CRES1**, $34, 10, \neg x_1^3$]
63.	$t_0 \rightarrow [2,3]\neg x_2^1$	[$\mathcal{N}$, **CRES1**, $42, 5, x_2^2$]
64.	$t_0 \rightarrow [1,2,3]x_1^1 \vee x_2^2$	[$\mathcal{N}$, **CRES1**, $42, 8, \neg x_2^1$]
65.	$t_0 \rightarrow [1,2,3]x_2^2$	[$\mathcal{N}$, **CRES1**, $53, 9, \neg x_1^2$]
66.	$t_0 \rightarrow [1,2,3]x_1^3$	[$\mathcal{N}$, **CRES1**, $42, 10, \neg x_2^3$]
67.	$t_0 \rightarrow [1,2,3]x_2^1$	[$\mathcal{N}$, **CRES1**, $57, 8, \neg x_1^1$]
68.	$t_0 \rightarrow [1,2,3]x_2^3$	[$\mathcal{N}$, **CRES1**, $58, 10, \neg x_1^3$]
69.	$t_0 \rightarrow [1,2,3]x_1^2$	[$\mathcal{N}$, **CRES1**, $60, 9, \neg x_2^2$]
70.	$t_0 \rightarrow [1,2,3]x_1^1$	[$\mathcal{N}$, **CRES1**, $63, 8, \neg x_2^1$]

Figure 1: Derivation from $\mathcal{C}_2^3$ by $\mathsf{RES_{CL}}$

2.	$t_0 \to [\emptyset]\neg x_1^1 \vee \neg x_1^2$	$[\mathcal{N}]$	11.	$t_0 \to [1]x_2^1 \vee \neg x_1^2$	$[\mathcal{N}$, **CRES1**, $2, 8, \neg x_1^1]$
3.	$t_0 \to [\emptyset]\neg x_1^1 \vee \neg x_1^3$	$[\mathcal{N}]$	13.	$t_0 \to [1]x_2^1 \vee \neg x_1^3$	$[\mathcal{N}$, **CRES1**, $3, 8, \neg x_1^1]$
4.	$t_0 \to [\emptyset]\neg x_1^2 \vee \neg x_1^3$	$[\mathcal{N}]$	20.	$t_0 \to [3]x_1^3 \vee \neg x_2^1$	$[\mathcal{N}$, **CRES1**, $6, 10, \neg x_2^3]$
5.	$t_0 \to [\emptyset]\neg x_2^1 \vee \neg x_2^2$	$[\mathcal{N}]$	22.	$t_0 \to [3]x_1^3 \vee \neg x_2^2$	$[\mathcal{N}$, **CRES1**, $7, 10, \neg x_2^3]$
6.	$t_0 \to [\emptyset]\neg x_2^1 \vee \neg x_2^3$	$[\mathcal{N}]$	23.	$t_0 \to [1,3]x_1^3 \vee \neg x_1^2$	$[\mathcal{N}$, **CRES1**, $11, 20, x_2^1]$
7.	$t_0 \to [\emptyset]\neg x_2^2 \vee \neg x_2^3$	$[\mathcal{N}]$	33.	$t_0 \to [1,3]x_2^1 \vee \neg x_2^2$	$[\mathcal{N}$, **CRES1**, $13, 22, \neg x_1^3]$
8.	$t_0 \to [1]x_1^1 \vee x_2^1$	$[\mathcal{N}]$	53.	$t_0 \to [1,3]\neg x_1^2$	$[\mathcal{N}$, **CRES1**, $23, 4, x_1^3]$
9.	$t_0 \to [2]x_1^2 \vee x_2^2$	$[\mathcal{N}]$	60.	$t_0 \to [1,3]\neg x_2^2$	$[\mathcal{N}$, **CRES1**, $33, 5, x_2^1]$
10.	$t_0 \to [3]x_1^3 \vee x_2^3$	$[\mathcal{N}]$	65.	$t_0 \to [1,2,3]x_2^2$	$[\mathcal{N}$, **CRES1**, $53, 9, \neg x_1^2]$

Clause (8) expresses that in any state s where t_0 is true, there is a certain move $f_1(s)$ by agent 1 such that whatever moves $m_2(s)$ and $m_3(s)$ agents 2 and 3 make, respectively, in the resulting state $s' = \delta(s, (f_1(s), m_2(s), m_3(s)))$ the disjunction $x_1^1 \vee x_2^1$ is true. Clauses (2) and (3) express that in any state s where t_0 is true, whatever moves the three agents make, the disjunctions $\neg x_1^1 \vee \neg x_1^2$ and $\neg x_1^1 \vee \neg x_1^3$ are true.

Now consider clauses (11) and (13). If these two clauses would be part of the initial coalition problem, then it would be natural to interpret them in analogy to clause (8), that is, clause (11) relates to a move $g_1(s)$ by agent 1, potentially different to the move $f_1(s)$, and potentially resulting in a different set of outcomes once combined with whatever moves the other agents make, while clause (13) relates to a move $h_1(s)$ by agent 1, yet again potentially different to $f_1(s)$ and $g_1(s)$, and potentially resulting in a different set of outcomes.

However, clauses (11) and (13) are derived by resolving clause (8) with clauses (2) and (3), respectively. Thus, clause (11) combines the constraints on a CGM expressed by clauses (8) and (2), namely, that in any state s where t_0 is true, there is a certain move $f_1(s)$ by agent 1 such that whatever moves $m_2(s)$ and $m_3(s)$ agents 2 and 3 make, respectively, in the resulting state $s' = \delta(s, (f_1(s), f_2(s), f_3(s)))$ the disjunctions $x_1^1 \vee x_2^1$ and $\neg x_1^1 \vee \neg x_1^2$ are true, and therefore the disjunction $x_2^1 \vee \neg x_1^2$ is true in s'. So, clause (11) refers to the same action $f_1(s)$ by agent 1 and the same set of states that may result from that action. The same applies to clause (13).

Clauses (10), (20) and (22) have to be read analogously to clauses (8), (11) and (13), respectively, that is, they all refer to the same action $h_3(s)$ by agent 3 and the same set of states that may result from that action when combined with arbitrary moves by agents 1 and 2.

Clause (23) is derived by resolving clauses (11) and (20), and expresses that in any state s where t_0 is true, the moves $f_1(s)$ and $h_3(s)$ by agents 1 and 3, respectively, when combined with an arbitrary move $m_2(s)$ by agent 2, result in a state $s' = \delta(s, (f_1(s), m_2(s), h_3(s)))$ where $x_1^3 \vee \neg x_1^2$ is true. Clauses (33), (53) and (60) have to be read analogously, all referring to the actions $f_1(s)$ and $h_3(s)$ and the same set of resulting states. In particular, clause (60) expresses that in any state s where t_0 is true, the moves $f_1(s)$ and $h_3(s)$ by agents 1 and 3, respectively, when combined with an arbitrary move $m_2(s)$ by agent 2, result in a state $s' = \delta(s, (f_1(s), m_2(s), h_3(s)))$ where $\neg x_2^2$ is true.

Clause (65) is derived by resolving clauses (53) and (9). Clause (9) expresses that in any state s where t_0 is true, there is a certain move $g_2(s)$ by agent 2 such that whatever moves $m_1(s)$ and $m_3(s)$ agents 1 and 3 make, respectively, in the resulting state $s' = \delta(s, (m_1(s), g_2(s), m_3(s)))$ the disjunction $x_1^2 \vee x_2^2$ is true. In clause (65) the move $g_2(s)$ is now combined with the moves $f_1(s)$ and $h_3(s)$ by agents 1 and 3, respectively. Thus, clause (65) expresses that in any state s where t_0 is true, in the state $s' = \delta(s, (f_1(s), g_2(s), h_3(s))$ the proposition x_2^2 is true.

Now, if we accept that clauses (60) and (65) have the semantics described above, then it follows that the two clauses together imply that in any state s where t_0 is true, the move vector $(f_1(s), g_2(s), h_3(s))$ would lead to a successor $s' = \delta(s, (f_1(s), g_2(s), h_3(s))$ where both $\neg x_2^2$ and x_2^2 are true. Consequently, the two clauses together imply $t_0 \to [1, 2, 3]$**false** which in turn implies that t_0 cannot be true at any state in a CGM satisfying $\mathcal{C}_2^3$. As t_0 is true in the state s_0 it follows that $\mathcal{C}_2^3$ is not satisfiable.

However, using $\mathsf{RES_{CL}}$, clauses (60) and (65) cannot be resolved as the sets of agents involved in the two clauses are not disjoint, the side condition on **CRES1** is not satisfied. Therefore, the clause $t_0 \to [1, 2, 3]$**false** cannot be derived and we are not able to derive a contradiction by an application of **RW1** to that clause, resulting in a global clause $\neg t_0$ followed by a resolution step using **IRES1** with that global clause and the initial clause t_0.

A solution is to enforce that our normal form transformation only produces coalition clauses with a unique, new propositional symbol below a coalition modality. For example, clause (8), $t_0 \to [1]x_1^1 \vee x_2^1$, will be replaced by the coalition clause

$$t_0 \to [1]t_7 \quad (8')$$

together with the global clause

$$\neg t_7 \vee x_1^1 \vee x_2^1 \quad (8'')$$

Clause (8') expresses that in any state s where t_0 is true, there is a certain move $f_1(s)$ by agent 1 such that whatever moves $m_2(s)$ and $m_3(s)$ agents 2 and 3 make, respectively, in the resulting state $s' = \delta(s,(f_1(s), m_2(s), m_3(s)))$ the propositional symbol t_7 is true. Clause (8') is the only clause in a coalition problem that contains a positive occurrence of t_7 and this property will be preserved by $\mathsf{RES}_{\mathsf{CL}}$.

The rewriting system R_3 produces coalition problems in this normal form, namely, coalition problems in unit $\mathsf{DSNF}_{\mathsf{CL}}$. For the formula φ_2^3 we obtain the coalition problem $\mathcal{D}_2^3$ consisting of the clauses (1) to (19) in Figure 2. Clauses (20) to (41) in Figure 2 are a refutation of $\mathcal{D}_2^3$ using $\mathsf{RES}_{\mathsf{CL}}$. The refutation proceeds by a sequence of applications of **GRES1** to global clauses in $\mathcal{D}_2^3$ to derive Clause (30) which expresses that the propositional symbols $t_1, \ldots, t_9$ cannot all be true in the same state. Clause (30) is then used in a sequence of applications of **CRES1** and **CRES2** to the coalition clauses in $\mathcal{D}_2^3$ to derive Clause (39), $t_0 \to [1,2,3]\mathbf{false}$. An application of **RW1** then gives us the global clause $\neg t_0$ from which we derive a contradiction by resolving with the initial clause t_0. Note that in contrast to the derivation in Figure 1, at no point in the derivation are two derived coalition clauses resolved with each other. To prove the completeness of $\mathsf{RES}_{\mathsf{CL}}$ for coalition problems in unit $\mathsf{DSNF}_{\mathsf{CL}}$ we make not only make use of the refutational completeness of propositional resolution (Theorem 2) but also of the completeness of propositional resolution for consequence-finding (Theorem 3):

Theorem 2 (Completeness of classical propositional resolution [22])**.** *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* $\{(D \vee l), (D' \vee \neg l)\} \vdash (D \vee D')$.

Theorem 3 (Lee [16])**.** *Let N be a set of propositional clauses and C be clause such that N logically implies C. Then there is a derivation by propositional resolution, from N, of a clause D such that D subsumes C.*

The revised completeness proof uses a correspondence between Goranko and Shkatov's tableau-based decision procedure for ATL, restricted here to a weaker logic, and inferences by $\mathsf{RES}_{\mathsf{CL}}$. In particular, we show that for an unsatisfiable coalition problem in unit $\mathsf{DSNF}_{\mathsf{CL}}$, a closed tableau can be constructed and that applications of the state deletion rules in the tableau construction correspond to applications of the resolution inference rules to subsets of the clauses in a coalition problem in unit $\mathsf{DSNF}_{\mathsf{CL}}$ that will result in a refutation of the coalition problem.

To make our presentation here self-contained we repeat the description of Goranko and Shkatov's tableau-based decision procedure and its use for our purposes. For further details see [17].

The procedure consists of three different phases: construction, prestate elimination, and state elimination.

Graph Construction. 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 Tables 1a and 1b have been exhaustively applied. The first column in Table 1a (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. 22) 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

1.	t_0	$[\mathcal{I}]$
2.	$\neg t_1 \vee \neg x_1^1 \vee \neg x_1^2$	$[\mathcal{U}]$
3.	$\neg t_2 \vee \neg x_1^1 \vee \neg x_1^3$	$[\mathcal{U}]$
4.	$\neg t_3 \vee \neg x_1^2 \vee \neg x_1^3$	$[\mathcal{U}]$
5.	$\neg t_4 \vee \neg x_2^1 \vee \neg x_2^2$	$[\mathcal{U}]$
6.	$\neg t_5 \vee \neg x_2^1 \vee \neg x_2^3$	$[\mathcal{U}]$
7.	$\neg t_6 \vee \neg x_2^2 \vee \neg x_2^3$	$[\mathcal{U}]$
8.	$\neg t_7 \vee x_1^1 \vee x_2^1$	$[\mathcal{U}]$
9.	$\neg t_8 \vee x_1^2 \vee x_2^2$	$[\mathcal{U}]$
10.	$\neg t_9 \vee x_1^3 \vee x_2^3$	$[\mathcal{U}]$
11.	$t_0 \rightarrow [\emptyset] t_1$	$[\mathcal{N}]$
12.	$t_0 \rightarrow [\emptyset] t_2$	$[\mathcal{N}]$
13.	$t_0 \rightarrow [\emptyset] t_3$	$[\mathcal{N}]$
14.	$t_0 \rightarrow [\emptyset] t_4$	$[\mathcal{N}]$
15.	$t_0 \rightarrow [\emptyset] t_5$	$[\mathcal{N}]$
16.	$t_0 \rightarrow [\emptyset] t_6$	$[\mathcal{N}]$
17.	$t_0 \rightarrow [1] t_7$	$[\mathcal{N}]$
18.	$t_0 \rightarrow [2] t_8$	$[\mathcal{N}]$
19.	$t_0 \rightarrow [3] t_9$	$[\mathcal{N}]$
20.	$x_2^1 \vee \neg t_1 \vee \neg t_7 \vee \neg x_1^2$	$[\mathcal{U}, \textsf{GRES1}, 2, 8, \neg x_1^1]$
21.	$x_2^1 \vee \neg t_2 \vee \neg t_7 \vee \neg x_1^3$	$[\mathcal{U}, \textsf{GRES1}, 3, 8, \neg x_1^1]$
22.	$x_1^2 \vee \neg t_4 \vee \neg t_8 \vee \neg x_2^1$	$[\mathcal{U}, \textsf{GRES1}, 5, 9, \neg x_2^2]$
23.	$x_1^3 \vee \neg t_5 \vee \neg t_9 \vee \neg x_2^1$	$[\mathcal{U}, \textsf{GRES1}, 6, 10, \neg x_2^3]$
24.	$x_1^2 \vee \neg t_6 \vee \neg t_8 \vee \neg x_2^3$	$[\mathcal{U}, \textsf{GRES1}, 7, 9, \neg x_2^2]$
25.	$x_1^3 \vee \neg t_1 \vee \neg t_5 \vee \neg t_7 \vee \neg t_9 \vee \neg x_1^2$	$[\mathcal{U}, \textsf{GRES1}, 20, 23, x_2^1]$
26.	$x_1^2 \vee \neg t_2 \vee \neg t_4 \vee \neg t_7 \vee \neg t_8 \vee \neg x_1^3$	$[\mathcal{U}, \textsf{GRES1}, 21, 22, x_2^1]$
27.	$x_1^2 \vee x_1^3 \vee \neg t_6 \vee \neg t_8 \vee \neg t_9$	$[\mathcal{U}, \textsf{GRES1}, 24, 10, \neg x_2^3]$
28.	$x_1^3 \vee \neg t_1 \vee \neg t_5 \vee \neg t_6 \vee \neg t_7 \vee \neg t_8 \vee \neg t_9$	$[\mathcal{U}, \textsf{GRES1}, 25, 27, \neg x_1^2]$
29.	$\neg t_2 \vee \neg t_3 \vee \neg t_4 \vee \neg t_7 \vee \neg t_8 \vee \neg x_1^3$	$[\mathcal{U}, \textsf{GRES1}, 26, 4, x_1^2]$
30.	$\neg t_1 \vee \neg t_2 \vee \neg t_3 \vee \neg t_4 \vee \neg t_5 \vee \neg t_6 \vee \neg t_7 \vee \neg t_8 \vee \neg t_9$	$[\mathcal{U}, \textsf{GRES1}, 28, 29, x_1^3]$
31.	$t_0 \rightarrow [\emptyset] \neg t_2 \vee \neg t_3 \vee \neg t_4 \vee \neg t_5 \vee \neg t_6 \vee \neg t_7 \vee \neg t_8 \vee \neg t_9$	$[\mathcal{N}, \textsf{CRES2}, 11, 30, t_1]$
32.	$t_0 \rightarrow [\emptyset] \neg t_3 \vee \neg t_4 \vee \neg t_5 \vee \neg t_6 \vee \neg t_7 \vee \neg t_8 \vee \neg t_9$	$[\mathcal{N}, \textsf{CRES2}, 12, 31, t_2]$
33.	$t_0 \rightarrow [\emptyset] \neg t_4 \vee \neg t_5 \vee \neg t_6 \vee \neg t_7 \vee \neg t_8 \vee \neg t_9$	$[\mathcal{N}, \textsf{CRES2}, 13, 32, t_3]$
34.	$t_0 \rightarrow [\emptyset] \neg t_5 \vee \neg t_6 \vee \neg t_7 \vee \neg t_8 \vee \neg t_9$	$[\mathcal{N}, \textsf{CRES2}, 14, 33, t_4]$
35.	$t_0 \rightarrow [\emptyset] \neg t_6 \vee \neg t_7 \vee \neg t_8 \vee \neg t_9$	$[\mathcal{N}, \textsf{CRES2}, 15, 34, t_5]$
36.	$t_0 \rightarrow [\emptyset] \neg t_7 \vee \neg t_8 \vee \neg t_9$	$[\mathcal{N}, \textsf{CRES2}, 16, 35, t_6]$
37.	$t_0 \rightarrow [1] \neg t_8 \vee \neg t_9$	$[\mathcal{N}, \textsf{CRES1}, 17, 36, t_7]$
38.	$t_0 \rightarrow [1, 2] \neg t_9$	$[\mathcal{N}, \textsf{CRES1}, 18, 37, t_8]$
39.	$t_0 \rightarrow [1, 2, 3] \mathbf{false}$	$[\mathcal{N}, \textsf{CRES1}, 19, 38, t_9]$
40.	$\neg t_0$	$[\mathcal{U}, \textsf{RW1}, 39]$
41.	**false**	$[\mathcal{I}, \textsf{IRES1}, 1, 40, t_0]$

Figure 2: Derivation from $\mathcal{D}_2^3$ by calculus $\mathsf{RES}_{\mathsf{CL}}$

α	$\alpha_1, \ldots, \alpha_n$
$\neg\neg\varphi$	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	$\beta_1 \mid \ldots \mid \beta_n$
$\varphi_1 \vee \ldots \vee \varphi_n$	$\varphi_1 \mid \ldots \mid \varphi_n$
$\varphi_1 \wedge \ldots \wedge \varphi_n \to \psi$	$\neg\varphi_1 \mid \ldots \mid \neg\varphi_n \mid \psi$

(b) β-rules

Table 1: Tableau Rules

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 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**, $\neg$**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 21. 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 22. 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 $\{\alpha_1, \ldots, \alpha_n\} \subseteq \Delta$;
- If $\beta \in \Delta$, then $\beta_1 \in \Delta$, or $\ldots$, or $\beta_n \in \Delta$.

Definition 23. 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 $\Gamma \subseteq \Delta' \subset \Delta$.

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 [31]. 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 $\langle\langle\emptyset\rangle\rangle\Box$ operator. According to the α decomposition rules in [9], $\langle\langle\emptyset\rangle\rangle\Box\varphi$ should be decomposed into φ and $\langle\langle\emptyset\rangle\rangle\bigcirc\langle\langle\emptyset\rangle\rangle\Box\varphi$. The ATL formula $\langle\langle\emptyset\rangle\rangle\bigcirc\langle\langle\emptyset\rangle\rangle\Box\varphi$ corresponds to the CL^+ formula $[\emptyset]\langle\langle\emptyset\rangle\rangle\Box\varphi$, which explains the decomposition rule we give for $\langle\langle\emptyset\rangle\rangle\Box\varphi$. 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 state Δ obtained in step (1), if Δ does not contain any coalition formulae, add $[\Sigma_\Phi]\mathbf{true}$ to Δ;

(3) For each state Δ resulting from 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 $[\Sigma_\varphi]\mathbf{true}$ 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

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

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 $prestates(\Delta) = \{\Gamma \mid \Delta \xrightarrow{\sigma} \Gamma \text{ for some } \sigma \in D(\Delta)\}$. 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 axiomatisation 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.

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 on page 15); 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 24. 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 ([17]). *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.*

Tableaux for Coalition Problems. In order to use the tableau procedure outlined above to determine the satisfiability of a coalition problem in (unit) $\mathsf{DSNF_{CL}}$ and then to use the resulting method in our completeness proof for $\mathsf{RES_{CL}}$ on coalition problems in unit $\mathsf{DSNF_{CL}}$, we need to translate coalition problems into CL^+.

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 $\{simp(\bigvee_{l\in\mathcal{M}} l) \mid \mathcal{M} \in 2^{\Lambda_\mathcal{C}}\}\backslash\{\mathbf{true},\mathbf{false}\}$, 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 $\mathcal{C}$. 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.*

Let $\mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_n$ be a derivation by $\mathsf{RES_{CL}}$, with $\mathcal{C}_i = (\mathcal{I}_i,\mathcal{U}_i,\mathcal{N}_i)$ for each i, $0 \leq i \leq n$. We construct the initial tableau $\mathcal{T}_0^{\mathcal{C}_i}$ for $\mathcal{C}_i$ from a prestate containing the following set of formulae:

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

We thereby obtain a sequence $\mathcal{T}_0^{\mathcal{C}_0}, \ldots, \mathcal{T}_0^{\mathcal{C}_n}$. 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 and we denote by $\mathcal{T}^{\mathcal{C}_i}$ the final tableau obtained from $\mathcal{T}_0^{\mathcal{C}_i}$ after both deletion rules **E1** and **E2** have been exhaustively applied.

We will use the following lemmata from [17]:

Lemma 3. 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**.

Lemma 4. 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 $\Delta \in \mathcal{S}_0^\mathcal{C}$.

Lemma 5. Let $\mathcal{C} = (\mathcal{I},\mathcal{U},\mathcal{N})$ be a coalition problem in $\mathsf{DSNF_{CL}}$ and $C \to D$ 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 $D \in \Delta$.

We now establish two further lemmata before we state and prove the completeness of $\mathsf{RES_{CL}}$ for coalition problems in unit $\mathsf{DSNF_{CL}}$.

Lemma 6. Let $\mathcal{C} = (\mathcal{I},\mathcal{U},\mathcal{N})$ be a coalition problem in (unit) $\mathsf{DSNF_{CL}}$ and $\mathcal{T}_+^\mathcal{C}$ be the tableau for $\mathcal{C}$ after the rule **E1** has been exhaustively applied. If $\mathcal{I} \cup \mathcal{U}$ is unsatisfiable, then $\mathcal{T}_+^\mathcal{C}$ is closed.

Proof. Let $\mathcal{T}_0^\mathcal{C}$ be the initial tableau for $\mathcal{C}$ and let $\mathcal{S}_0^\mathcal{C}$ be set of states in $\mathcal{T}_0^\mathcal{C}$. Let Δ be an arbitrary initial state in $\mathcal{T}_0^\mathcal{C}$. According to Lemma 4, $\mathcal{U} \subseteq \Delta$. Also, by construction, $\mathcal{I} \subseteq \Delta$. If $\mathbf{false} \in \mathcal{I} \cup \mathcal{U}$, then $\mathbf{false} \in \Delta$ and Δ would be eliminated by an application of **E1** and Δ would not occur in $\mathcal{T}_+^\mathcal{C}$. Consequently, $\mathcal{T}_+^\mathcal{C}$ would be closed.

Assume that $\mathbf{false} \notin \mathcal{I} \cup \mathcal{U}$. Since Δ is downward closed, for each propositional clause $\bigvee_{j=1}^n l_j \in \mathcal{I} \cup \mathcal{U}$, Δ contains at least one literal l_j, $1 \leq j \leq n$. Let $\mathfrak{P}(\Delta) = \{l \in \Delta \mid C \in \mathcal{I} \cup \mathcal{U}$ and l is a literal in $C\}$. Suppose there is no propositional symbol $p \in \Pi$ such that $\{p, \neg p\} \subseteq \mathfrak{P}(\Delta)$. Then $\mathfrak{P}(\Delta) \cap \Pi$ would be a propositional model of $\mathcal{I} \cup \mathcal{U}$ contradicting that $\mathcal{I} \cup \mathcal{U}$ is unsatisfiable. Thus, there exists $p \in \Pi$ such that $\{p, \neg p\} \subseteq \mathfrak{P}(\Delta)$, Δ would be eliminated by an application of **E1**, it would therefore not occur in $\mathcal{T}_+^\mathcal{C}$, and $\mathcal{T}_+^\mathcal{C}$ would be closed. $\Box$

Lemma 7. Let $\mathcal{C} = (\mathcal{I},\mathcal{U},\mathcal{N})$ be an unsatisfiable coalition problem in unit $\mathsf{DSNF_{CL}}$ such that $\mathcal{I} \cup \mathcal{U}$ is satisfiable. Let $\Delta \in \mathcal{T}_+^\mathcal{C}$ be the first state to be eliminated by rule **E2** in the state elimination phase that will result in the final tableau $\mathcal{T}^\mathcal{C}$ for $\mathcal{C}$. Then we can derive a global clause $C \notin \mathcal{U}$ from $\mathcal{C}$.

Proof. Since $\mathcal{I} \cup \mathcal{U}$ is satisfiable, $\mathcal{T}^{\mathcal{C}}_{+}$ is not closed, but since $\mathcal{C}$ is unsatisfiable, the final tableau $\mathcal{T}^{\mathcal{C}}$ must be closed. Therefore, $\mathcal{T}^{\mathcal{C}}_{+}$ contains at least one state that can be deleted by an application of the deletion rule **E2**. Let Δ be the first state to which rule **E2** is applied. By definition of **E2**, Δ is deleted if there is a move vector $\sigma \in D(\Delta)$ such that there is no Δ' with $\Delta \overset{\sigma}{\longrightarrow} \Delta'$. Let $\mathfrak{L}(\Delta)$ be the ordered list of coalition formulae in Δ and for any coalition formula $\varphi \in \Delta$ let $pos(\varphi, \mathfrak{L}(\Delta))$ be the position of φ in $\mathfrak{L}(\Delta)$. From Lemma 4, all clause in $\mathcal{U}$ and in $\mathcal{N}$ are in Δ and each coalition formula in Δ corresponds to the right-hand side of coalition clause in $\mathcal{N}$. By Lemma 5, the right-hand side of coalition formulae are in the states where the left-hand side is satisfied. Therefore, by Lemmas 4 and 5, 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 $\mathcal{U} \cup \Theta_{\mathcal{C}} \cup \mathcal{P}^p_0 \cup \mathcal{P}^n_0 \cup \mathcal{N}$ where

$$\begin{aligned}
\mathcal{P}^+_0 &= \{p \mid [\mathcal{A}]p \in \mathcal{E}^+_0\}\\
\mathcal{E}^+_0 &= \{[\mathcal{A}]p \mid P \to [\mathcal{A}]p \in \mathcal{N}^+_0\}\\
\mathcal{N}^+_0 &= \{P \to [\mathcal{A}]p \mid P \to [\mathcal{A}]p \in \mathcal{N}, \Delta \models P, \sigma_a = pos([\mathcal{A}]p, \mathfrak{L}(\Delta)) \text{ for all } a \in \mathcal{A}\}\\
\mathcal{P}^-_0 &= \{p \mid \langle\mathcal{A}\rangle p \in \mathcal{E}^-_0\} \setminus \mathcal{P}^+_0\\
\mathcal{E}^+_0 &= \{\langle\mathcal{A}\rangle p \mid P \to \langle\mathcal{A}\rangle p \in \mathcal{N}^-_0\}\\
\mathcal{N}^-_0 &= \{P \to \langle\mathcal{A}\rangle p \mid P \to \langle\mathcal{A}\rangle p \in \mathcal{N}, \Delta \models P, \Sigma_{\mathcal{C}} \setminus \mathcal{A} \subseteq N(\sigma), neg(\sigma) = pos(\langle\mathcal{A}\rangle p, \mathfrak{L}(\Delta))\}
\end{aligned}$$

Recall that (a-i) for any $[\mathcal{A}]p, [\mathcal{A}']p' \in \mathcal{E}^+_0$ with $[\mathcal{A}]p \neq [\mathcal{A}']p'$ we have $\mathcal{A} \cap \mathcal{A}' = \emptyset$, (a-ii) $\mathcal{N}^-_0$, $\mathcal{E}^-_0$ and $\mathcal{P}^-_0$ are either all empty sets or all singleton sets, and (a-iii) for any $[\mathcal{A}]p \in \mathcal{E}^+_0$ and $\langle\mathcal{A}'\rangle p' \in \mathcal{E}^-_0$ we have $\mathcal{A} \subseteq \mathcal{A}'$.

Since Δ' is not in $\mathcal{T}^{\mathcal{C}}_{+}$, 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. As neither the tautologies in $\Theta_{\mathcal{C}}$ nor the formula in $\mathcal{N}$ contribute to propositional inconsistencies in Δ', the formula $\bigwedge_{C \in \mathcal{S}_0} C$ where $\mathcal{S}_0 = \mathcal{U}_0 \cup \mathcal{P}^+_0 \cup \mathcal{P}^-_0$, with $\mathcal{U}_0 = \mathcal{U}$, is unsatisfiable. Since both $\mathcal{U}$ and $\mathcal{P}^+_0 \cup \mathcal{P}^-_0$ are satisfiable, $\mathcal{S}_0$ can only be unsatisfiable if the clause $C = \bigvee_{p \in \mathcal{P}^+_0 \cup \mathcal{P}^-_0} \neg p$ is logically implied by $\mathcal{U}$. Therefore, by Theorem 3, there is a clause C' that can by derived from $\mathcal{U}$ that subsumes C. Since C' subsumes C, there must be sets $\mathcal{Q}^+_0 \subseteq \mathcal{P}^+_0$ and $\mathcal{Q}^-_0 \subseteq \mathcal{P}^-_0$ such that $C' = \bigvee_{p \in \mathcal{Q}^+_0 \cup \mathcal{Q}^-_0} \neg p$.

Linearly order $\mathcal{Q}^+_0 \cup \mathcal{Q}^-_0$ such that for non-empty $\mathcal{Q}^-_0$ its one element comes last. Let $(p_1, \ldots, p_m)$ be the resulting list. We associate each p_i in this list with a set of coalition clauses using the following functions:

$$\begin{aligned}
cl^+(p_i) &= \{P \to [\mathcal{A}]p_i \mid P \to [\mathcal{A}]p_i \in \mathcal{N}, \Delta \models P, \sigma_a = pos([\mathcal{A}]p_i, \mathfrak{L}(\Delta))\}\\
cl^-(p_i) &= \{P \to \langle\mathcal{A}\rangle p_i \mid P \to \langle\mathcal{A}\rangle p_i \in \mathcal{N}, \Delta \models P, \Sigma_{\mathcal{C}} \setminus \mathcal{A} \subseteq N(\sigma), neg(\sigma) = pos(\langle\mathcal{A}\rangle p, \mathfrak{L}(\Delta))\}\\
cl(p_i) &= \begin{cases} cl^+(p_i) & \text{if } cl^+(p_i) \neq \emptyset\\ cl^-(p_i) & \text{otherwise}\end{cases}
\end{aligned}$$

Note that by construction, $cl(p_i)$ is non-empty for every i, $1 \leq i \leq m$. With each $p_i \in \mathcal{Q}^+_0 \cup \mathcal{Q}^-_0$, $1 \leq i \leq m$, we then uniquely associate a clause κ_i, with $\kappa_i = P_i \to [\mathcal{A}_i]p_i$ or $\kappa_i = P_i \to \langle\mathcal{A}_i\rangle p_i$, in $cl(p_i)$. If $cl(p_i)$ contains more than one clause, then we can choose κ_i arbitrarily among the elements of $cl(p_i)$.

A refutation $\mathcal{S}_0, \ldots, \mathcal{S}_n$, with $n \in \mathbb{N}$, of $\mathcal{S}_0$ can then be constructed that satisfies the following properties:

(b-i) there is an index k, $0 \leq k < n$, such that $C' \in \mathcal{S}_k$ and for each i, $0 \leq i < k$, $\mathcal{S}_{i+1} = \mathcal{U}_{i+1} \cup \mathcal{P}^+_0 \cup \mathcal{P}^-_0$ and $\mathcal{U}_{i+1} = \mathcal{U}_i \cup \{D_{i+1}\}$ where D_{i+1} is a resolvent of two clauses in $\mathcal{U}_i$;

(b-ii) $\mathcal{S}_{k+1} = \mathcal{S}_k \cup \{D_{k+1}\}$ where D_{k+1} is a resolvent of C' and p_1;

(b-iii) for every i, $k+1 \leq i \leq k+m = n$, $\mathcal{S}_{i+1} = \mathcal{S}_i \cup \{D_{i+1}\}$ where D_{i+1} is a resolvent of D_i and p_{i+1-k}, that is, the clause $C', D_{k+1}, \ldots, D_n$ form a linear input derivation.

Property (b-i) reflects that C' can be derived from $\mathcal{U}$ alone; the slightly complicated formulation of the property is necessitated by the fact that $\mathcal{U}$ and $\mathcal{P}^+_0 \cup \mathcal{P}^-_0$ may not be disjoint. Clearly, in the derivation of C' we can make use of unit clauses that occur in both $\mathcal{U}$ and $\mathcal{P}_0^+ \cup \mathcal{P}_0^-$ but not those that only occur in $\mathcal{P}_0^+ \cup \mathcal{P}_0^-$.

That a refutation satisfying the last two properties exists follows from the fact that C' is a negative clause and that we can 'eliminate' the literals in C' one by one using unit resolution steps with positive unit clauses in $\mathcal{Q}_0^+ \cup \mathcal{Q}_0^-$, and that we can do so in an arbitrary order.

We inductively construct a derivation $\mathcal{C} = \mathcal{C}_0, \ldots, \mathcal{C}_n$ that satisfies the following properties:

(c-i) for all i, $0 \leq i \leq k$, $\mathcal{C}_i = (\mathcal{I}, \mathcal{U}_i, \mathcal{N})$;

(c-ii) $\mathcal{C}_{k+1} = (\mathcal{I}, \mathcal{U}_k, \mathcal{N}_{k+1})$ where $\mathcal{N}_{k+1} = \mathcal{N} \cup \{P_1 \to [\mathcal{A}_1]D_{k+1}\}$ if $\kappa_1 = P_1 \to [\mathcal{A}_1]p_1$ or $\mathcal{N}_{k+1} = \mathcal{N} \cup \{P_1 \to \langle\mathcal{A}_1\rangle D_{k+1}\}$ if $\kappa_1 = P_1 \to \langle\mathcal{A}_1\rangle p_1$;

(c-iii) for all i, $k+1 \leq i \leq n-2$, $\mathcal{C}_{i+1} = (\mathcal{I}, \mathcal{U}_k, \mathcal{N}_{i+1})$ with $\mathcal{N}_{i+1} = \mathcal{N}_i \cup \{P_1 \wedge \ldots \wedge P_{i+1-k} \to [\mathcal{A}_1 \cup \cdots \cup \mathcal{A}_{i+1-k}]D_{i+1}\}$;

(c-iv) $\mathcal{C}_n = (\mathcal{I}, \mathcal{U}_k, \mathcal{N}_n)$ where $\mathcal{N}_n$ contains either a clause of the form $P_1 \wedge \ldots \wedge P_m \to [\mathcal{A}_1 \cup \cdots \cup \mathcal{A}_m]\mathbf{false}$ or $P_1 \wedge \ldots \wedge P_m \to \langle\mathcal{A}_m \setminus (\mathcal{A}_1 \cup \cdots \cup \mathcal{A}_{m-1})\rangle\mathbf{false}$.

In the base case, $i = 0$, $\mathcal{C}_0 = \mathcal{C} = (\mathcal{I}, \mathcal{U}, \mathcal{N})$ and as $\mathcal{U} = \mathcal{U}_0$, property (c-i) is satisfied.

For the induction step, let $\mathcal{C}_0, \ldots, \mathcal{C}_i$, $i \geq 0$, be the derivation already constructed and we assume that properties (c-i) to (c-iii) hold for that derivation.

1. If $i < k$, then $\mathcal{S}_{i+1} = \mathcal{S}_i \cup \{D \vee D'\}$ where $D \vee D'$ is obtained by resolving clauses $D \vee l$ and $D' \vee \neg l \in \mathcal{U}_i$. Since $\mathcal{C}_i = (\mathcal{I}, \mathcal{U}_i, \mathcal{N})$, $D \vee l$ and $\theta(D' \vee \neg l)$ are both global clauses in $\mathcal{C}_i$. These can be resolved using **GRES1** resulting in the global clause $D \vee D'$. We define $\mathcal{C}_{i+1} = (\mathcal{I}, \mathcal{U}_i \cup \{D \vee D'\}, \mathcal{N})$. Then, $\mathcal{C}_{i+1} = (\mathcal{I}, \mathcal{U}_{i+1}, \mathcal{N})$ and property (c-i) holds for $\mathcal{C}_1, \ldots, \mathcal{C}_{i+1}$.

2. If $i = k$, then $\mathcal{S}_{i+1} = \mathcal{S}_i \cup \{D_{i+1}\}$ where D_{i+1} is obtained by resolving the clause $C' = D_i = (D_{i+1} \vee \neg p_1)$ with the unit clause p_1 from $\mathcal{Q}_0^p \cup \mathcal{Q}_0^n \subseteq \mathcal{S}_i$.

   For the construction of $\mathcal{C}_{i+1}$ there are two possibilities depending on the form κ_1:

   (a) $\kappa_1 = P_1 \to [\mathcal{A}_1]p_1 \in \mathcal{N}_0^+$. Then let $\mathcal{C}_{i+1} = (\mathcal{I}, \mathcal{U}_k, \mathcal{N} \cup \{P_1 \to [\mathcal{A}_1]D_{i+1}\})$, where $P_1 \to [\mathcal{A}_1]D_{i+1}$ is obtained by an application of **CRES2** to the global clause $D_{i+1} \vee \neg p_1 \in \mathcal{U}_k$ and the coalition clause $\kappa_1 \in \mathcal{N}_0^+ \subseteq \mathcal{N}$. Property (c-ii) holds for $\mathcal{C}_1, \ldots, \mathcal{C}_{i+1}$. If $i+1 = n$, then $D_{i+1} = \mathbf{false}$, $\mathcal{C}_{i+1} = \mathcal{C}_n$ contains $P_1 \to [\mathcal{A}_1]\mathbf{false}$, and Property (c-iv) holds for $\mathcal{C}_1, \ldots, \mathcal{C}_n$.

   (b) $\kappa_1 = P_1 \to \langle\mathcal{A}_1\rangle p_1 \in \mathcal{N}_0^-$. Then let $\mathcal{C}_{i+1} = (\mathcal{I}, \mathcal{U}_k, \mathcal{N} \cup \{P_1 \to \langle\mathcal{A}_1\rangle D_{i+1}\})$, where $P_1 \to \langle\mathcal{A}_1\rangle D_{i+1}$ is obtained by an application of **CRES4** to the global clause $D_{i+1} \vee \neg p_1 \in \mathcal{U}_k$ and the coalition clause $\kappa_1 \in \mathcal{N}_0^- \subseteq \mathcal{N}$. Note that by construction of $(p_1, \ldots, p_m)$, κ_1 is only a negative coalition clause if $m = 1$ and therefore this is the last step in the refutation and $D_{i+1} = \mathbf{false}$. Property (c-iv) holds for $\mathcal{C}_1, \ldots, \mathcal{C}_{i+1} = \mathcal{C}_n$.

3. If $i \geq k+1$, then $\mathcal{S}_{i+1} = \mathcal{S}_i \cup \{D_{i+1}\}$ where D_{i+1} is obtained by resolving the clause $D_i = (D_{i+1} \vee \neg p_{i+1-k})$ in $\mathcal{S}_i$ with the unit clause p_{i+1-k} from $\mathcal{Q}_0^p \cup \mathcal{Q}_0^n \subseteq \mathcal{S}_i$.

   By the induction hypothesis $\mathcal{C}_i = (\mathcal{I}, \mathcal{U}_k, \mathcal{N}_i)$ where $\mathcal{N}_i = \mathcal{N}_{i-1} \cup \{P_1 \wedge \ldots \wedge P_{i-k} \to [\mathcal{A}_1 \cup \cdots \cup \mathcal{A}_{i-k}]D_i$ and $D_i = D_{i+1} \vee \neg p_{i+1-k}$. For the construction of $\mathcal{C}_{i+1}$ there are two possibilities depending on the form κ_{i+1-k}:

   (a) $\kappa_{i+1-k} = P_{i+1-k} \to [\mathcal{A}_{i+1-k}]p_{i+1-k}$. Let $\mathcal{C}_{i+1} = (\mathcal{I}, \mathcal{U}_k, \mathcal{N}_i \cup \{P_1 \wedge \ldots \wedge P_{i+1-k} \to [\mathcal{A}_1 \cup \cdots \cup \mathcal{A}_{i-k} \cup \mathcal{A}_{i+1-k}]D_{i+1}\})$, where $P_1 \wedge \ldots \wedge P_{i+1-k} \to [\mathcal{A}_1 \cup \cdots \cup \mathcal{A}_{i-k} \cup \mathcal{A}_{i+1-k}]D_{i+1}$ is obtained by an application of **CRES1** to $P_1 \wedge \ldots \wedge P_{i-k} \to [\mathcal{A}_1 \cup \cdots \cup \mathcal{A}_{i-k}]D_i$ and κ_{i+1-k}; this inference is possible since by property (a-i), $\mathcal{A}_{i+1-k} \cap (\mathcal{A}_1 \cup \cdots \cup \mathcal{A}_{i-k}) = \emptyset$. Property (c-iii) holds for $\mathcal{C}_1, \ldots, \mathcal{C}_{i+1}$. If $i+1 = n$, then $D_{i+1} = \mathbf{false}$, $\mathcal{C}_{i+1} = \mathcal{C}_n$ contains $P_1 \wedge \ldots \wedge P_{i+1-k} \to [\mathcal{A}_1 \cup \cdots \cup \mathcal{A}_{i-k} \cup \mathcal{A}_{i+1-k}]\mathbf{false}$, and Property (c-iv) holds for $\mathcal{C}_1, \ldots, \mathcal{C}_n$.

   (b) $\kappa_{i+1-k} = P_{i+1-k} \to \langle\mathcal{A}_{i+1-k}\rangle p_{i+1-k}$. Let $\mathcal{C}_{i+1} = (\mathcal{I}, \mathcal{U}_k, \mathcal{N}_i \cup \{P_1 \wedge \ldots \wedge P_{i+1-k} \to \langle\mathcal{A}_{i+1-k} \setminus (\mathcal{A}_1 \cup \cdots \cup \mathcal{A}_{i-k})\rangle D_{i+1}\})$, where $P_1 \wedge \ldots \wedge P_{i+1-k} \to \langle\mathcal{A}_{i+1-k} \setminus (\mathcal{A}_1 \cup \cdots \cup \mathcal{A}_{i-k} \cup \mathcal{A}_{i+1-k})\rangle D_{i+1}$ is obtained by an application of **CRES3** to $P_1 \wedge \ldots \wedge P_{i-k} \to [\mathcal{A}_1 \cup \cdots \cup \mathcal{A}_{i-k}]D_i$ and κ_{i+1-k}; this inference is possible since by property (a-iii), $(\mathcal{A}_1 \cup \cdots \cup \mathcal{A}_{i-k}) \subseteq \mathcal{A}_{i+1-k}$. Property (c-iv) holds for $\mathcal{C}_1, \ldots, \mathcal{C}_{i+1} = \mathcal{C}_n$.

In the end we obtain a derivation $\mathcal{C}_0,\ldots,\mathcal{C}_n$ where $\mathcal{C}_n$ contains either $P_1\wedge\ldots\wedge P_m \to [\mathcal{A}_1\cup\cdots\cup\mathcal{A}_m]\mathbf{false}$, to which we can apply the rewrite rule **RW1**, or $P_1\wedge\ldots\wedge P_m \to \langle\mathcal{A}_m\setminus(\mathcal{A}_1\cup\cdots\cup\mathcal{A}_{m-1})\rangle\mathbf{false}$, to which we can apply the rewrite rule **RW2**. In both cases we obtain the global clause $C'' = \neg P_1 \vee \ldots \vee \neg P_m$.

We claim that $C'' \notin \mathcal{U}$. Assume the opposite. From Lemma 4, global clauses are in every state of $\mathcal{T}_+^{\mathcal{C}}$. So, the state Δ contains C''. As Δ is a minimal downward saturated set, Δ entails $\neg P_i$ for some i, $1 \leq i \leq m$. Δ also contains all formulae in $\mathcal{N}_0^+$, $\mathcal{E}_0^+$ $\mathcal{N}_0^-$, and $\mathcal{E}_0^-$. By construction, either $P_i \to [\mathcal{A}_i]p_i \in \mathcal{N}_0^+$ or $P_i \to \langle\mathcal{A}_i\rangle p_i \in \mathcal{N}_0^-$. By definition of $\mathcal{N}_0^+$ and $\mathcal{N}_0^-$, $\Delta \models P_i$. However, $\Delta \models \neg P_i$ and $\Delta \models P_i$ implies that Δ is inconsistent and should have been removed by an application of rule **E1**. This contradicts our assumption that Δ is in $\mathcal{T}_+^{\mathcal{C}}$. ☐

Theorem 5 (Completeness of $\mathsf{RES_{CL}}$). *Let $\mathcal{C} = (\mathcal{I},\mathcal{U},\mathcal{N})$ be an unsatisfiable coalition problem in unit $\mathsf{DSNF_{CL}}$. Then there is a refutation for $\mathcal{C}$ using the inference rules* **IRES1**, **GRES1**, **CRES1-4**, *and* **RW1-2**.

Proof. Let $\mathcal{C} = (\mathcal{I},\mathcal{U},\mathcal{N})$ be an unsatisfiable coalition problem in unit $\mathsf{DSNF_{CL}}$. Firstly, if $\mathcal{C}$ is unsatisfiable and only if $\mathcal{C}$ is unsatisfiable, by Theorem 4, we have that $\mathcal{T}^{\mathcal{C}}$ is closed. Obviously, if $\mathcal{C}$ is unsatisfiable, every coalition problem in a derivation from $\mathcal{C}$ is also unsatisfiable. We show that if $\mathcal{C}$ is unsatisfiable, then we can construct a refutation $\mathcal{R}_{\mathcal{C}} = \mathcal{C}_0,\ldots,\mathcal{C}_n$, $n \in \mathbb{N}$.

Let $\mathcal{C}_0^0 = \mathcal{C}$. If $\mathcal{T}_+^{\mathcal{C}_0^0}$ is closed, then all initial states in $\mathcal{T}_+^{\mathcal{C}_0^0}$ have been removed by applications of **E1** which means that $\mathcal{I}\cup\mathcal{U}$ is unsatisfiable. By Lemma 3 there exists a refutation $\mathcal{C}_0^0,\ldots,\mathcal{C}_0^{m_0}$ for $\mathcal{C}_0^0$ using only the inference rules **IRES1** and **GRES1**. If $\mathcal{T}_+^{\mathcal{C}_0^0}$ is not closed, then by Lemma 7, we can construct a derivation $\mathcal{C}_0^0,\ldots,\mathcal{C}_0^{m'_0} = \mathcal{C}_1^0 = (\mathcal{I},\mathcal{U}_1,\mathcal{N}_1)$ such that there is a global clause κ with $\kappa \in \mathcal{U}_1$ but $\kappa \notin \mathcal{U}$. Depending on whether $\mathcal{T}_0^{\mathcal{C}_1^0}$ is closed, we proceed as for $\mathcal{C}_0^0$ in the construction of the derivation.

We continue this construction until we derive a coalition problem $\mathcal{C}_i^0$ for which $\mathcal{T}_+^{\mathcal{C}_i^0}$ is closed and for which we can then complete the construction of the refutation using Lemma 3. We know that we will eventually derive such a coalition problem $\mathcal{C}_i^0$ as the number of global clauses is finite, that is, it cannot indefinitely be the case that $\mathcal{T}_+^{\mathcal{C}_i^0}$ is open while the final tableau $\mathcal{T}^{\mathcal{C}_i^0}$ is closed. On the other hand if we were to derive a coalition problem $\mathcal{C}_i^0$ such that both $\mathcal{T}_+^{\mathcal{C}_i^0}$ and $\mathcal{T}^{\mathcal{C}_i^0}$ are open, then by Theorem 4 $\mathcal{C}_i^0 = (\mathcal{I},\mathcal{U}_i,\mathcal{N}_i)$ is satisfiable. As $\mathcal{U} \subseteq \mathcal{U}_i$ and $\mathcal{N} \subseteq \mathcal{N}$ this contradicts the assumption that $\mathcal{C} = (\mathcal{I},\mathcal{U},\mathcal{N})$ is unsatisfiable. ☐

4 Ordered Resolution for Coalition Logic

Ordering refinements are a commonly used approach to reduce the search space of resolution for classical propositional and first-order logic. They are utilised by all state-of-the-art resolution-based theorem provers for first-order logic, including E [24, 23], SPASS [30, 25], and Vampire [14, 27]. Ordering refinements have also been used in the context of modal and temporal logics including PLTL [13], and CTL [32].

An *atom ordering* for $\mathsf{RES}_{\mathsf{CL}}^{\succ}$ is a well-founded and total ordering $\succ$ on the set Π. The ordering $\succ$ is extended to literals such that for each $p \in \Pi$, $\neg p \succ p$, and for each $q \in \Pi$ such that $q \succ p$ then $q \succ \neg p$ and $\neg q \succ \neg p$.

A literal l is (*strictly*) *maximal* with respect to a propositional disjunction C iff for every literal l' in C, $l' \not\succ l$ ($l' \not\succeq l$). Note that as long as a propositional disjunction C does not contain duplicate literals, any literal l that is maximal with respect to C is also strictly maximal with respect to C.

We could use the ordering $\succ$ to restrict the applicability of the rules **IRES1**, **GRES1**, **CRES1** to **CRES4** so that a rule is only applicable if and only if the literal l in $C \vee l$ is maximal with respect C and the literal $\neg l$ in $D \vee \neg l$ is maximal with respect to D.

One would normally expect that the calculus we obtain by way of this restriction is complete for any ordering, see, for example [4, 12, 13, 33].

However, it turns out that such a restriction would render our calculus incomplete, even on coalition problems in unit $\mathsf{DSNF_{CL}}$. Consider the following example:

1.	t_0	$[\mathcal{I}]$
2.	$\neg t_1 \vee p$	$[\mathcal{U}]$
3.	$\neg t_1 \vee \neg p$	$[\mathcal{U}]$
4.	$t_0 \rightarrow [1] t_1$	$[\mathcal{N}]$

Assume that the ordering on propositional symbols is $t_0 \succ t_1 \succ p$. Then the only inferences possible are the following:

5.	$t_0 \rightarrow [1] p$	$[\mathcal{N}$, **CRES2**, $2, 4, t_1]$
6.	$t_0 \rightarrow [1] \neg p$	$[\mathcal{N}$, **CRES2**, $3, 4, t_1]$

An resolution inference using **CRES1** with Clauses (5) and (6) as premises is not possible as the sets of agents in the two clauses is not disjoint. Using the unrefined calculus $\mathsf{RES}_{\mathsf{CL}}$ or using a different ordering, namely, $p \succ t_1 \succ t_0$ allows us to construct a refutation for this example:

5'.	$\neg t_1$	$[\mathcal{U}$, **GRES1**, $2, 3, p]$
6'.	$t_0 \rightarrow [1]$**false**	$[\mathcal{N}$, **CRES2**, $4, 5', t_1]$
7'.	$\neg t_0$	$[\mathcal{U}$, **RW1**, $6']$
8'.	**false**	$[\mathcal{U}$, **IRES1**, $1, 7', t_0]$

As the example illustrates, the incompleteness of a naive ordering refinement of $\mathsf{RES}_{\mathsf{CL}}$ on coalition problems in unit $\mathsf{DSNF}_{\mathsf{CL}}$ is closely related to the incompleteness of $\mathsf{RES}_{\mathsf{CL}}$ on coalition problems in $\mathsf{DSNF}_{\mathsf{CL}}$. We have observed that the latter is related to the fact that a derived clause does not accurately reflect the constraints on the agents' moves that are inherited from the premises that were used in the resolution step. While unit $\mathsf{DSNF}_{\mathsf{CL}}$ is a sufficient approximation for $\mathsf{RES}_{\mathsf{CL}}$ to be complete, for an ordering refinement of $\mathsf{RES}_{\mathsf{CL}}$ we need a better representation of the constraints imposed by the clauses in a coalition problem. To this end we introduce the notions of *coalition vector*, *positive $\mathcal{A}$-coalition vector* and *negative $\mathcal{A}$-coalition vector*, and use these to replace coalition modalities in coalition clauses.

Definition 25. Let $|\Sigma| = k$. A *coalition vector* $\vec{c}$ is a k-tuple such that for every a, $1 \leq a \leq k$, $\vec{c}[a]$ is either an integer number not equal to zero or the symbol $*$ and for every a, a', $1 \leq a < a' \leq k$, if $\vec{c}[a] < 0$ and $\vec{c}[a'] < 0$ then $\vec{c}[a] = \vec{c}[a']$.

A coalition vector $\vec{c}$ is *negative* if $\vec{c}[a] < 0$ for some a, $1 \leq a \leq k$. Otherwise, $\vec{c}$ is *positive.* We denote by $\vec{c}^{\,+}$ that $\vec{c}$ is positive and by $\vec{c}^{\,-}$ that $\vec{c}$ is negative.

We define the *absolute value* $\mathrm{abs}(\vec{c_1})$ of a coalition vector $\vec{c_1}$ is the positive coalition vector $\vec{c_2}$ with $\vec{c_2}[a] = |\vec{c_1}[a]|$ for every a, $1 \leq a \leq k$.

The *set of free agents* $\mathrm{FA}(\vec{c})$ *of a coalition vector* $\vec{c}$ is the set of indices a with $\vec{c}[a] = *$: $\mathrm{FA}(\vec{c}) = \{a \mid 1 \leq a \leq k \wedge \vec{c}[a] = *\}$.

In analogy, the *set of free agents* $\mathrm{FA}(\sigma_{\mathcal{A}})$ of a $\mathcal{A}$-move is the set of agents a with $\sigma_{\mathcal{A}}[a] = *$: $\mathrm{FA}(\vec{c}) = \{a \mid 1 \leq a \leq k \wedge \sigma_{\mathcal{A}}[a] = *\}$.

The *set of bound agents* $\mathrm{BA}(\vec{c})$ *of a coalition vector* $\vec{c}$ is the set of indices a with $\vec{c}[a] > 0$: $\mathrm{BA}(\vec{c}) = \{a \mid 1 \leq a \leq k \wedge \vec{c}[a] > 0\}$.

The *set of reactive agents* $\mathrm{RA}(\vec{c})$ *of a coalition vector* $\vec{c}$ is the set of indices a with $\vec{c}[a] < 0$: $\mathrm{RA}(\vec{c}) = \{a \mid 1 \leq a \leq k \wedge \vec{c}[a] < 0\}$.

For example, given $\Sigma = \{1, \ldots, 6\}$, $\vec{c_1} = (1, *, *, 3, *, 1, *)$, $\vec{c_2} = (*, -2, *, *, *, -2)$, and $\vec{c}_3 = (1, -2, *, 3, *, -2)$ are coalition vectors, $\vec{c_1}$ is positive, while $\vec{c_2}$ and $\vec{c}_3$ are negative. For $\vec{c}_3$ we have $\mathrm{FA}(\vec{c}_3) = \{3, 5\}$, $\mathrm{BA}(\vec{c}_3) = \{1, 4\}$, and $\mathrm{RA}(\vec{c}_3) = \{2, 6\}$.

Definition 26. A move vector σ *extends* a (positive or negative) coalition vector $\vec{c}$, denoted by $\vec{c} \sqsubseteq \sigma$ or $\sigma \sqsupseteq \vec{c}$, if $\sigma(a) = \vec{c}[a]$ for every $a \in \mathrm{BA}(\vec{c})$.

Definition 27. Let $\mathcal{A}$, $\emptyset \subseteq \mathcal{A} \subseteq \Sigma$, be a coalition of agents with $|\Sigma| = k$ and let $i > 0$ be a natural number. A *positive $\mathcal{A}$-coalition vector $\vec{c}^{\,i}_{\mathcal{A}}$ with index i* (*positive $\mathcal{A}$-coalition vector* for short) is a k-tuple such that $\vec{c}^{\,i}_{\mathcal{A}}(a) = i$ for every $a \in \mathcal{A}$ and $\vec{c}^{\,i}_{\mathcal{A}}(a') = *$ for every $a' \notin \mathcal{A}$.

Definition 28. Let $\mathcal{A}$, $\emptyset \subseteq \mathcal{A} \subseteq \Sigma$, be a coalition of agents with $|\Sigma| = k$ and let $i > 0$ be a natural number. A *negative $\mathcal{A}$-coalition vector $\vec{c}_{\mathcal{A}}^{\,-i}$ with index i* (*negative $\mathcal{A}$-coalition vector* for short) is a k-tuple such that $\vec{c}_{\mathcal{A}}^{\,-i}(a') = -i$ for every $a' \notin \mathcal{A}$ and $\vec{c}_{\mathcal{A}}^{\,i}(a) = *$ for every $a \in \mathcal{A}$.

Definition 29. The set $\mathsf{WFF}_{\mathsf{VCL}}$ of VCL *formulae* is inductively defined as follows.

- if p is a propositional symbols in Π, then p and $\neg p$ (negation) are VCL formulae;
- if φ and ψ are VCL formulae, then so are $(\varphi \to \psi)$ (implication);
- if φ_i, $1 \leq i \leq n$, $n \in \mathbb{N}_0$, are VCL formula, then so are $(\varphi_1 \wedge \ldots \wedge \varphi_n)$ (conjunction), also written $\bigwedge_{i=1}^{n} \varphi_i$, and $(\varphi_1 \vee \ldots \vee \varphi_n)$ (disjunction), also written $\bigvee_{i=1}^{n} \varphi_i$; and
- if $\vec{c}$ is a coalition vector and φ is a VCL formula, then so is $\vec{c}\varphi$.

In order to define the semantics of $\mathsf{WFF}_{\mathsf{VCL}}$ formulae we can reuse Concurrent Game Frames, but need to extend Concurrent Game Models with *choice functions* that give meaning to coalition vectors.

Definition 30. A *Concurrent Game Model with Choice Functions* ($\mathrm{CGM}_{\mathrm{CF}}$) is a tuple $\mathcal{M} = (\mathcal{F}, \Pi, \pi, F^+, F^-)$, where

- $\mathcal{F} = (\Sigma, \mathcal{S}, s_0, d, \delta)$ is a CGF;
- Π is the set of propositional symbols;
- $\pi : \mathcal{S} \longrightarrow 2^{\Pi}$ is a valuation function;
- $F^+ = \{f^i \mid i \in \mathbb{N}\}$ is a set of functions such that $f^i : \mathcal{S} \times \Sigma \longrightarrow \mathbb{N}_0$ and $f^i(s, a) \in D(a, s)$ for every $i \in \mathbb{N}$, $a \in \Sigma$, and $s \in \mathcal{S}$;
- $F^- = \{g_n^i \mid i \in \mathbb{N}, n \in \mathbb{N}_0, \text{ and } n \leq |\Sigma|\}$ is a set of functions such that $g_n^i : \mathcal{S} \times \Sigma \times \mathbb{N}_0^n \longrightarrow \mathbb{N}_0$ and $g_n^i(s, a, (m_1, \ldots, m_n)) \in D(a, s)$ for every $i \in \mathbb{N}$, $a \in \Sigma$, $s \in \mathcal{S}$, $(m_1, \ldots, m_n) \in \mathbb{N}_0^n$.

Definition 31. Let $\mathcal{M} = (\mathcal{F}, \Pi, \pi, F^+, F^-)$ be a $\mathrm{CGM}_{\mathrm{CF}}$ and let s be a state in $\mathcal{S}$. Let $\vec{c}$ be a coalition vector where $\mathrm{FA}(\vec{c}) \cup \mathrm{BA}(\vec{c}) = \{a_1, \ldots, a_n\}$ with $a_1 < \cdots < a_n$. A move vector σ *instantiates* the coalition vector $\vec{c}$ at state s, denoted by $\vec{c} \sqsubseteq \sigma$, if:
- $\sigma[a] = f^{\vec{c}[a]}(s, a)$ for every $a \in \mathrm{BA}(\vec{c})$,
- $\sigma[a'] = g_n^{|\vec{c}[a']|}(s, a', (\sigma[a_1], \ldots, \sigma[a_n]))$ for every $a' \in \mathrm{RA}(\vec{c})$.

The intuition underlying Definition 31 is the following. A coalition vector such as, for example, $(1, -2, *, 3, *, -2)$, indicates that agents 1 and 4 are committed to moves m_1 and m_4 that depend only on the state s they are currently in and are determined by the choice functions f^1 and f^3: $m_1 = f^1(s, 1)$ and $m_4 = f^3(s, 4)$, respectively. Agents 3 and 5 will perform arbitrary moves m_3 and m_5 of their choice in s. Finally, agents 2 and 6 will choose their moves m_2 and m_6 in reaction to the moves of all the other four agents and their moves are determined by the choice function $g_4^{|-2|} = g_4^2$: $m_2 = g_4^2(s, 3, (m_1, m_3, m_4, m_5))$ and $m_6 = g_4^2(s, 6, (m_1, m_3, m_4, m_5))$, respectively.

Definition 32. Let $\mathcal{M} = (\mathcal{F}, \Pi, \pi, F^+, F^-)$ be a $\mathrm{CGM}_{\mathrm{CF}}$ with $s \in \mathcal{S}$. The satisfaction relation $\models$ between $\mathcal{M}$, s and VCL formulae is inductively defined as follows.

- $\langle \mathcal{M}, s \rangle \models p$ iff $p \in \pi(s)$, for all $p \in \Pi$;
- $\langle \mathcal{M}, s \rangle \models \neg\varphi$ iff $\langle \mathcal{M}, s \rangle \not\models \varphi$;
- $\langle \mathcal{M}, s \rangle \models (\varphi \to \psi)$ iff $\langle \mathcal{M}, s \rangle \models \varphi$ implies $\langle \mathcal{M}, s \rangle \models \psi$;
- $\langle \mathcal{M}, s \rangle \models \bigwedge_{i=1}^{n} \varphi_i$ iff $\langle \mathcal{M}, s \rangle \models \varphi_i$ for all i, $1 \leq i \leq n$;
- $\langle \mathcal{M}, s \rangle \models \bigvee_{i=1}^{n} \varphi_i$ iff $\langle \mathcal{M}, s \rangle \models \varphi_i$ for some i, $1 \leq i \leq n$;
- $\langle \mathcal{M}, s \rangle \models \vec{c}\varphi$ iff for all $\sigma \in D(s)$, $\vec{c} \sqsubseteq \sigma$ implies $\langle \mathcal{M}, \delta(s, \sigma) \rangle \models \varphi$.

The notions of satisfiability of a VCL formula and a set of VCL formulae are defined as in Definitions 10 and 11 but with respect to $\mathrm{CGM}_{\mathrm{CF}}$'s instead of CGM's.

4.1 Normal Form Transformation

Definition 33. 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 $\mathsf{WFF}_{\mathsf{VCL}}$; and $\mathcal{N}$, the set of coalition formulae, is a finite set of coalition formulae, that is, formulae in which a coalition modality occurs.

In analogy to coalition problems in $\mathsf{DSNF}_{\mathsf{CL}}$, we introduce the notion of coalition problems in $\mathsf{DSNF}_{\mathsf{VCL}}$. As before, we use a clausal normal form:

$$\begin{array}{ll} \text{initial clauses} & \bigvee_{j=1}^{n} l_j \\ \text{global clauses} & \bigvee_{j=1}^{n} l_j \\ \text{coalition clauses} & \bigwedge_{i=1}^{m} l'_i \;\rightarrow\; \vec{c}\bigvee_{j=1}^{n} l_j \end{array}$$

where $m, n \geq 0$ and l'_i, l_j, for all $1 \leq i \leq m$, $1 \leq j \leq n$, are literals such that within every conjunction and every disjunction literals are pairwise different, and $\vec{c}$ is a coalition vector.

Definition 34. A *coalition problem in $\mathsf{DSNF}_{\mathsf{VCL}}$* 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 coalition clauses.

We define the rewriting system R_4 as $R_4 = \{\Rightarrow^1_{\wedge}, \Rightarrow^1_{\rightarrow,\mathcal{U}}, \Rightarrow^1_{\rightarrow,\mathcal{N}}, \Rightarrow^1_{\langle\Sigma_\varphi\rangle}, \Rightarrow^2_{\vee}, \Rightarrow^2_{[\text{-}]}, \Rightarrow^2_{\langle\text{-}\rangle}, \Rightarrow^4_{[\text{-}]}, \Rightarrow^4_{\langle\text{-}\rangle}\}$ together with simplification, where

$$\begin{array}{lll} (\mathcal{I},\mathcal{U},\mathcal{N} \cup \{t \rightarrow [\mathcal{A}]\psi\}) & \Rightarrow^4_{[\text{-}]} & (\mathcal{I},\mathcal{U},\mathcal{N} \cup \{t \rightarrow \vec{c}^{\,i}_{\mathcal{A}}\psi\}) \\ & & \text{where } \psi \text{ is a disjunction of literals, } i \text{ is a natural number} \\ & & \text{not occurring as an index of some coalition vector in } \mathcal{N}\text{, and} \\ & & \vec{c}^{\,i}_{\mathcal{A}} \text{ is a positive coalition vector with index } i \\ (\mathcal{I},\mathcal{U},\mathcal{N} \cup \{t \rightarrow \langle\mathcal{A}\rangle\psi\}) & \Rightarrow^4_{\langle\text{-}\rangle} & (\mathcal{I},\mathcal{U},\mathcal{N} \cup \{t \rightarrow \vec{c}^{\,-i}_{\mathcal{A}}\psi\}) \\ & & \text{where } \psi \text{ is a disjunction of literals, } \mathcal{A} \neq \Sigma_\varphi\text{, } i \text{ is a natural} \\ & & \text{number not occurring as an index of some coalition vector in} \\ & & \mathcal{N}\text{, and } \vec{c}^{\,-i}_{\mathcal{A}} \text{ is a negative coalition vector with index } i \end{array}$$

Note that the condition that i is a natural number not occurring as an index of some coalition vector in $\mathcal{N}$ implies that i only occurs in one particular clause and is uniquely associated with the CL formula $t \rightarrow [A]\psi$ or $t \rightarrow \langle A\rangle\psi$ to which the rewriting rules is applied.

Lemma 8. Let $\mathcal{C}$ be a coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$. Let $\mathcal{C}_0, \mathcal{C}_1, \ldots$ be a sequence of coalition problems such that $\mathcal{C}_0 = \mathcal{C}$ and for each i, $i > 0$, $\mathcal{C}_{i+1}$ is obtained from $\mathcal{C}_i$ by applying a rewriting rule in R_4 combined with zero or more applications of the simplification rules to a formula in $\mathcal{C}_i$. for each i, $i > 0$, $\mathcal{C}_{i+1}$ is obtained from $\mathcal{C}_i$ by applying a rewriting rule in R_4 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 rewriting rule can be applied to $\mathcal{C}_n$. Furthermore, $\mathcal{C}_n$ is a coalition problem in $\mathsf{DSNF}_{\mathsf{VCL}}$, the size of $\mathcal{C}_n$ is linear in the size of $\mathcal{C}$, and $\mathcal{C}_n$ is satisfiable if, and only if, $\mathcal{C}$ is satisfiable.

Proof. Let $\mathcal{C} = \mathcal{C}_0 = (\mathcal{I}_0, \mathcal{U}_0, \mathcal{N}_0)$. We construct a sequence $\mathcal{C}_0, \ldots$ such that for all i, $i \geq 0$, $\mathcal{C}_{i+1} = (\mathcal{I}_0, \mathcal{U}_0, \mathcal{N}_{i+1})$ is obtained from $\mathcal{C}_i = (\mathcal{I}_0, \mathcal{U}_0, \mathcal{N}_i)$ by applying $\Rightarrow^4_{[\text{-}]}$ or $\Rightarrow^4_{\langle\text{-}\rangle}$ combined with zero or more applications of the simplification rules to a formula in $\mathcal{C}_i$. If $n_{CL}(\mathcal{C})$ is the number of occurrences of a formula of the form $[\mathcal{A}]\psi$ or $\langle\mathcal{A}\rangle\psi$, for some coalition $\mathcal{A}$, in $\mathcal{C}$, then $n_{CL}(\mathcal{C}_i) = n_{CL}(\mathcal{C}_{i+1}) + 1$, for every i, $i \geq m$. Thus, the sequence $\mathcal{C}_0, \ldots$ terminates, in particular, it terminates after $n_{CL}(\mathcal{C}_0) = |\mathcal{N}_0|$ applications of the two rewriting rules and no rewriting rule can be applied to $\mathcal{C}_n$ where $n = m + |\mathcal{N}_0|$. Since each application of $\Rightarrow^4_{[\text{-}]}$ and $\Rightarrow^4_{\langle\text{-}\rangle}$ replaces a CL formula by a VCL formula of the same size, we have $|\mathcal{C}_0| = |\mathcal{C}_n|$ and therefore the size of $\mathcal{C}_n$ is linear in the size of φ. Also, $\mathcal{C}_n$ is in $\mathsf{DSNF}_{\mathsf{VCL}}$.

It remains to show that $\mathcal{C}_n$ is satisfiable iff φ is satisfiable which is equivalent to showing that $\mathcal{C}_0$ is satisfiable iff $\mathcal{C}_n$ is satisfiable.

First, assume that $\mathcal{C}_0$ is satisfiable and that $\mathcal{M} = (\mathcal{F}, \Pi, \pi)$ with $\mathcal{F} = (\Sigma, \mathcal{S}, s_0, d, \delta)$ is a model of $\mathcal{C}_0$. We construct a CGM$_{\text{CF}}$ $\mathcal{M}' = (\mathcal{F}, \Pi, \pi, F^+, F^-)$ for $\mathcal{C}_n$ as follows:

(a) To construct F^+ we proceed as follows. Let $t \to [A]\psi$ be a formula in $\mathcal{N}_0$ and let $t \to \vec{c}^{\,i}_{\mathcal{A}}\psi$ be the formula it is replaced with by $\Rightarrow^4_{[\text{-}]}$ where $\vec{c}^{\,i}_{\mathcal{A}}$ is a positive coalition vector with index i and i is a natural number not occurring in any other formula. Recall that $\vec{c}^{\,i}_{\mathcal{A}}$ is a $|\Sigma|$-tuple such that $\vec{c}^{\,i}_{\mathcal{A}}[a] = i$ for every $a \in \mathcal{A}$ and $\vec{c}^{\,i}_{\mathcal{A}}[a'] = *$ for every $a' \notin \mathcal{A}$. We need to define a function $f^i : \mathcal{S} \times \Sigma \longrightarrow \mathbb{N}_0$.

Let $s \in \mathcal{S}$ be a state such that $\langle \mathcal{M}, s\rangle \models [A]\psi$. Then there exists an $\mathcal{A}$-move $\sigma_{\mathcal{A}} \in D(\mathcal{A}, s)$ such that for all $\sigma \in D(s)$ $\sigma_{\mathcal{A}} \sqsubseteq \sigma$ implies $\langle \mathcal{M}, \delta(s,\sigma)\rangle \models \psi$. Note that $\sigma_{\mathcal{A}}$ is a $|\Sigma|$-tuple such that $\sigma_{\mathcal{A}}[a] \in D(a, s)$ for every $a \in \mathcal{A}$ and $\sigma_{\mathcal{A}}[a'] = *$ for every $a' \notin \mathcal{A}$. We define $f^i(s,a) = \sigma_{\mathcal{A}}[a]$ for every $a \in \mathcal{A}$ and $f^i(s,a') = 0$ for every $a' \notin \mathcal{A}$.

For any state $s \in \mathcal{S}$ such that $\langle \mathcal{M}, s\rangle \not\models [A]\psi$, we define $f^i(s,a) = 0$ for every $a \in \Sigma$.

Finally, for any natural number i for which there is no positive coalition vector with index i in $\mathcal{C}_n$ we define $f^i(s,a) = 0$ for every $s \in \mathcal{S}$ and every $a \in \Sigma$.

(b) To construct F^- we proceed as follows. Let $t \to \langle A\rangle\psi$ be a formula in $\mathcal{N}_0$ and let $t \to \vec{c}^{\,-i}_{\mathcal{A}}\psi$ be the formula it is replaced with by $\Rightarrow^4_{\langle\text{-}\rangle}$ where $\vec{c}^{\,-i}_{\mathcal{A}}$ is a negative coalition vector with index i and i is a natural number not occurring in any other formula. Let $\mathcal{A} = \{a_1, \ldots, a_n\}$ with $a_1 < \cdots < a_n$. Recall that $\vec{c}^{\,-i}_{\mathcal{A}}$ is a $|\Sigma|$-tuple such that $\vec{c}^{\,-i}_{\mathcal{A}}(a') = -i$ for every $a' \notin \mathcal{A}$ and $\vec{c}^{\,i}_{\mathcal{A}}(a) = *$ for every $a \in \mathcal{A}$. We need to define a function $g^i_n : \mathcal{S} \times \Sigma \times \mathbb{N}_0^n \longrightarrow \mathbb{N}_0$.

Let $s \in \mathcal{S}$ be a state such that $\langle \mathcal{M}, s\rangle \models \langle \mathcal{A}\rangle\psi$. Then for all $\mathcal{A}$-moves $\sigma_{\mathcal{A}} \in D(\mathcal{A}, s)$ exists $\sigma \in D(s)$ such that $\sigma_{\mathcal{A}} \sqsubseteq \sigma$ and $\langle \mathcal{M}, \delta(s,\sigma)\rangle \models \psi$. Again, $\sigma_{\mathcal{A}}$ is a $|\Sigma|$-tuple such that $\sigma_{\mathcal{A}}[a] = m_a \in D(a,s)$ for every $a \in \mathcal{A}$ and $\sigma_{\mathcal{A}}[a'] = *$ for every $a' \notin \mathcal{A}$. Since $\sigma_{\mathcal{A}} \sqsubseteq \sigma$ we have $\sigma_{\mathcal{A}}[a] = \sigma[a]$ for every $a \in \mathcal{A}$. Let $(m_{a_1}, \ldots, m_{a_n})$ be the sequence of all of agents in $\mathcal{A}$.

We define $g^i_n(s, a', (m_{a_1}, \ldots, m_{a_n})) = \sigma[a']$ for every $a' \in \Sigma$. In fact, it is sufficient that we define $g^i_n(s, a', (m_{a_1}, \ldots, m_{a_n})) = \sigma[a']$ for every $a' \notin \mathcal{A}$ while for agents $a \in \mathcal{A}$ we could choose an arbitrary move in $D(a,s)$. So, we might as well choose $\sigma[a]$. We also define $g^i_n(s, a', (m'_1, \ldots, m'_n)) = 0$ for every sequence $(m_1, \ldots, m_n) \in \mathbb{N}_0^n$ such that $m_a \notin D(a,s)$ for some $a \in \mathcal{A}$.

For any natural number n' with $n' \neq n$, we define $g^i_n(s, a, (m_1, \ldots, m_{n'})) = 0$ for every $s \in \mathcal{S}$, every $a \in \Sigma$ and every $(m_1, \ldots, m_{n'}) \in \mathbb{N}_0^{n'}$.

Finally, for any natural number i for which there is no negative coalition vector with index i in $\mathcal{C}_n$ and for every $n \in \mathbb{N}$ we define $g^i_n(s, a, (m_1, \ldots, m_n)) = 0$ for every $s \in \mathcal{S}$, every $a \in \Sigma$ and every $(m_1, \ldots, m_n) \in \mathbb{N}_0^n$.

We now need to show that $\mathcal{M}', s_0 \models \mathcal{C}_n$.

- As the frame underlying both $\mathcal{M}$ and $\mathcal{M}'$ and the valuation function π are the same, for every propositional formula θ and every $s \in \mathcal{S}$ we have $\langle \mathcal{M}, s\rangle \models \theta$ iff $\langle \mathcal{M}', s\rangle \models \theta$.

  As the sets of initial and universal clauses in $\mathcal{C}_n = (\mathcal{I}_0, \mathcal{U}_0, \mathcal{N}_0)$ are the same as in $\mathcal{C}_0$ and $\langle \mathcal{M}, s_0\rangle \models I_0$ and for every $s \in \mathcal{S}$, $\langle \mathcal{M}, s\rangle \models \mathcal{U}_0$, we therefore have $\langle \mathcal{M}', s_0\rangle \models \mathcal{I}_0$ and for every $s \in \mathcal{S}$, $\langle \mathcal{M}', s\rangle \models \mathcal{U}_0$.

- It remains to show that for every $\Gamma \in \mathcal{N}_n$, $\langle \mathcal{M}', s\rangle \models \Gamma$.
  - Let $t \to [A]\psi$ be a formula in $\mathcal{N}_0$ and let $t \to \vec{c}^{\,i}_{\mathcal{A}}\psi$ be the formula it is replaced with by $\Rightarrow^4_{[\text{-}]}$ where $\vec{c}^{\,i}_{\mathcal{A}}$ is a positive coalition vector with index i and i is a natural number not occurring in any other formula, $\vec{c}^{\,i}_{\mathcal{A}}$ is a $|\Sigma|$-tuple such that $\vec{c}^{\,i}_{\mathcal{A}}[a] = i$ for every $a \in \mathcal{A}$ and $\vec{c}^{\,i}_{\mathcal{A}}[a'] = *$ for every $a' \notin \mathcal{A}$. Note that $\text{BA}(\vec{c}^{\,i}_{\mathcal{A}}) = \mathcal{A}$.
    If $\langle \mathcal{M}', s\rangle \not\models t$, then $\langle \mathcal{M}', s\rangle \models t \to \vec{c}^{\,i}_{\mathcal{A}}\psi$ holds.
    If $\langle \mathcal{M}', s\rangle \models t$, then since $\langle \mathcal{M}, s\rangle \models t$ iff $\langle \mathcal{M}', s\rangle \models t$, we have $\langle \mathcal{M}, s\rangle \models t$ and $\langle \mathcal{M}, s\rangle \models [A]\psi$. So, there exists an $\mathcal{A}$-move $\sigma_{\mathcal{A}}$ such that for all $\sigma \in D(s)$, $\sigma_{\mathcal{A}} \sqsubseteq \sigma$ implies $\langle \mathcal{M}, \delta(s,\sigma)\rangle \models \psi$.
    By construction of f^i in case (a) above, for every $a \in \mathcal{A}$, $f^i(s,a) = \sigma_{\mathcal{A}}[a]$. So, for all $\sigma \in D(s)$, $\sigma[a] = f^i(s,a)$ for every $a \in \text{BA}(\vec{c})$ implies $\sigma[a] = \sigma_{\mathcal{A}}[a]$ for every $a \in \mathcal{A}$ which in turn implies $\sigma_{\mathcal{A}} \sqsubseteq \sigma$.

Also, as the frame underlying both $\mathcal{M}$ and $\mathcal{M}'$ and the valuation function π are the same, for every propositional formula θ, for every state s', for every $\sigma' \in D(s')$, we $\langle \mathcal{M}, \delta(s', \sigma') \rangle \models \theta$ iff $\langle \mathcal{M}', \delta(s', \sigma') \rangle \models \theta$.

Therefore, for every $\sigma \in D(s)$, $\vec{c}^{\,i}_{\mathcal{A}} \sqsubseteq \sigma$ is equivalent to $\sigma[a] = f^i(s, a)$ for every $a \in \text{BA}(\vec{c})$ which implies $\sigma_{\mathcal{A}} \sqsubseteq \sigma$, and $\sigma_{\mathcal{A}} \sqsubseteq \sigma$, implies $\langle \mathcal{M}, \delta(s, \sigma) \rangle \models \psi$ implies $\langle \mathcal{M}', \delta(s, \sigma) \rangle \models \psi$. Thus, by definition of the semantics of VCL, $\langle \mathcal{M}', s \rangle \models \vec{c}^{\,i}_{\mathcal{A}} \psi$ and $\langle \mathcal{M}', s \rangle \models t \to \vec{c}^{\,i}_{\mathcal{A}} \psi$.

– Let $t \to \langle A \rangle \psi$ be a formula in $\mathcal{N}_0$ and let $t \to \vec{c}^{\,-i}_{\mathcal{A}} \psi$ be the formula it is replaced with by $\Rightarrow^4_{[\text{-}]}$ where $\vec{c}^{\,-i}_{\mathcal{A}}$ is a negative coalition vector with index i and i is a natural number not occurring in any other formula, $\vec{c}^{\,-i}_{\mathcal{A}}$ is a $|\Sigma|$-tuple such that $\vec{c}^{\,-i}_{\mathcal{A}}(a') = -i$ for every $a' \notin \mathcal{A}$ and $\vec{c}^{\,-i}_{\mathcal{A}}(a) = *$ for every $a \in \mathcal{A}$. Note that $\text{RA}(\vec{c}^{\,-i}_{\mathcal{A}}) = \Sigma \setminus \mathcal{A}$. Let $n = |\mathcal{A}|$ and $\mathcal{A} = \{a_1, \ldots, a_n\}$ with $a_1 < \cdots < a_n$.

If $\langle \mathcal{M}', s \rangle \not\models t$, then $\langle \mathcal{M}', s \rangle \models t \to \vec{c}^{\,-i}_{\mathcal{A}} \psi$.

If $\langle \mathcal{M}', s \rangle \models t$, then since $\langle \mathcal{M}, s \rangle \models t$ iff $\langle \mathcal{M}', s \rangle \models t$, we have $\langle \mathcal{M}, s \rangle \models t$ and $\langle \mathcal{M}, s \rangle \models \langle A \rangle \psi$. So, for all $\mathcal{A}$-moves $\sigma_{\mathcal{A}} \in D(\mathcal{A}, s)$ exists $\sigma \in D(s)$ such that $\sigma_{\mathcal{A}} \sqsubseteq \sigma$ and $\langle \mathcal{M}, \delta(s, \sigma) \rangle \models \psi$.

Let $\sigma \in D(s)$ be a move vector such that $\vec{c}^{\,-i}_{\mathcal{A}} \sqsubseteq \sigma$, that is, $\sigma[a'] = g^i_n(s, a', (\sigma[a_1], \ldots, \sigma[a_n]))$ for every $a' \in \text{RA}(\vec{c}^{\,-i}_{\mathcal{A}})$. Define $\sigma_{\mathcal{A}}$ as $\sigma_{\mathcal{A}}[a] = \sigma[a]$ for $a \in \mathcal{A}$ and $\sigma_{\mathcal{A}}[a'] = *$ for $a \in \text{RA}(\vec{c}^{\,-i}_{\mathcal{A}})$. Then $\sigma_{\mathcal{A}}$ is an $\mathcal{A}$-move in $D(\mathcal{A}, s)$ and $\sigma_{\mathcal{A}} \sqsubseteq \sigma$. By construction of g^i_n in case (b) above, $\langle \mathcal{M}, \delta(s, \sigma) \rangle \models \psi$ which in turn implies $\langle \mathcal{M}', \delta(s, \sigma) \rangle \models \psi$ as ψ.

Therefore, for every $\sigma \in D(s)$, $\sigma \sqsubseteq \vec{c}^{\,-i}_{\mathcal{A}}$ is equivalent to $\sigma[a'] = g^i_n(s, a', (\sigma[a_1], \ldots, \sigma[a_n]))$ for every $a' \in \text{RA}(\vec{c}^{\,-i}_{\mathcal{A}})$ which implies $\sigma_{\mathcal{A}} \sqsubseteq \sigma$, and $\sigma_{\mathcal{A}} \sqsubseteq \sigma$ implies $\langle \mathcal{M}, \delta(s, \sigma) \rangle \models \psi$ implies $\langle \mathcal{M}', \delta(s, \sigma) \rangle \models \psi$. Thus, by definition of the semantics of VCL, $\langle \mathcal{M}', s \rangle \models \vec{c}^{\,-i}_{\mathcal{A}} \psi$ and $\langle \mathcal{M}', s \rangle \models t \to \vec{c}^{\,-i}_{\mathcal{A}} \psi$. for every i, $1 \le i \le n$,

Second, assume that $\mathcal{C}_n = (\mathcal{I}_0, \mathcal{U}_0, \mathcal{U}_n)$ is satisfiable and that $\mathcal{M}' = (\mathcal{F}, \Pi, \pi, F^+, F^-)$ with $\mathcal{F} = (\Sigma, \mathcal{S}, s_0, d, \delta)$ is a Concurrent Game Model with Choice Functions such that $\langle \mathcal{M}', s_0 \rangle \models \mathcal{C}_n$. Let $\mathcal{M} = (\mathcal{F}, \Pi, \pi)$. We show that $\langle \mathcal{M}, s_0 \rangle \models \mathcal{C}_0$.

- As the frame underlying both $\mathcal{M}$ and $\mathcal{M}'$ and the valuation function π are the same, for every propositional formula θ and every $s \in \mathcal{S}$ we have $\langle \mathcal{M}, s \rangle \models \theta$ iff $\langle \mathcal{M}', s \rangle \models \theta$.

  We therefore have $\langle \mathcal{M}, s_0 \rangle \models \mathcal{I}_0$ and for every $s \in \mathcal{S}$, $\langle \mathcal{M}, s \rangle \models \mathcal{U}_0$.

- It remains to show that for every $\Gamma \in \mathcal{N}_0$ and every $s \in \mathcal{S}$, $\langle \mathcal{M}, s \rangle \models \Gamma$.

  – Let $t \to [A]\psi$ be a formula in $\mathcal{N}_0$ and let $t \to \vec{c}^{\,i}_{\mathcal{A}} \psi$ be the formula it is replaced with by $\Rightarrow^4_{[\text{-}]}$ where $\vec{c}^{\,i}_{\mathcal{A}}$ is a positive coalition vector with index i and i is a natural number not occurring in any other formula, $\vec{c}^{\,i}_{\mathcal{A}}$ is a $|\Sigma|$-tuple such that $\vec{c}^{\,i}_{\mathcal{A}}[a] = i$ for every $a \in \mathcal{A}$ and $\vec{c}^{\,i}_{\mathcal{A}}[a'] = *$ for every $a' \notin \mathcal{A}$. Let $\mathcal{A} = \{a_1, \ldots, a_n\}$ with $a_1 < \cdots < a_n$. Note that $\text{BA}(\vec{c}^{\,i}_{\mathcal{A}}) = \mathcal{A}$.

  If $\langle \mathcal{M}, s \rangle \not\models t$, then $\langle \mathcal{M}, s \rangle \models t \to [A]\psi$ holds.

  If $\langle \mathcal{M}, s \rangle \models t$, then since $\langle \mathcal{M}, s \rangle \models t$ iff $\langle \mathcal{M}', s \rangle \models t$, we have $\langle \mathcal{M}', s \rangle \models t$ and $\langle \mathcal{M}', s \rangle \models \vec{c}^{\,i}_{\mathcal{A}} \psi$. So, for every $\sigma \in D(s)$, if $\vec{c}^{\,i}_{\mathcal{A}} \sqsubseteq \sigma$ then $\langle \mathcal{M}', \delta(s, \sigma) \rangle \models \psi$. Recall that $\vec{c}^{\,i}_{\mathcal{A}} \sqsubseteq \sigma$ if $\sigma[a] = f^i(s, a)$ for every $a \in \text{BA}(\vec{c}^{\,i}_{\mathcal{A}})$. Let $\sigma_{\mathcal{A}}$ be the $\mathcal{A}$-move defined by $\sigma_{\mathcal{A}}[a] = f^i(s, a)$ for every $a \in \text{BA}(\vec{c}^{\,i}_{\mathcal{A}}) = \mathcal{A}$ and $\sigma_{\mathcal{A}}[a'] = *$ for every $a' \notin \mathcal{A}$. Then for every $\sigma \in D(s)$, $\vec{c}^{\,i}_{\mathcal{A}} \sqsubseteq \sigma$ implies $\sigma_{\mathcal{A}} \sqsubseteq \sigma$ and therefore, $\sigma_{\mathcal{A}} \sqsubseteq \sigma$ implies $\langle \mathcal{M}', \delta(s, \sigma) \rangle \models \psi$. Thus, there exists an $\mathcal{A}$-move $\sigma_{\mathcal{A}}$ such that for all $\sigma \in D(s)$ $\sigma_{\mathcal{A}} \sqsubseteq \sigma$ implies $\langle \mathcal{M}, \delta(s, \sigma) \rangle \models \varphi$, and consequently, $\langle \mathcal{M}, s \rangle \models [A]\psi$ and $\langle \mathcal{M}, s \rangle \models t \to [A]\psi$.

  – Let $t \to \langle A \rangle \psi$ be a formula in $\mathcal{N}_0$ and let $t \to \vec{c}^{\,-i}_{\mathcal{A}} \psi$ be the formula it is replaced with by $\Rightarrow^4_{\langle \text{-} \rangle}$ where $\vec{c}^{\,-i}_{\mathcal{A}}$ is a negative coalition vector with index i and i is a natural number not occurring in any other formula, $\vec{c}^{\,-i}_{\mathcal{A}}$ is a $|\Sigma|$-tuple such that $\vec{c}^{\,-i}_{\mathcal{A}}(a') = -i$ for every $a' \notin \mathcal{A}$ and $\vec{c}^{\,-i}_{\mathcal{A}}(a) = *$ for every $a \in \mathcal{A}$. Let $n = |\mathcal{A}|$ and $\mathcal{A} = \{a_1, \ldots, a_n\}$ with $a_1 < \cdots < a_n$. Note that $\text{RA}(\vec{c}^{\,-i}_{\mathcal{A}}) = \Sigma \setminus \mathcal{A}$.

  If $\langle \mathcal{M}, s \rangle \not\models t$, then $\langle \mathcal{M}, s \rangle \models t \to \langle A \rangle \psi$ holds.

  If $\langle \mathcal{M}, s \rangle \models t$, then since $\langle \mathcal{M}, s \rangle \models t$ iff $\langle \mathcal{M}', s \rangle \models t$, we have $\langle \mathcal{M}', s \rangle \models t$ and $\langle \mathcal{M}', s \rangle \models \vec{c}^{\,-i}_{\mathcal{A}} \psi$. Then for every $\sigma \in D(s)$, $\vec{c}^{\,-i}_{\mathcal{A}} \sqsubseteq \sigma$ implies $\langle \mathcal{M}', \delta(s, \sigma) \rangle \models \psi$. Recall that $\vec{c}^{\,-i}_{\mathcal{A}} \sqsubseteq \sigma$ if $\sigma[a'] = g^i_n(s, a', (\sigma[a_1], \ldots, \sigma[a_n]))$ for every $a' \in \text{RA}(\vec{c}^{\,-i}_{\mathcal{A}})$.

  Let $\sigma_{\mathcal{A}}$ be an arbitrary $\mathcal{A}$-move. Let σ be the move vector defined by $\sigma[a] = \sigma_{\mathcal{A}}[a]$ for every $a \in \mathcal{A}$ and $\sigma[a'] = g^i_n(s, a', (\sigma[a_1], \ldots, \sigma[a_n]))$ for every $a' \in \Sigma \setminus \mathcal{A} = \text{RA}(\vec{c}^{\,-i}_{\mathcal{A}})$. Then $\sigma_{\mathcal{A}} \sqsubseteq \sigma$ and

$\vec{c}^{\,-i}_{\mathcal{A}} \sqsubseteq \sigma$ and therefore $\langle \mathcal{M}', \delta(s,\sigma)\rangle \models \psi$. Thus, for every $\mathcal{A}$-move $\sigma_{\mathcal{A}}$ there exists $\sigma \in D(s)$ such that $\sigma_{\mathcal{A}} \sqsubseteq \sigma$ and $\langle \mathcal{M}, \delta(s,\sigma)\rangle$, and consequently, $\langle \mathcal{M}, s\rangle \models \langle A\rangle\psi$ and $\langle \mathcal{M}, s\rangle \models t \rightarrow \langle A\rangle\psi$. □

Theorem 6. *Let $\varphi \in \mathsf{WFF}_{\mathsf{CL}}$. Let $\mathcal{C}_0, \mathcal{C}_1, \ldots$ be a sequence of coalition problems such that $\mathcal{C}_0 = (\{t_\varphi\}, \{t_\varphi \rightarrow \tau_0(\varphi)\}, \emptyset)$ and $\mathcal{C}_{i+1}$ is obtained from $\mathcal{C}_i$ by applying a rewriting rule in R_4 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 rewriting rule can be applied to $\mathcal{C}_n$. Furthermore, $\mathcal{C}_n$ is a coalition problem in $\mathsf{DSNF}_{\mathsf{VCL}}$, the size of $\mathcal{C}_n$ is linear in the size of φ, and $\mathcal{C}_n$ is satisfiable if, and only if, φ is satisfiable.*

Proof. The rewriting system R_4 extends the rewriting system R_2 with the additional rules $\Rightarrow^4_{[\text{-}]}$ and $\Rightarrow^4_{\langle\text{-}\rangle}$. Without loss of generality we can assume that R_2 is first exhaustively applied to $\mathcal{C}_0 = (\{t_0\}, \{t_0 \rightarrow \tau_0(\varphi)\}, \{\})$. By Theorem 1 we obtain a terminating sequence $\mathcal{C}_0, \ldots, \mathcal{C}_m = (\mathcal{I}_m, \mathcal{U}_m, \mathcal{N}_m)$ such that no rewriting rule of R_2 can be applied to $\mathcal{C}_m$, $\mathcal{C}_m$ is in $\mathsf{DSNF}_{\mathsf{CL}}$, and (i) the size of $\mathcal{C}_m$ is linear in the size of φ and (ii) $\mathcal{C}_m$ is satisfiable if, and only if φ is satisfiable.

Then by Lemma 8, there is a terminating sequence $\mathcal{C}_m, \ldots, \mathcal{C}_n$ such that for each i, $m \leq i < n$, $\mathcal{C}_{i+1}$ is obtained from $\mathcal{C}_i$ by applying a rewriting rule in R_4 combined with zero or more applications of the simplification rules to a formula in $\mathcal{C}_i$, and (iii) $\mathcal{C}_n$ is a coalition problem in $\mathsf{DSNF}_{\mathsf{VCL}}$, (iv) the size of $\mathcal{C}_n$ is linear in the size of $\mathcal{C}_m$, and (v) $\mathcal{C}_n$ is satisfiable if, and only if, $\mathcal{C}_m$ is satisfiable.

Properties (i) and (iv) together imply that $\mathcal{C}_n$ is linear in the size of φ, and properties (ii) and (v) together imply that $\mathcal{C}_n$ is satisfiable if, and only if, φ is satisfiable. □

We return to our previous example, a formalisation of the pigeon hole problem with three pigeons and two holes in Coalition Logic, given by the formula φ^3_2 (page 9). The transformation of φ^3_2 into a coalition problem $\mathcal{C}'^3_2$ in $\mathsf{DSNF}_{\mathsf{VCL}}$ using rewriting system R_4 results in the ten clauses in Figure 3.

4.2 Resolution Calculus $\mathsf{RES}^{\succ}_{\mathsf{CL}}$

Definition 35. Let $\vec{c}_1$ and $\vec{c}_2$ be two coalition vectors. The coalition vector $\vec{c}_2$ is an *instance* of $\vec{c}_1$ and $\vec{c}_1$ is *more general than* $\vec{c}_2$, written $\vec{c}_1 \sqsubseteq \vec{c}_2$, if $\vec{c}_1[a] = \vec{c}_2[a]$ for every a, $1 \leq a \leq |\Sigma|$, with $\vec{c}_1[a] \neq *$. We say that a coalition vector $\vec{c}_3$ is a *common instance* of $\vec{c}_1$ and $\vec{c}_2$ if $\vec{c}_3$ is an instance of both $\vec{c}_1$ and $\vec{c}_2$. A coalition vector $\vec{c}_3$ is a *most general common instance* or *merge* of $\vec{c}_1$ and $\vec{c}_2$ if $\vec{c}_3$ is a common instance of $\vec{c}_1$ and $\vec{c}_2$, and for any common instance $\vec{c}_4$ of $\vec{c}_1$ and $\vec{c}_2$ we have $\vec{c}_3 \sqsubseteq \vec{c}_4$. If there exists a merge for two coalition vectors $\vec{c}_2$ and $\vec{c}_2$ then we say that $\vec{c}_1$ and $\vec{c}_2$ are *mergeable*.

For example, $(1,3,2,*)$, $(1,*,2,1)$, $(1,3,2,1)$ are all instances of $(1,*,2,*)$, while $(1,3,2,*)$ and $(1,3,2,1)$ are instances of $(*,3,2,*)$. The coalition vectors $(1,*,2,*)$ and $(*,3,2,*)$ are mergeable and $(1,3,2,*)$ is the merge of the two vectors. While $(1,3,2,1)$ is an instance of both $(1,*,2,*)$ and $(*,3,2,*)$, $(1,3,2,*)$ is more general than $(1,3,2,1)$. In contrast, $(1,*)$ and $(2,1)$ are not mergeable as they do not have a common instance. Also, $(1,-1,*)$ and $(1,*,-2)$ are not mergeable as the vector $(1,-1,-2)$ is not a coalition vector, as it contains two different negative numbers.

Lemma 9. Let $\vec{c}_1$ and $\vec{c}_2$ be two coalition vectors. Let $\vec{c}_3$ be the vector over $\mathbb{Z} \cup \{\bot\}$ defined by

$$\vec{c}_3[i] = \begin{cases} \vec{c}_1[i], & \text{if (1) } \vec{c}_2[i] = * \\ \vec{c}_2[i], & \text{if (2) } \vec{c}_1[i] = * \\ \vec{c}_1[1], & \text{if (3) } \vec{c}_1[i] = \vec{c}_2[i] \\ \bot, & \text{if (4) } \vec{c}_1[i] \neq * \text{ and } \vec{c}_2[i] \neq * \text{ and } \vec{c}_1[i] \neq \vec{c}_2[i] \end{cases}$$

1.	t_0	$[\mathcal{I}]$	6.	$t_0 \rightarrow (*,*,*)\neg x^1_2 \vee \neg x^3_2$	$[\mathcal{N}]$
2.	$t_0 \rightarrow (*,*,*)\neg x^1_1 \vee \neg x^2_1$	$[\mathcal{N}]$	7.	$t_0 \rightarrow (*,*,*)\neg x^2_2 \vee \neg x^3_2$	$[\mathcal{N}]$
3.	$t_0 \rightarrow (*,*,*)\neg x^1_1 \vee \neg x^3_1$	$[\mathcal{N}]$	8.	$t_0 \rightarrow (1,*,*)x^1_1 \vee x^1_2$	$[\mathcal{N}]$
4.	$t_0 \rightarrow (*,*,*)\neg x^2_1 \vee \neg x^3_1$	$[\mathcal{N}]$	9.	$t_0 \rightarrow (*,2,*)x^2_1 \vee x^2_2$	$[\mathcal{N}]$
5.	$t_0 \rightarrow (*,*,*)\neg x^1_2 \vee \neg x^2_2$	$[\mathcal{N}]$	10.	$t_0 \rightarrow (*,*,3)x^3_1 \vee x^3_2$	$[\mathcal{N}]$

Figure 3: Pigeon hole problem in $\mathsf{DSNF}_{\mathsf{VCL}}$

for every i, $1 \leq i \leq |\Sigma|$. If $\vec{c}_3$ does not contain $\bot$ and does not contain two distinct negative numbers, then $\vec{c}_1$ and $\vec{c}_2$ are mergeable and $\vec{c}_3$ is a merge of $\vec{c}_1$ and $\vec{c}_2$.

Proof. If $\vec{c}_3$ contains $\bot$, then there is some i with both $\vec{c}_1[i], \vec{c}_2[i] \in \mathbb{Z}$ and $\vec{c}_1[i] \neq \vec{c}_2[i]$. This implies that $\vec{c}_1$ and $\vec{c}_2$ do not have a common instance. In the remainder of the proof we assume that $\vec{c}_3$ does not contain $\bot$.

By definition of coalition vectors, a coalition vector does not contain two different negative numbers. If only one of $\vec{c}_1$ and $\vec{c}_2$ contains a negative number, say, $-k$, then $\vec{c}_3$ also contains $-k$ and no other negative number. If $\vec{c}_1$ contains the negative number $-k_{\vec{c}_1}$ and $\vec{c}_2$ contains the negative number $-k_{\vec{c}_2}$, then $\vec{c}_3$ contains both $-k_{\vec{c}_1}$ and $-k_{\vec{c}_2}$. If $-k_{\vec{c}_1} \neq -k_{\vec{c}_2}$, then $\vec{c}_3$ contains two different negative numbers and is not a coalition vector. This again implies that $\vec{c}_1$ and $\vec{c}_2$ do not have a common instance. If $-k_{\vec{c}_1} = -k_{\vec{c}_2}$, then $\vec{c}_3$ is a coalition vector.

Thus, $\vec{c}_3$ is a coalition vector. Next, we need to show that it a common instance of $\vec{c}_1$ and of $\vec{c}_2$. By definition, $\vec{c}_1 \sqsubseteq \vec{c}_3$ if $\vec{c}_3[i] = \vec{c}_1[i]$ for every i, $1 \leq i \leq |\Sigma|$, with $\vec{c}_1[i] \neq *$. If $\vec{c}_1[i] \neq *$, then $\vec{c}_3[i] = \vec{c}_1[i]$ by case (1) or case (3) in the definition of $\vec{c}_3[i]$, as neither case (2) nor case (4) can apply. In analogy, $\vec{c}_2 \sqsubseteq \vec{c}_3$ if $\vec{c}_3[i] = \vec{c}_2[i]$ for every i, $1 \leq i \leq |\Sigma|$, with $\vec{c}_2[i] \neq *$. If $\vec{c}_2[i] \neq *$, then $\vec{c}_3[i] = \vec{c}_2[i]$ by case (2) or $\vec{c}_3[i] = \vec{c}_1[i] = \vec{c}_2[i]$ by case (3) in the definition of $\vec{c}_3[i]$.

It remains to show that for any common instance $\vec{c}_4$ of $\vec{c}_1$ and $\vec{c}_2$ we have $\vec{c}_3 \sqsubseteq \vec{c}_4$, that is, $\vec{c}_3[i] = \vec{c}_4[i]$ for every i, $1 \leq i \leq |\Sigma|$, with $\vec{c}_3[i] \neq *$. If $\vec{c}_3[i] \neq *$, then $\vec{c}_1[i] \in \mathbb{Z}$ and $\vec{c}_3[i] = \vec{c}_1[i]$, or $\vec{c}_2[i] \in \mathbb{Z}$ and $\vec{c}_3[i] = \vec{c}_2[i]$. To be an instance of both $\vec{c}_1$ and $\vec{c}_2$, $\vec{c}_4[i] = \vec{c}_1[i]$ if $\vec{c}_1[i] \in \mathbb{Z}$, and therefore $\vec{c}_4[i] = \vec{c}_1[i] = \vec{c}_3[i]$, or if $\vec{c}_2[i] \in \mathbb{Z}$, $\vec{c}_4[i] = \vec{c}_2[i]$ and therefore $\vec{c}_4[i] = \vec{c}_2[i] = \vec{c}_3[i]$.

Thus, $\vec{c}_3$ is a merge of $\vec{c}_1$ and $\vec{c}_2$. □

Lemma 10. Let $\vec{c}_1$ and $\vec{c}_2$ be two mergeable coalition vector and let both $\vec{c}_3$ and $\vec{c}_4$ be merges of $\vec{c}_1$ and $\vec{c}_2$. Then $\vec{c}_3 = \vec{c}_4$.

Proof. Since both $\vec{c}_3$ and $\vec{c}_4$ are merges of $\vec{c}_1$ and $\vec{c}_2$, they are both most general common instances of $\vec{c}_1$ and $\vec{c}_2$. In particular, $\vec{c}_3 \sqsubseteq \vec{c}_4$, that is, (i) $\vec{c}_4[a] = \vec{c}_3[a]$ for every a, $1 \leq a \leq |\Sigma|$, with $\vec{c}_3[a] \neq *$, and $\vec{c}_4 \sqsubseteq \vec{c}_3$, that is, (ii) $\vec{c}_3[a] = \vec{c}_4[a]$ for every a, $1 \leq a \leq |\Sigma|$, with $\vec{c}_4[a] \neq *$. Because of (i) and (ii), for every a, $1 \leq a \leq |\Sigma|$, if $\vec{c}_3[a] \neq *$ or $\vec{c}_4[a] \neq *$ then $\vec{c}_3[a] = \vec{c}_4[a]$. On the other hand, for every a, $1 \leq a \leq |\Sigma|$, if $\vec{c}_3[a] = *$ and $\vec{c}_4[a] = *$ then trivially $\vec{c}_3[a] = \vec{c}_4[a]$. Thus, $\vec{c}_3 = \vec{c}_4$. □

Lemma 9 gives us a way to compute a merge of two coalition vectors and Lemma 10 shows that there the merge of two coalition vectors is unique. We denote the merge of $\vec{c}_1$ and $\vec{c}_2$ by $\vec{c}_1 \downarrow \vec{c}_2$ and write $\vec{c}_1 \downarrow \vec{c}_2 = \textbf{undef}$ if $\vec{c}_1$ and $\vec{c}_2$ are not mergeable.

Lemma 11. Let $\vec{c}_1$ and $\vec{c}_2$ be two mergeable coalition vectors and let $\vec{c}_3 = \vec{c}_1 \downarrow \vec{c}_2$ be their merge. Then $\mathrm{FA}(\vec{c}_3) = \mathrm{FA}(\vec{c}_1) \cap \mathrm{FA}(\vec{c}_2)$, $\mathrm{BA}(\vec{c}_3) = \mathrm{BA}(\vec{c}_1) \cup \mathrm{BA}(\vec{c}_2)$ and $\mathrm{RA}(\vec{c}_3) = \mathrm{RA}(\vec{c}_1) \cup \mathrm{RA}(\vec{c}_2)$.

Proof. By Definition 25, $\mathrm{FA}(\vec{c}_i) = \{a \mid 1 \leq a \leq |\Sigma| \wedge \vec{c}_i[a] = *\}$, for every i, $1 \leq i \leq 3$. It is straightforward to see from the definition of a merge in Lemma 9 that $\vec{c}_1 \downarrow \vec{c}_2[a] = \vec{c}_3[a] = *$ iff $\vec{c}_1[a] = *$ and $\vec{c}_2[a] = *$, for any a, $1 \leq a \leq |\Sigma|$. Thus, $\mathrm{FA}(\vec{c}_1 \downarrow \vec{c}_2) = \mathrm{FA}(\vec{c}_1) \cap \mathrm{FA}(\vec{c}_2)$.

By Definition 25, $\mathrm{BA}(\vec{c}_i) = \{a \mid 1 \leq a \leq k \wedge \vec{c}_i[a] > 0\}$, for every i, $1 \leq i \leq 3$. Again, we can see from the definition of a merge in Lemma 9 that $\vec{c}_1 \downarrow \vec{c}_2[a] = \vec{c}_3[a] > 0$ iff $\vec{c}_1[a] > 0$ or $\vec{c}_2[a] > 0$, for any a, $1 \leq a \leq |\Sigma|$. Thus, $\mathrm{BA}(\vec{c}_1 \downarrow \vec{c}_2) = \mathrm{BA}(\vec{c}_1) \cup \mathrm{FA}(\vec{c}_2)$.

By Definition 25, $\mathrm{RA}(\vec{c}_i) = \{a \mid 1 \leq a \leq k \wedge \vec{c}_i[a] < 0\}$, for every i, $1 \leq i \leq 3$. In analogy to the previous case, $\vec{c}_1 \downarrow \vec{c}_2[a] = \vec{c}_3[a] < 0$ iff $\vec{c}_1[a] < 0$ or $\vec{c}_2[a] < 0$, for any a, $1 \leq a \leq |\Sigma|$. Thus, $\mathrm{RA}(\vec{c}_1 \downarrow \vec{c}_2) = \mathrm{RA}(\vec{c}_1) \cup \mathrm{RA}(\vec{c}_2)$. □

Lemma 12. Let $\vec{c}_1$ and $\vec{c}_2$ be two mergeable coalition vectors and let σ be a move vector. If $\vec{c}_1 \downarrow \vec{c}_2 \sqsubseteq \sigma$ then $\vec{c}_1 \sqsubseteq \sigma$ and $\vec{c}_2 \sqsubseteq \sigma$.

Proof. Let $\vec{c}_3 = \vec{c}_1 \downarrow \vec{c}_2 \sqsubseteq \sigma$. Since $\vec{c}_3 \sqsubseteq \sigma$, (i) $\sigma[a] = f^{\vec{c}_3[a]}(s, a)$ for every $a \in \mathrm{BA}(\vec{c}_3)$ and (ii) $\sigma[a'] = g_n^{|\vec{c}_3[a']|}(s, a', (\sigma[a_1], \ldots, \sigma[a_n]))$ for every $a' \in \mathrm{RA}(\vec{c}_3)$. By Lemma 11, $\mathrm{BA}(\vec{c}_3) = \mathrm{BA}(\vec{c}_1) \cup \mathrm{BA}(\vec{c}_2)$ and $\mathrm{RA}(\vec{c}_3) = \mathrm{RA}(\vec{c}_1) \cup \mathrm{RA}(\vec{c}_2)$.

To prove $\vec{c}_1 \sqsubseteq \sigma$, we need to prove (iii) $\sigma[a] = f^{\vec{c}_1[a]}(s,a)$ for every $a \in \mathrm{BA}(\vec{c}_1)$ as well as (iv) $\sigma[a'] = g_n^{|\vec{c}_1[a']|}(s,a',(\sigma[a_1],\ldots,\sigma[a_n]))$ for every $a' \in \mathrm{RA}(\vec{c}_1)$. If $a \in \mathrm{BA}(\vec{c}_1)$, then $a \in \mathrm{BA}(\vec{c}_3)$ and also, by the definition of $\vec{c}_1{\downarrow}\vec{c}_2$ in Lemma 9, $\vec{c}_3[a] = \vec{c}_1[a]$. So, $f^{\vec{c}_1[a]} = f^{\vec{c}_3[a]}$ which means that $\sigma[a] = f^{\vec{c}_3[a]}(s,a)$ implies $\sigma[a] = f^{\vec{c}_1[a]}(s,a)$. Analogously, $a' \in \mathrm{RA}(\vec{c}_1)$, then $a' \in \mathrm{RA}(\vec{c}_3)$ and also, by the definition of $\vec{c}_1{\downarrow}\vec{c}_2$ in Lemma 9, $\vec{c}_3[a] = \vec{c}_1[a']$. So, $g_n^{|\vec{c}_1[a']|} = g_n^{|\vec{c}_3[a']|}$ which means that $\sigma[a'] = g_n^{|\vec{c}_3[a']|}(s,a',(\sigma[a_1],\ldots,\sigma[a_n]))$ implies $\sigma[a'] = g_n^{|\vec{c}_1[a']|}(s,a',(\sigma[a_1],\ldots,\sigma[a_n]))$.

That $\vec{c}_2 \sqsubseteq \sigma$ can be shown analogously. □

Definition 36. Let Γ be a finite set of coalition vectors such that for all coalition vectors $\vec{c}_1$ and $\vec{c}_2$ in Γ:

(i) for each a, $1 \leq a \leq |\vec{c}_1| = |\vec{c}_2|$, if $\vec{c}_1[a] \in \mathbb{N}$ and $\vec{c}_2[a] \in \mathbb{N}$, then $\vec{c}_1[a] = \vec{c}_2[a]$;

(ii) for each a and a', $1 \leq a < a' \leq |\vec{c}_1|$, if $\vec{c}_1[a] < 0$ and $\vec{c}_2[a'] < 0$, then $\vec{c}_1[a] = \vec{c}_2[a']$.

Then Γ is a *pairwise mergeable set of coalition vectors.*

Lemma 13. Let Γ be a pairwise mergeable set of coalition vectors. Let $\vec{c}_1$ and $\vec{c}_2$ be coalition vectors in Γ. Then $\vec{c}_1$ and $\vec{c}_2$ are mergeable and for $\Gamma' = \Gamma \cup \{\vec{c}_1{\downarrow}\vec{c}_2\}$ the following hold:

(a) For all $\vec{c}_3$ and $\vec{c}_4$ in Γ', for each a, $1 \leq a \leq |\vec{c}_3| = |\vec{c}_4|$, if $\vec{c}_3[a] \in \mathbb{N}$ and $\vec{c}_4[a] \in \mathbb{N}$, then $\vec{c}_3[a] = \vec{c}_4[a]$.

(b) For all $\vec{c}_3$ and $\vec{c}_4$ in Γ', for each a and a', $1 \leq a < a' \leq |\vec{c}_3|$, if $\vec{c}_3[a] < 0$ and $\vec{c}_4[a'] < 0$, then $\vec{c}_3[a] = \vec{c}_4[a']$.

Proof. If $\vec{c}_1 = \vec{c}_2$, then the two vectors are trivially mergeable: $\vec{c}_1{\downarrow}\vec{c}_2 = \vec{c}_1$. Also, since $\vec{c}_1{\downarrow}\vec{c}_2 \in \Gamma$, properties (a) and (b) follow from conditions (i) and (b) for Γ.

If $\vec{c}_1 \neq \vec{c}_2$, let $\vec{c}_4$ be defined as in Lemma 9:

$$\vec{c}_4[i] = \begin{cases} \vec{c}_1[a], & \text{if (1) } \vec{c}_2[a] = * \\ \vec{c}_2[a], & \text{if (2) } \vec{c}_1[a] = * \\ \vec{c}_1[a], & \text{if (3) } \vec{c}_1[a] = \vec{c}_2[i] \\ \bot, & \text{if (4) } \vec{c}_1[a] \neq * \text{ and } \vec{c}_2[a] \neq * \text{ and } \vec{c}_1[a] \neq \vec{c}_2[a] \end{cases}$$

By condition (i), for each agent a, $1 \leq a \leq |\vec{c}_1|$, if $\vec{c}_1[a] \neq *$ and $\vec{c}_2[a] \neq *$ then $\vec{c}_1[a] = \vec{c}_2[a]$. So, condition (4) in the definition of $\vec{c}_4$ never applies. Also, by condition (ii), the set $\{i \mid \vec{c}_\in\Gamma, 1 \leq a \leq |\vec{c}_|, \vec{c}[a] = i < 0\}$ is a singleton set. So, $\vec{c}_4$ cannot contain two distinct negative numbers. Thus, $\vec{c}_1$ and $\vec{c}_2$ are mergeable and $\vec{c}_1{\downarrow}\vec{c}_2 = \vec{c}_4$.

Let $\Gamma' = \Gamma \cup \{\vec{c}_1{\downarrow}\vec{c}_2\}$. As $\vec{c}_1{\downarrow}\vec{c}_2$ does not contain any negative index that was not present in $\vec{c}_1$ or $\vec{c}_2$, property (b) holds for Γ'.

Note that in order to prove property (a) for Γ', we just have to show that for all $\vec{c}_3 \in \Gamma$, for each a, $1 \leq a \leq |\vec{c}_3|$, if $\vec{c}_3[a] \in \mathbb{N}$ and $\vec{c}_4[a] = \vec{c}_1{\downarrow}\vec{c}_2[a] \in \mathbb{N}$, then $\vec{c}_3[a] = \vec{c}_1{\downarrow}\vec{c}_2[a]$. By condition (i), for each a, $1 \leq a \leq |\vec{c}_3|$, if $\vec{c}_3[a] \in \mathbb{N}$ and $\vec{c}_1[a] \in \mathbb{N}$ then $\vec{c}_3[a] = \vec{c}_1[a]$ and by definition of $\vec{c}_1{\downarrow}\vec{c}_2 = \vec{c}_4$ above, $\vec{c}_1{\downarrow}\vec{c}_2[a] = \vec{c}_1[a]$, so $\vec{c}_3[a] = \vec{c}_1{\downarrow}\vec{c}_2[a]$. Likewise, for each a, $1 \leq a \leq |\vec{c}_3|$, if $\vec{c}_3[a] \in \mathbb{N}$ and $\vec{c}_2[a] \in \mathbb{N}$ then $\vec{c}_3[a] = \vec{c}_2[a]$ and by definition of $\vec{c}_1{\downarrow}\vec{c}_2 = \vec{c}_4$ above, $\vec{c}_1{\downarrow}\vec{c}_2[a] = \vec{c}_2[a]$, so $\vec{c}_3[a] = \vec{c}_1{\downarrow}\vec{c}_2[a]$. Since $\vec{c}_1{\downarrow}\vec{c}_2[a] \in \mathbb{N}$ iff $\vec{c}_1[a] \in \mathbb{N}$ or $\vec{c}_2[a] \in \mathbb{N}$, for all $\vec{c}_3 \in \Gamma$, for each a, $1 \leq a \leq |\vec{c}_3|$, if $\vec{c}_3[a] \in \mathbb{N}$ and $\vec{c}_4[a] = \vec{c}_1{\downarrow}\vec{c}_2[a] \in \mathbb{N}$, then $\vec{c}_3[a] = \vec{c}_1{\downarrow}\vec{c}_2[a]$. □

Corollary 1. *Let Γ be a pairwise mergeable set of coalition vectors and let $\Gamma' = \Gamma \cup \{\vec{c}_1{\downarrow}\vec{c}_2\}$ for coalition vectors $\vec{c}_1$ and $\vec{c}_2$ in Γ. Then Γ' is a pairwise mergeable set of coalition vectors.*

Let $(\mathcal{I},\mathcal{U},\mathcal{N})$ be a coalition problem in $\mathsf{DSNF_{VCL}}$; P, Q be conjunctions of literals; C, D be disjunctions of literals; l, l_i be literals; and $\vec{c}$, $\vec{c}_2$ be coalition vectors.

1.	t_0	$[\mathcal{I}]$	11.	$t_0 \to (*,2,*)\neg x_2^1 \vee x_1^2$	$[\mathcal{N}$, **VRES1**, $5, 9, x_2^2]$
2.	$t_0 \to (*,*,*)\neg x_1^1 \vee \neg x_1^2$	$[\mathcal{N}]$	12.	$t_0 \to (*,*,3)\neg x_2^2 \vee x_1^3$	$[\mathcal{N}$, **VRES1**, $7, 10, x_2^3]$
3.	$t_0 \to (*,*,*)\neg x_1^1 \vee \neg x_1^3$	$[\mathcal{N}]$	13.	$t_0 \to (*,*,3)\neg x_1^1 \vee \neg x_2^2$	$[\mathcal{N}$, **VRES1**, $12, 3, x_1^3]$
4.	$t_0 \to (*,*,*)\neg x_1^2 \vee \neg x_1^3$	$[\mathcal{N}]$	14.	$t_0 \to (*,2,3)\neg x_1^1 \vee x_1^2$	$[\mathcal{N}$, **VRES1**, $13, 9, x_2^2]$
5.	$t_0 \to (*,*,*)\neg x_2^1 \vee \neg x_2^2$	$[\mathcal{N}]$	15.	$t_0 \to (*,2,3)\neg x_1^1$	$[\mathcal{N}$, **VRES1**, $14, 2, x_1^2]$
6.	$t_0 \to (*,*,*)\neg x_2^1 \vee \neg x_2^3$	$[\mathcal{N}]$	16.	$t_0 \to (*,*,3)\neg x_2^1 \vee x_1^3$	$[\mathcal{N}$, **VRES1**, $6, 10, x_2^3]$
7.	$t_0 \to (*,*,*)\neg x_2^2 \vee \neg x_2^3$	$[\mathcal{N}]$	17.	$t_0 \to (*,*,3)\neg x_2^1 \vee \neg x_1^2$	$[\mathcal{N}$, **VRES1**, $16, 4, x_1^3]$
8.	$t_0 \to (1,*,*)x_1^1 \vee x_2^1$	$[\mathcal{N}]$	18.	$t_0 \to (*,2,3)\neg x_2^1$	$[\mathcal{N}$, **VRES1**, $17, 11, x_1^2]$
9.	$t_0 \to (*,2,*)x_1^2 \vee x_2^2$	$[\mathcal{N}]$	19.	$t_0 \to (1,2,3)x_1^1$	$[\mathcal{N}$, **VRES1**, $18, 8, x_2^1]$
10.	$t_0 \to (*,*,3)x_1^3 \vee x_2^3$	$[\mathcal{N}]$	20.	$t_0 \to (1,2,3)\mathbf{false}$	$[\mathcal{N}$, **VRES1**, $19, 15, x_1^1]$
			21.	$\neg t_0$	$[\mathcal{U}$, **RW**, $20]$
			22.	**false**	$[\mathcal{I}$, **IRES1**, $21, 1, t_0]$

Figure 4: Derivation from $\mathcal{C'}_2^3$ by $\mathsf{RES}_{\mathsf{CL}}^{\succ}$

The resolution calculus $\mathsf{RES}_{\mathsf{CL}}^{\succ}$, where $\succ$ is an atom ordering, consists of the following rules:

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

$$\textbf{VRES1} \quad \begin{array}{rcll} P & \to & \vec{c}_1(C \vee l) & \in \mathcal{N} \\ Q & \to & \vec{c}_2(D \vee \neg l) & \in \mathcal{N} \\ \hline P \wedge Q & \to & \vec{c}_1 {\downarrow} \vec{c}_2(C \vee D) & \in \mathcal{N} \end{array} \qquad \textbf{VRES2} \quad \begin{array}{rcll} & & C \vee l & \in \mathcal{U} \\ Q & \to & \vec{c}(D \vee \neg l) & \in \mathcal{N} \\ \hline Q & \to & \vec{c}(C \vee D) & \in \mathcal{N} \end{array}$$

$$\textbf{RW} \quad \begin{array}{ll} \bigwedge_{i=1}^{n} l_i \to \vec{c}\,\mathbf{false} & \in \mathcal{N} \\ \hline \bigvee_{i=1}^{n} \neg l_i & \in \mathcal{U} \end{array}$$

where

- in **VRES1**, $\vec{c}_1$ and $\vec{c}_2$ are mergeable; and
- in **IRES1**, **GRES1**, **VRES1** and **VRES2**, l is be maximal with respect to C and $\neg l$ is maximal with respect to D.

Definition 37. A *derivation* from a coalition problem in $\mathsf{DSNF}_{\mathsf{VCL}}$ $\mathcal{C} = (\mathcal{I}, \mathcal{U}, \mathcal{N})$ by $\mathsf{RES}_{\mathsf{CL}}^{\succ}$ 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
- $(\mathcal{I}_i, \mathcal{U}_i, \mathcal{N}_i \cup \{D\})$, where D is the conclusion of an application of **VRES1** or **VRES2**.

Figure 4 shows a refutation of the coalition problem $\mathcal{C'}_2^3$ using the atom ordering $x_2^3 \succ x_1^3 \succ x_2^2 \succ x_1^2 \succ x_2^1 \succ x_1^1$. The crucial step in the refutation is the derivation of Clause (20) from Clauses (15) and (19). If we were to use coalition modalities as in $\mathsf{RES}_{\mathsf{CL}}$, instead of coalition vectors, then Clause (15) would correspond to the clause $t_0 \to [2,3](\neg x_1^1)$ and Clause (19) to the clause $t_0 \to [1,2,3](x_1^1)$. It would not be possible to resolve these clauses as the sets of agents involved are not disjoint.

4.3 Soundness

Lemma 14. Let $\mathcal{M}$ be a CGM and s be a state in $\mathcal{M}$. Let $\vec{c}$ be a coalition vector and $\sigma_{\mathrm{FA}(\vec{c})}$ be a $\mathrm{FA}(\vec{c})$-move at s. Then $\mathrm{FA}(\sigma_{\mathrm{FA}(\vec{c})}) = \Sigma \setminus \mathrm{FA}(\vec{c})$.

Proof. By definition, $\sigma_{\mathrm{FA}(\vec{c})}$ is a k-tuple such that $\sigma_{\mathrm{FA}(\vec{c})}(a) \in D(a, s)$ for every $a \in \mathrm{FA}(\vec{c})$ and $\sigma_{\mathrm{FA}(\vec{c})}(a') = *$ for every $a \notin \mathrm{FA}(\vec{c})$. Since the free agents $\mathrm{FA}(\sigma_{\mathrm{FA}(\vec{c})}) = \{a \mid 1 \le a \le |\Sigma| \wedge \sigma_{\mathrm{FA}(\vec{c})} = *\}$ we have $\mathrm{FA}(\sigma_{\mathrm{FA}(\vec{c})}) = \Sigma \setminus \mathrm{FA}(\vec{c})$. □

Lemma 15 (**Resolution**)**.** Let $\mathcal{M} = (\Sigma_\varphi, \mathcal{S}, s_0, d, \delta, \Pi, \pi)$ be a CGM, such that $\langle \mathcal{M}, s \rangle \models C \vee l$ and $\langle \mathcal{M}, s \rangle \models D \vee \neg l$, for some $s \in \mathcal{S}$. Then $\langle \mathcal{M}, s \rangle \models C \vee D$.

Lemma 16 (**IRES1**)**.** Let $\mathcal{C} = (\mathcal{I}, \mathcal{U}, \mathcal{N})$ be a coalition problem in $\mathsf{DSNF_{VCL}}$, such that $C \vee l \in \mathcal{I}$ 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 17 (**GRES1**)**.** Let $\mathcal{C} = (\mathcal{I}, \mathcal{U}, \mathcal{N})$ be a coalition problem in $\mathsf{DSNF_{VCL}}$, such that $C \vee l \in \mathcal{U}$ and $D \vee \neg l \in \mathcal{U}$. If $\mathcal{C}$ is satisfiable, then $(\mathcal{I}, \mathcal{U} \cup \{C \vee D\}, \mathcal{N})$ is satisfiable.

The proofs of Lemmas 15, 16 and 17 follow from the soundness of resolution for propositional logic [22].

Lemma 18 (**VRES1**)**.** Let $\mathcal{C} = (\mathcal{I}, \mathcal{U}, \mathcal{N})$ be a coalition problem in $\mathsf{DSNF_{VCL}}$, such that $P \to \vec{c}_1(C \vee l) \in \mathcal{N}$ and $Q \to \vec{c}_2(D \vee \neg l) \in \mathcal{N}$. Let $P \wedge Q \to \vec{c}_1 \downarrow \vec{c}_2(C \vee D)$ be derivable by an application of CRES1 to $P \to \vec{c}_1(C \vee l)$ and $Q \to \vec{c}_2(D \vee \neg l)$. If $\mathcal{C}$ is satisfiable, then $(\mathcal{I}, \mathcal{U}, \mathcal{N} \cup \{P \wedge Q \to \vec{c}_1 \downarrow \vec{c}_2(C \vee D)\})$ is satisfiable.

Proof. Let $\mathcal{M} = (\mathcal{F}, \Pi, \pi, F^+, F^-)$ with $\mathcal{F} = (\Sigma, \mathcal{S}, s_0, d, \delta)$ be a $\mathrm{CGM_{CF}}$ 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 $\langle \mathcal{M}, s \rangle \models P \to \vec{c}_1(C \vee l)$ and $\langle \mathcal{M}, s \rangle \models Q \to \vec{c}_2(D \vee \neg l)$. If $\langle \mathcal{M}, s \rangle \not\models P \wedge Q$, then the implication $P \wedge Q \to \vec{c}_1 \downarrow \vec{c}_2(C \vee D)$ is satisfied at s. Assume that $\langle \mathcal{M}, s \rangle \models P \wedge Q$. Then $\langle \mathcal{M}, s \rangle \models \vec{c}_1(C \vee l)$ and $\langle \mathcal{M}, s \rangle \models \vec{c}_2(D \vee \neg l)$, that is,

(i) for all $\sigma \in D(s)$, $\vec{c}_1 \sqsubseteq \sigma$ implies $\langle \mathcal{M}, \delta(s, \sigma) \rangle C \vee l$ and

(ii) for all $\sigma \in D(s)$, $\vec{c}_2 \sqsubseteq \sigma$ implies $\langle \mathcal{M}, \delta(s, \sigma) \rangle D \vee \neg l$.

Let σ be an arbitrary move vector in $D(s)$. By Lemma 12, if $\vec{c}_1 \downarrow \vec{c}_2 \sqsubseteq \sigma$ then $\vec{c}_1 \sqsubseteq \sigma$ and $\vec{c}_2 \sqsubseteq \sigma$. So, by (i) and (ii), $\langle \mathcal{M}, \delta(s, \sigma) \rangle C \vee l$ and $\langle \mathcal{M}, \delta(s, \sigma) \rangle D \vee \neg l$, which implies $\langle \mathcal{M}, \delta(s, \sigma) \rangle (C \vee l) \wedge (D \vee \neg l)$. By propositional reasoning, $\langle \mathcal{M}, \delta(s, \sigma) \rangle (C \vee D)$.

So, for all $\sigma \in D(s)$, $\vec{c}_1 \downarrow \vec{c}_2 \sqsubseteq \sigma$ implies $\langle \mathcal{M}, \delta(s, \sigma) \rangle C \vee D$, which means that $\langle \mathcal{M}, s \rangle \models \vec{c}_1 \downarrow \vec{c}_2(C \vee D)$.

Thus, $\langle \mathcal{M}, s \rangle \models \vec{c}_1 \downarrow \vec{c}_2(C \vee D)$ and $\mathcal{M}$ is a model of $(\mathcal{I}, \mathcal{U}, \mathcal{N} \cup \{P \wedge Q \to \vec{c}_1 \downarrow \vec{c}_2(C \vee D)\})$. □

Lemma 19. Let $|\Sigma| = k$ and let $\vec{c} = (*, \ldots, *)$ be a coalition vector of length k. Let $\mathcal{C} = (\mathcal{I}, \mathcal{U}, \mathcal{N})$ be a coalition problem in $\mathsf{DSNF_{VCL}}$ and $\mathcal{M}$ be a model such that $\mathcal{M} \models \mathcal{C}$. If φ is a formula in $\mathcal{U}$, then $\mathcal{M} \models (\mathcal{I}, \mathcal{U}, \mathcal{N} \cup \{\mathbf{true} \to \vec{c}\varphi\})$.

Lemma 20 (**CRES2**)**.** Let $\mathcal{C} = (\mathcal{I}, \mathcal{U}, \mathcal{N})$ be a coalition problem in $\mathsf{DSNF_{VCL}}$, such that $(C \vee l) \in \mathcal{U}$ and $C \to \vec{c}(D \vee \neg l) \in \mathcal{N}$. If $\mathcal{C}$ is satisfiable, then $(\mathcal{I}, \mathcal{U}, \mathcal{N} \cup \{C \to \vec{c}(C \vee D)\})$ is satisfiable.

Lemma 21 (**RW1**)**.** Let $\mathcal{C} = (\mathcal{I}, \mathcal{U}, \mathcal{N})$ be a coalition problem in $\mathsf{DSNF_{VCL}}$, such that $P \to \vec{c}\mathbf{false} \in \mathcal{N}$. If $\mathcal{C}$ is satisfiable, then $(\mathcal{I}, \mathcal{U} \cup \{\neg P\}, \mathcal{N})$ is satisfiable.

Theorem 7 (Soundness of $\mathsf{RES}^{\succ}_{\mathsf{CL}}$)**.** *Let $\mathcal{C}$ be a coalition problem in $\mathsf{DSNF_{CL}}$. Let $\mathcal{C}'$ be the coalition problem in $\mathsf{DSNF_{VCL}}$ obtained from $\mathcal{C}$ by applying any of the inference rules* **IRES1***,* **GRES1***,* **CRES1***,* **CRES2** *and* **RW1** *to $\mathcal{C}$. If $\mathcal{C}$ is satisfiable, then $\mathcal{C}'$ is satisfiable.*

4.4 Termination

Regarding termination, assume that we start a derivation with a coalition problem $\mathcal{C}$. The number of propositional symbols in $\mathcal{C}$ 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. As we keep propositional conjunctions and disjunction in simplified form, there are only finitely many that may occur in a derivation. Also, in $\mathcal{C}$ only a finite set $I \subset \mathbb{Z}$ of numbers occurs in coalition vectors and all coalition vectors in $\mathcal{C}$ have the same length, say, k. Then the number of coalition vectors that may occur in a derivation is bounded by $(|I| + 1)^k$. 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 8. *Let $\mathcal{C} = (\mathcal{I}, \mathcal{U}, \mathcal{N})$ be a coalition problem in $\mathsf{DSNF_{VCL}}$. Then any derivation from $\mathcal{C}$ by $\mathsf{RES}^{\succ}_{\mathsf{CL}}$ terminates.*

4.5 Completeness

In our completeness proof for $\mathsf{RES}^{\succ}_{\mathsf{CL}}$ we show that a refutation of a CL formula φ by a tableau procedure can be used to guide the construction of a refutation of φ in $\mathsf{RES}^{\succ}_{\mathsf{CL}}$. The tableau procedure used in the proof is an adaptation of Goranko and Shkatov's tableau procedure for ATL [9]. The procedure proceeds in two phases, a construction phase in which a graph structure for φ is build, and a elimination phase in which parts of the graph that cannot be used to create a CGM for φ are deleted. The formula φ is satisfiable iff at the end of the elimination phase a non-empty graph remains.

In order to be able to use Goranko and Shkatov's tableau procedure with minimal changes, we need to transform coalition problems into a single formula. This is possible in the logic CL^+, the extension CL with the ATL-operator $\langle\langle\emptyset\rangle\rangle\Box$ which we only allow to occur positively in CL^+ formulae. The semantics of the $\langle\langle\emptyset\rangle\rangle\Box$ is defined in terms of a run:

Definition 38. 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.

Definition 39. Let $\mathcal{F} = (\Sigma, \mathcal{S}, s_0, d, \delta)$ be a CGF. A *strategy $F_\emptyset$ for $\emptyset$* (or $\emptyset$-strategy) at a state $s \in \mathcal{S}$ 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 $i \geq 0$.

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 $i \geq 0$.

We extend the definition of *positive coalition formulae* to include formulae of the form $[\mathcal{A}]\varphi$, where φ is a CL^+ formula. Negative coalition formulae and coalition formulae are defined as before.

Let $\mathcal{C} = (\mathcal{I}, \mathcal{U}, \mathcal{N})$ be a coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$. Using the operator $\langle\langle\emptyset\rangle\rangle\Box$, we define $[\mathcal{C}]_{\mathsf{CL}^+}$ to be the CL^+ formulae

$$\textstyle\bigwedge_{D\in\mathcal{I}} D \wedge \bigwedge_{D'\in\mathcal{U}\cup\mathcal{N}} \langle\langle\emptyset\rangle\rangle\Box D'.$$

Then $\mathcal{C}$ is satisfiable iff $[\mathcal{C}]_{\mathsf{CL}^+}$ is satisfiable.

We also make two adaptations to the Goranko and Shkatov's tableau procedure. First, we adapt the procedure to coalition problems instead of full ATL, by removing one of the elimination rules that is only relevant for language constructs specific to ATL. Second, we slightly change the notion of a downward saturated set in order to be able to establish a correspondence between deletions during the elimination phase and resolution inferences.

In the following, we briefly present the adapted tableau procedure. Before we present the construction phase, we give two definitions that will be used later.

Definition 40. Let Δ be a set of CL^+ formulae. We say that Δ is *downward saturated* if Δ satisfies the following properties:

i. If $\neg\neg\varphi \in \Delta$, then $\varphi \in \Delta$;

ii. If $\bigwedge_{i=1}^n \varphi_i \in \Delta$ or $\neg(\bigvee_{i=1}^n \varphi_i) \in \Delta$, then $\varphi_i \in \Delta$ for every i, $1 \leq i \leq n$;

iii. If $\langle\langle\emptyset\rangle\rangle\Box\varphi \in \Delta$, then $\{\varphi, [\emptyset]\langle\langle\emptyset\rangle\rangle\Box\varphi\} \subseteq \Delta$;

iv. If $\bigvee_{i=1}^n \varphi_i \in \Delta$ or $\neg(\bigwedge_{i=1}^n \varphi_i) \in \Delta$, then $\varphi_i \in \Delta$ for some i, $1 \leq i \leq n$.

v. If $(\varphi \to \psi) \in \Delta$, then $\neg\varphi \in \Delta$ or $\{\varphi, \psi\} \subseteq \Delta$.

Property (v) differs from Goranko and Shkatov's definition. The later requires that if $\varphi \to \psi \in \Delta$ then $\neg\varphi \in \Delta$ or $\psi \in \Delta$, which corresponds to an analytic cut on $\varphi \to \psi$. Our definition corresponds to a semantic cut on φ, creating cases $\neg\varphi$ and φ, following by an application of modus ponens to φ and $(\varphi \to \psi)$ in the second case.

Definition 41. 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:

i. $\Gamma \subseteq \Delta$;

ii. Δ is downward saturated;

iii. 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 [31]. 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 $\langle\langle\emptyset\rangle\rangle\Box$ operator. According to the α decomposition rules in [9], $\langle\langle\emptyset\rangle\rangle\Box\varphi$ should be decomposed into φ and $\langle\langle\emptyset\rangle\rangle\bigcirc\langle\langle\emptyset\rangle\rangle\Box\varphi$. The ATL formula $\langle\langle\emptyset\rangle\rangle\bigcirc\langle\langle\emptyset\rangle\rangle\Box\varphi$ corresponds to the CL^+ formula $[\emptyset]\langle\langle\emptyset\rangle\rangle\Box\varphi$, which explains the decomposition rule we give for $\langle\langle\emptyset\rangle\rangle\Box\varphi$. 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 $[\Sigma_\Phi]\mathbf{true}$ 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 $[\Sigma_\varphi]\mathbf{true}$ 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

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

   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 $prestates(\Delta) = \{\Gamma \mid \Delta \xrightarrow{\sigma} \Gamma \text{ for some } \sigma \in D(\Delta)\}$. 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 axiomatisation 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 on page 32); 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 $\mathcal{T}^{\Phi}_{m+1}$ from $\mathcal{T}^{\Phi}_{m}$ by eliminating Δ, i.e. let $\mathcal{S}^{\Phi}_{m+1} = \mathcal{S}^{\Phi}_{m} \setminus \{\Delta\}$;

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

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

Theorem 9. *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. 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. That our modification of the definition of a downward saturated set does not affect soundness and completeness follows from the fact that $\varphi \rightarrow \psi$ and $\neg\varphi \vee (\varphi \wedge \psi)$ are equivalent formulae and that we could replace all occurrences of $\varphi \rightarrow \psi$ in Φ by $\neg\varphi \vee (\varphi \wedge \psi)$ to achieve the same effect. □

In the following we use $\mathcal{P}^{\mathcal{C}}$ to denote $\mathcal{P}_0^{\{[\mathcal{C}]_{\mathsf{CL}^+}\}}$, that is, the pretableau for $\{[\mathcal{C}]_{\mathsf{CL}^+}\}$, also called the *pretableau for* $\mathcal{C}$, $\mathcal{T}_0^{\mathcal{C}}$ to denote $\mathcal{T}_0^{\{[\mathcal{C}]_{\mathsf{CL}^+}\}}$, that is, the initial tableau for $\{[\mathcal{C}]_{\mathsf{CL}^+}\}$, also called the *inital tableau for* $\mathcal{C}$, $\mathcal{T}_+^{\mathcal{C}}$ to denote the result of exhaustively applying the deletion rule **E1** to $\mathcal{T}_0^{\mathcal{C}}$, and $\mathcal{T}^{\mathcal{C}}$ to denote $\mathcal{T}^{\{[\mathcal{C}]_{\mathsf{CL}^+}\}}$, that is, the final tableau for $\{[\mathcal{C}]_{\mathsf{CL}^+}\}$, also called the *final tableau for* $\mathcal{C}$.

Lemma 22. Let $\mathcal{C}$ be a coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$. The tableau construction for $\mathcal{C}$ terminates in time exponential in the size of $\mathcal{C}$ and $\mathcal{C}$ is unsatisfiable if, and only if, the final tableau for $\mathcal{C}$, $\mathcal{T}^{\mathcal{C}}$, is closed.

Proof. The tableau construction for $\mathcal{C}$ is a tableau construction for the set $\{[\mathcal{C}]_{\mathsf{CL}^+}\}$ where the formula $[\mathcal{C}]_{\mathsf{CL}^+}$ satisfiable iff $\mathcal{C}$ is satisfiable and the size of $[\mathcal{C}]_{\mathsf{CL}^+}$ is linear in the size of $\mathcal{C}$. By Theorem 9, the tableau construction for $[\mathcal{C}]_{\mathsf{CL}^+}$ terminates in time exponential in the size of $[\mathcal{C}]_{\mathsf{CL}^+}$, and therefore in time exponential in the size of $\mathcal{C}$, and $\mathcal{T}^{\{[\mathcal{C}]_{\mathsf{CL}^+}\}}$ is closed iff $\{[\mathcal{C}]_{\mathsf{CL}^+}\}$ is unsatisfiable iff $\mathcal{C}$ is unsatisfiable. □

Recall that a derivation, as given in Definition 16, is a finite sequence $\mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_n$ of coalition problems 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 $\mathcal{T}_+^{\mathcal{C}_0}, \mathcal{T}_+^{\mathcal{C}_1}, \mathcal{T}_+^{\mathcal{C}_2}, \ldots, \mathcal{T}_+^{\mathcal{C}_n}$.

Lemma 23. Let $\mathcal{C} = (\mathcal{I}, \mathcal{U}, \mathcal{N})$ be a coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$. 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 $\gamma \in \mathcal{U} \cup \mathcal{N}$, then the following holds:

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

Proof. The construction of the tableau proceeds by alternate rounds of applications of the rules **SR** and **Next**.

1. Assume that $\langle\langle\emptyset\rangle\rangle\Box\gamma$ 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 γ and $[\emptyset]\langle\langle\emptyset\rangle\rangle\Box\gamma$. Thus, every state created from Γ contains γ.

2. Assume that Δ is a state that contains $[\emptyset]\langle\langle\emptyset\rangle\rangle\Box\gamma$. 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 $\langle\langle\emptyset\rangle\rangle\Box\gamma$.

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

Lemma 24. 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 $\gamma \in \mathcal{U} \cup \mathcal{N}$, then $\gamma \in \Delta$, for all $\Delta \in \mathcal{S}_0^{\mathcal{C}}$.

Proof. From Lemma 23, if $\gamma \in \mathcal{U} \cup \mathcal{N}$, then γ 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 γ. ☐

Lemma 25. Let $\mathcal{C} = (\mathcal{I}, \mathcal{U}, \mathcal{N})$ be a coalition problem in $\mathsf{DSNF_{CL}}$ and $P \to \psi$ be a clause in $\mathcal{N}$, where $P = l_1 \wedge \cdots \wedge l_n$, for some $n \geq 0$. Let $\mathcal{T}_+^{\mathcal{C}}$ be the tableau for $\mathcal{C}$ after the **E1** has been exhaustively and let Δ be a state in $\mathcal{T}_+^{\mathcal{C}}$. If $\{l_1, \ldots, l_n\} \subseteq \Delta$, then $\psi \in \Delta$.

Proof. If $P \to \psi$ is in $\mathcal{N}$, then by Lemma 24, $P \to \psi$ is in every state of $\mathcal{T}^{\mathcal{C}}$. If $n = 0$, then P is the empty conjunction (**true**). Because Δ is downward saturated, it must contain either $\neg$**true** or both **true** and ψ. As states containing $\neg$**true** are removed by applications of **E1**, Δ must contain ψ. If $n > 0$, assume $\{l_0, \ldots, l_n\} \subseteq \Delta$. As states are downward saturated, by Definition definition:downward saturated, every state contains either a literal in $\{\neg l_1, \ldots, \neg l_n\}$ or both P and ψ. 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 $\psi \in \Delta$. ☐

Lemma 26. Let $\mathcal{C} = (\mathcal{I}, \mathcal{U}, \mathcal{N})$ be a coalition problem in $\mathsf{DSNF_{CL}}$ and $\mathcal{T}_+^{\mathcal{C}}$ be the tableau for $\mathcal{C}$ after the **E1** has been exhaustively applied. Then $\mathcal{T}_+^{\mathcal{C}}$ is closed iff $\mathcal{I} \cup \mathcal{U}$ is unsatisfiable.

Proof. By Lemma 24, if $\gamma \in \mathcal{U} \cup \mathcal{N}$, then $\gamma \in \Delta$, for all $\Delta \in \mathcal{T}_0^{\mathcal{C}}$ and, therefore, γ is in every initial state. By construction, if $\gamma \in \mathcal{I}$, because γ is in the initial prestate and states are downward saturated, then γ is in all initial states.

Let us first assume that $\mathcal{T}_+^{\mathcal{C}}$ is closed. Then all initial states have been eliminated by **E1**, that is, all initial states contain propositional inconsistencies. If all initial states are inconsistent, we have that

$$\bigwedge_{\gamma \in \mathcal{I}} \gamma \wedge \bigwedge_{\gamma' \in \mathcal{U}} \gamma' \wedge \bigwedge_{(P \to [A]C) \in \mathcal{N}} (\neg P \vee (P \wedge [A]C)) \bigwedge_{(P' \to \langle A' \rangle C') \in \mathcal{N}} (\neg P' \vee (P' \wedge \langle A \rangle C'))$$

is unsatisfiable. Since coalition modalities are not propositional, they do not contribute to the propositional (un)satisfiability of a state and can be ignored, that is, assumed to be true.

Therefore, the formula

$$\bigwedge_{\gamma \in \mathcal{I}} \gamma \wedge \bigwedge_{\gamma' \in \mathcal{U}} \gamma' \wedge \bigwedge_{(P \to [A]C) \in \mathcal{N}} (\neg P \vee P) \bigwedge_{(P' \to \langle A' \rangle C') \in \mathcal{N}} (\neg P' \vee P')$$

must be unsatisfiable. Now, obviously, tautologies $(\neg P \vee P)$ and $(\neg P' \vee P')$ also do not contribute to the unsatisfiability of this formula which implies that

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

is unsatisfiable and thus $\mathcal{I} \cup \mathcal{U}$ is unsatisfiable.

Now assume that $\mathcal{I} \cup \mathcal{U} = \{\gamma_1, \ldots, \gamma_n\}$ is unsatisfiable and let Δ be an arbitrary initial state. Since each the elements of $\mathcal{I}$ and $\mathcal{U}$ are propositional disjunctions and Δ is downward closed, for each $\gamma_i \in \mathcal{I} \cup \mathcal{U}$ with $\gamma_i = l_1^i \vee \cdots \vee l_{m_i}^i$, there is some $l_{j_i}^i$, $1 \leq j_i \leq m_i$, with $l_{j_i}^i$ in Δ. Let $V = \{l_{j_1}^1, \ldots, l_{j_n}^n\}$ and $V' = V \cap \Pi$. If V is consistent, that is, there is no propositional symbols $p \in \Pi$ such that $p \in V$ and $\neg p \in V$, then V' is a valuation that satisfies $\mathcal{I} \cup \mathcal{U}$, contradicting that $\mathcal{I} \cup \mathcal{U}$ is unsatisfiable. If V is not consistent then there exists a propositional symbol $p \in \Pi$ with $p \in V \subset \Delta$ and $\neg p \in V \subset \Delta$, which means that Δ can be eliminated by an application of **E1** and could not occur in $\mathcal{T}_+^{\mathcal{C}}$. ☐

The proof of Lemma 27 below uses the completeness of propositional ordered resolution:

Theorem 10 (Completeness of propositional ordered resolution [15]). *Let $\succ$ be a well-founded and total ordering $\succ$ on the set Π. If a set $\mathcal{S}$ of propositional clauses over Π is unsatisfiable, then there is a refutation from $\mathcal{S}$ by $\mathsf{RES}^{\succ}$, where the inference rule RES is given by $\{(C \vee l), (D \vee \neg l)\} \vdash (C \vee D)$, if l is maximal with respect to C and $\neg l$ is maximal with respect to D.*

Lemma 27. Let $\mathcal{C} = (\mathcal{I}, \mathcal{U}, \mathcal{N})$ be a coalition problem in $\mathsf{DSNF}_{\mathsf{VCL}}$. If $\mathcal{I} \cup \mathcal{U}$ is unsatisfiable, there is a refutation for $\mathcal{I} \cup \mathcal{U}$ by $\mathsf{RES}^{\succ}_{\mathsf{CL}}$ using only the inference rules **IRES1** and **GRES1**.

Proof. If $\mathcal{I} \cup \mathcal{U}$ is unsatisfiable, by Theorem 10, there is a refutation by ordered resolution with the ordering $\succ$ from $\mathcal{I} \cup \mathcal{U}$. Let $\mathcal{S}_0, \ldots, \mathcal{S}_n$, with $n \in \mathbb{N}$, be a sequence of sets of propositional clauses, where $\mathcal{S}_0 = \mathcal{I} \cup \mathcal{U}$, $\mathbf{false} \in \mathcal{S}_n$, and, for each $1 \leq i \leq n$, $\mathcal{S}_{i+1}$ is the set of clauses obtained by adding to $\mathcal{S}_i$ the resolvent of an application of the ordered resolution rule $\mathsf{RES}^{\succ}$ to clauses in $\mathcal{S}_i$. We inductively construct a refutation $\mathcal{C}_0, \ldots, \mathcal{C}_n$ for $\mathcal{C} = (\mathcal{I}, \mathcal{U}, \mathcal{N})$ such that for every i, $1 \leq i \leq n$, $\mathcal{S}_i = \mathcal{I}_i \cup \mathcal{U}_i$, as follows. In the base case, $\mathcal{C}_0 = \mathcal{C}$ and clearly $\mathcal{S}_0 = \mathcal{I} \cup \mathcal{U} = \mathcal{I}_0 \cup \mathcal{U}_0$. For the induction step, let $\mathcal{C}_0, \ldots, \mathcal{C}_i$ be the derivation already constructed. In $\mathcal{S}_0, \ldots, \mathcal{S}_i, \mathcal{S}_{i+1}$, we obtained $(D \vee D')$ by an application of $\mathsf{RES}^{\succ}$ to $(D \vee l)$, where l is maximal with respect to D, and $(D' \vee \neg l)$, $\neg l$ is maximal with respect to D', in $\mathcal{S}_i$. As $\mathcal{S}_i = \mathcal{I}_i \cup \mathcal{U}_i$, every clause in $\mathcal{S}_i$ occurs in $\mathcal{I}_i$ or $\mathcal{U}_i$ (or both). We say that a clause D *originates* from $\mathcal{I}_i$ if D is in $\mathcal{I}_i$, otherwise we say that D *originates* from $\mathcal{U}_i$.

(i) If both $(D \vee l)$ and $(D' \vee \neg l)$ in $\mathcal{S}_i$ 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 $\mathcal{S}_{i+1} = \mathcal{I}_{i+1} \cup \mathcal{U}_{i+1}$;

(ii) If $(D \vee l) \in \mathcal{S}_i$ originates from a clause in $\mathcal{I}_i$ $\mathcal{C}_{i+1} = (\mathcal{I}_i \cup \{D \vee D'\}, \mathcal{U}_i, \mathcal{N}_i)$, 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 $\mathcal{S}_{i+1} = \mathcal{I}_{i+1} \cup \mathcal{U}_{i+1}$;

(iii) If $(D' \vee \neg l) \in \mathcal{S}_i$ originates from a clause in $\mathcal{I}_i$ then we proceed as in case (ii) using **IRES1** to construct $\mathcal{C}_{i+1}$.

Note that applications of **IRES1** and **GRES1** as described above are possible as l is maximal with respect to D and $\neg l$ is maximal with respect to D'.

By construction, since $\mathbf{false} \in \mathcal{S}_n$, we have $\mathbf{false} \in \mathcal{I}_n \cup \mathcal{U}_n$, and thus there is a refutation of $\mathcal{C}$ in $\mathsf{RES}^{\succ}_{\mathsf{CL}}$ using only the inference rules **IRES1** and **GRES1**. ☐

Lemma 28. Let $\mathcal{C} = (\mathcal{I}, \mathcal{U}, \mathcal{N})$ be an unsatisfiable coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$ such that $\mathcal{I} \cup \mathcal{U}$ is satisfiable and let $\mathcal{C}' = (\mathcal{I}, \mathcal{U}, \mathcal{N}')$ be the coalition problem in $\mathsf{DSNF}_{\mathsf{VCL}}$ resulting from exhaustively applying the rules of rewriting system R_4 to $\mathcal{C}$. Let $\Delta \in \mathcal{T}^{\mathcal{C}}_{+}$ be the first state to be eliminated by rule **E2** in the state elimination phase that will result in the final tableau $\mathcal{T}^{\mathcal{C}}$ for $\mathcal{C}$. Then we can derive a global clause $C \notin \mathcal{U}$ from $\mathcal{C}'$ by $\mathsf{RES}^{\succ}_{\mathsf{CL}}$.

Proof. Since $\mathcal{I} \cup \mathcal{U}$ is satisfiable, $\mathcal{T}^{\mathcal{C}}_{+}$ is not closed, but since $\mathcal{C}$ is unsatisfiable, the final tableau $\mathcal{T}^{\mathcal{C}}$ must be closed. Therefore, $\mathcal{T}^{\mathcal{C}}_{+}$ contains at least one state that can be deleted by an application of the deletion rule **E2**. Let Δ be the first state to which rule **E2** is applied. By 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 for any coalition formula $\varphi \in \Delta$ let $pos(\varphi, \mathfrak{L}(\Delta))$ be the position of φ in $\mathfrak{L}(\Delta)$. From Lemma 24, all clause in $\mathcal{U}$ and in $\mathcal{N}$ are in Δ. By Lemma 25, the right-hand side of coalition clauses are in the states where the left-hand side is satisfied. Therefore, by Lemmas 24 and 25, 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 $\mathcal{U} \cup \mathcal{P}_0^+ \cup \mathcal{P}_0^- \cup \mathcal{N}$ where

$$\begin{aligned}
\mathcal{P}_0^+ &= \{C \mid [\mathcal{A}]C \in \mathcal{E}_0^+\}\\
\mathcal{E}_0^+ &= \{[\mathcal{A}]C \mid P \to [\mathcal{A}]C \in \mathcal{N}_0^+\}\\
\mathcal{L}_0^+ &= \{P \mid P \to [\mathcal{A}]C \in \mathcal{N}_0^+\}\\
\mathcal{N}_0^+ &= \{P \to [\mathcal{A}]C \mid P \to [\mathcal{A}]C \in \mathcal{N}, \Delta \models P, \sigma_a = pos([\mathcal{A}]C, \mathfrak{L}(\Delta)) \text{ for all } a \in \mathcal{A}\}\\
\mathcal{P}_0^- &= \{C \mid \langle\mathcal{A}\rangle C \in \mathcal{E}_0^-\} \setminus \mathcal{P}_0^+\\
\mathcal{E}_0^+ &= \{\langle\mathcal{A}\rangle C \mid P \to \langle\mathcal{A}\rangle C \in \mathcal{N}_0^-\}\\
\mathcal{L}_0^- &= \{P \mid P \to \langle\mathcal{A}\rangle C \in \mathcal{N}_0^-\}\\
\mathcal{N}_0^- &= \{P \to \langle\mathcal{A}\rangle C \mid P \to \langle\mathcal{A}\rangle C \in \mathcal{N}, \Delta \models P, \Sigma_{\mathcal{C}} \setminus \mathcal{A} \subseteq N(\sigma), neg(\sigma) = pos(\langle\mathcal{A}\rangle C, \mathfrak{L}(\Delta))\}
\end{aligned}$$

Note that for every clause $P \to [A]C$ in $\mathcal{N}$ there is a corresponding clause $P \to \vec{c}^{\,i}_{\mathcal{A}}C$ in $\mathcal{N}'$ and for every clause $P \to \langle A\rangle C$ in $\mathcal{N}$ there is a corresponding clause $P \to \vec{c}^{\,-i}_{\mathcal{A}}C$ in $\mathcal{N}'$.

We associate each C in $\mathcal{U} \cup \mathcal{P}_0^+ \cup \mathcal{P}_0^-$ with a subset of $\mathcal{U} \cup \mathcal{N}'$ as follows.

$$\begin{aligned}
cl^u(C) &= \{C\} \cap \mathcal{U}\\
cl^+(C) &= \{P \to \vec{c}^{\,i}_{\mathcal{A}}C \mid P \to [\mathcal{A}]C \in \mathcal{N}, \Delta \models P, \sigma_a = pos([\mathcal{A}]C, \mathfrak{L}(\Delta))\}\\
cl^-(C) &= \{P \to \vec{c}^{\,-i}_{\mathcal{A}}C \mid P \to \langle\mathcal{A}\rangle C \in \mathcal{N}, \Delta \models P, \Sigma_{\mathcal{C}} \setminus \mathcal{A} \subseteq N(\sigma), neg(\sigma) = pos(\langle\mathcal{A}\rangle C, \mathfrak{L}(\Delta))\}\\
cl(C) &= \begin{cases} cl^u(C) & \text{if } cl^u(C) \neq \emptyset\\ cl^+(C) & \text{if } cl^u(C) = \emptyset \text{ and } cl^+(C) \neq \emptyset\\ cl^-(C) & \text{otherwise}\end{cases}
\end{aligned}$$

Note that by construction, $cl(C)$ is non-empty for every $C \in \mathcal{U} \cup \mathcal{P}_0^+ \cup \mathcal{P}_0^-$. With each $C \in \mathcal{U} \cup \mathcal{P}_0^+ \cup \mathcal{P}_0^-$, we then uniquely associate a clause κ_C, with $\kappa_C = C$ or $\kappa_C = P_C \to \vec{c}^{\,i}_{\mathcal{A}}C$ or $\kappa_C = P_C \to \vec{c}^{\,-i}_{\mathcal{A}}C$ in $cl(C)$. If $cl(C)$ contains more than one clause, then we can choose κ_C arbitrarily among the elements of $cl(C)$.

Recall that (a-i) for any $[\mathcal{A}]C, [\mathcal{A}']C' \in \mathcal{E}_0^+$ with $[\mathcal{A}]C \neq [\mathcal{A}']C'$ we have $\mathcal{A} \cap \mathcal{A}' = \emptyset$, (a-ii) $\mathcal{N}_0^-$, $\mathcal{E}_0^-$ and $\mathcal{P}_0^-$ are either all empty sets or all singleton sets, and (a-iii) for any $[\mathcal{A}]C \in \mathcal{E}_0^+$ and $\langle\mathcal{A}'\rangle C' \in \mathcal{E}_0^-$ we have $\mathcal{A} \subseteq \mathcal{A}'$. Let Γ be the set of coalition vectors defined by:

$$\begin{aligned}
\Gamma = &\{\vec{c}^{\,i}_{\mathcal{A}} \mid P \to [\mathcal{A}]C \in \mathcal{N}, \Delta \models P, \sigma_a = pos([\mathcal{A}]C, \mathfrak{L}(\Delta))\}\\
&\cup \{\vec{c}^{\,-i}_{\mathcal{A}} \mid P \to \langle\mathcal{A}\rangle C \in \mathcal{N}, \Delta \models P, \Sigma_{\mathcal{C}} \setminus \mathcal{A} \subseteq N(\sigma), neg(\sigma) = pos(\langle\mathcal{A}\rangle C, \mathfrak{L}(\Delta))\}
\end{aligned}$$

It follows from properties (a-i) to (a-iii) that Γ is a pairwise mergeable set of coalition vectors.

Since Δ' is not in $\mathcal{T}_+^{\mathcal{C}}$, 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. As the formulae in $\mathcal{N}$ do not contribute to propositional inconsistencies in Δ', the set of propositional clauses $\mathcal{S}_0 = \mathcal{U}_0 \cup \mathcal{P}_0^+ \cup \mathcal{P}_0^-$, with $\mathcal{U}_0 = \mathcal{U}$, is unsatisfiable. Since $\mathcal{S}_0$ is unsatisfiable, there must be a refutation by ordered resolution for this set. Let $\mathcal{S}_0, \ldots, \mathcal{S}_n$, with $n \in \mathbb{N}$, be a sequence of sets of propositional clauses, where $\mathcal{S}_n$ contains the constant **false** and, for each $1 \leq i \leq n$, $\mathcal{S}_{i+1}$ is the set of clauses obtained by adding to $\mathcal{S}_j$ the resolvent of an application of $\mathbf{RES}^{\succ}$ to clauses in $\mathcal{S}_j$.

We inductively construct

(a) a derivation $\mathcal{C}'_0, \ldots, \mathcal{C}'_n$ such that $\mathcal{C}'_j = (\mathcal{I}, \mathcal{U}_j, \mathcal{N}'_j)$ for every j, $1 \leq j \leq n$, and $\mathcal{C}'_n$ contains a clause of the form $P \to \vec{c}\,\mathbf{false}$ such that P is conjunction of elements of $\mathcal{L}_0^+ \cup \mathcal{L}_0^-$;

(b) a sequence $\mathcal{M}'_0, \ldots, \mathcal{M}'_n$ of sets of coalition clauses such that for every j, $1 \leq j \leq n$, $\mathcal{M}'_j \subseteq \mathcal{N}'_j$ and $\{\kappa_C \mid C \in \mathcal{S}_j\} \subseteq \mathcal{M}'_j \cup \mathcal{U}_j$ and for every $P \to \vec{c}C \in \mathcal{M}'_j$, P is a conjunction of elements of $\mathcal{L}_0^+ \cup \mathcal{L}_0^-$;

(c) a sequence $\Gamma_0, \ldots, \Gamma_n$ of sets of pairwise mergeable coalition vectors such that for every j, $1 \leq j \leq n$, $\{\vec{c} \mid P \to \vec{c}C \in \mathcal{M}'_j\} \subseteq \Gamma_j$.

At the same time we extend the association between propositional clauses C in $\mathcal{S}_j$ and clauses κ_C in $\mathcal{C}'_j$, for all j, $1 \leq j \leq n$.

In the base case, $\mathcal{C}'_0 = \mathcal{C}'$, $\mathcal{M}'_0 = \{P \to \vec{c}^{\,i}_{\mathcal{A}} C \mid P \to [\mathcal{A}]C \in \mathcal{N}, \Delta \models P, \sigma_a = pos([\mathcal{A}]C, \mathfrak{L}(\Delta))\} \cup \{P \to \vec{c}^{\,-i}_{\mathcal{A}} C \mid P \to \langle\mathcal{A}\rangle C \in \mathcal{N}, \Delta \models P, \Sigma_{\mathcal{C}} \setminus \mathcal{A} \subseteq N(\sigma), neg(\sigma) = pos(\langle\mathcal{A}\rangle C, \mathfrak{L}(\Delta))\}$, $\Gamma_0 = \Gamma$, and for every clause C in $\mathcal{S}_0$ we have already defined a corresponding clause κ_C. Note that Γ_0 is the set of all coalition vectors occurring in $\mathcal{M}'_0$, $\{\kappa_C \mid C \in \mathcal{S}_0\} \subseteq \mathcal{M}'_0 \cup \mathcal{U}$, and for every $P \to \vec{c}C \in \mathcal{M}'_0$, P is an element of $\mathcal{L}^+_0 \cup \mathcal{L}^-0$.

For the induction step, assume that we have already constructed the derivation $\mathcal{C}'_0, \ldots, \mathcal{C}'_j = (\mathcal{I}, \mathcal{U}_j, \mathcal{N}'_j)$, the sequence $\mathcal{M}'_0, \ldots, \mathcal{M}'_j$ of coalition clauses, and the sequence $\Gamma_0, \ldots, \Gamma_j$ such that $\mathcal{M}'_j \subseteq \mathcal{N}'_j$; $\{\kappa_C \mid C \in \mathcal{S}_j\} \subseteq \mathcal{M}'_j \cup \mathcal{U}_j$'; for all $P \to \vec{c}C \in \mathcal{M}'_j$, P is a conjunction of elements of $\mathcal{L}^+_0 \cup \mathcal{L}^-_0$; $\{\vec{c} \mid P \to \vec{c}C \in \mathcal{M}'_j\} \subseteq \Gamma_j$; and Γ_j is a pairwise mergeable set of coalition clauses.

In $\mathcal{S}_0, \ldots, \mathcal{S}_j, \mathcal{S}_{j+1}$, we obtained $(C \vee D)$ by an application of **RES**$^{\succ}$ to $(C \vee l) \in S_j$, where l is maximal with respect to C, and $(D \vee \neg l) \in S_j$, where $\neg l$ is maximal with respect to D.

Depending on whether $\kappa_{(C \vee l)}$ and $\kappa_{(D \vee \neg l)}$ are universal or coalition clauses we consider the following cases:

1. Assume $\kappa_{(C \vee l)} = P \to \vec{c}_1(C \vee l)$ and $\kappa_{(D \vee \neg l)} = Q \to \vec{c}_2(D \vee \neg l)$. By induction hypothesis, $\kappa_{(C \vee l)}$ and $\kappa_{(D \vee \neg l)}$ are both in $\mathcal{M}'_j$ and $\vec{c}_1$ and $\vec{c}_2$ are both in Γ_j. So, $\vec{c}_1$ and $\vec{c}_2$ are mergeable, let $\vec{c}_3 = \vec{c}_1 \downarrow \vec{c}_2$. Using **VRES1** we can derive the clause $\gamma_{j+1} = P \wedge Q \to \vec{c}_3(C \vee D)$. Let $\mathcal{C}'_{j+1} = (\mathcal{I}, \mathcal{U}_{j+1}, \mathcal{N}'_{j+1})$ with $\mathcal{U}_{j+1} = \mathcal{U}_j$ and $\mathcal{N}'_{j+1} = \mathcal{N}'_j \cup \{\gamma_{j+1}\}$, $\mathcal{M}'_{j+1} = \mathcal{M}'_j \cup \{\gamma_{j+1}\}$, $\Gamma_{j+1} = \Gamma_j \cup \{\vec{c}_3\}$, $\kappa_{(C \vee D)} = \gamma_{j+1}$.

   We need to establish that these definitions satisfy the conditions for our construction:

   - By induction hypothesis, $\mathcal{M}'_j \subseteq \mathcal{N}'_j$ and since by our definition $\mathcal{N}'{}_{j+1} = \mathcal{N}'{}_j \cup \{\gamma_{j+1}\}$ and $\mathcal{M}'{}_{j+1} = \mathcal{M}'{}_j \cup \{\gamma_{j+1}\}$, we have $\mathcal{M}'_{j+1} \subseteq \mathcal{N}'_{j+1}$.
   - Also, we have $\mathcal{S}_{j+1} = \mathcal{S}_j \cup \{(C \vee D)\}$ and, by induction hypothesis, $\{\kappa_C \mid C \in \mathcal{S}_j\} \subseteq \mathcal{M}'_j \cup \mathcal{U}_j$. So, $\{\kappa_C \mid C \in \mathcal{S}_{j+1}\} = \{\kappa_C \mid C \in \mathcal{S}_j\} \cup \{(C \vee D)\} \subseteq \mathcal{M}'_j \cup \{\kappa_{(C \vee D)}\} \cup \mathcal{U}_j = \mathcal{M}'_{j+1} \cup \mathcal{U}_j = \mathcal{M}'_{j+1} \cup \mathcal{U}_{j+1}$.
   - By induction hypothesis, both P and Q are conjunctions of elements of $\mathcal{L}^+_0 \cup \mathcal{L}^-_0$ and so is $P \wedge Q$.
   - By induction hypothesis $\{\vec{c} \mid P \to \vec{c}C \in \mathcal{M}'_j\} \subseteq \Gamma_j$. So, $\{\vec{c} \mid P \to \vec{c}C \in \mathcal{M}'_{j+1}\} = \{\vec{c} \mid P \to \vec{c}C \in \mathcal{M}'_j\} \cup \{\vec{c}_3\} \subseteq \Gamma_j \cup \{\vec{c}_3\} = \Gamma_{j+1}$.
   - By Corollary 1, $\Gamma_{j+1} = \Gamma_j \cup \{\vec{c}_1 \downarrow \vec{c}_2\}$ with $\vec{c}_1$ and $\vec{c}_2$ in Γ_j is a pairwise mergeable set of coalition vectors.

2. Assume $\kappa_{(C \vee l)} = (C \vee l) \in \mathcal{U}_j$ and $\kappa_{(D \vee \neg l)} = Q \to \vec{c}_2(D \vee \neg l)$. By induction hypothesis, $\kappa_{(D \vee \neg l)} \in \mathcal{M}'_j$. Using **VRES2** we can derive the clause $\gamma_{j+1} = Q \to \vec{c}_2(C \vee D)$. Let $\mathcal{C}'_{j+1} = (\mathcal{I}, \mathcal{U}_{j+1}, \mathcal{N}'_{j+1})$ with $\mathcal{U}_{j+1} = \mathcal{U}_j$ and $\mathcal{N}'{}_{j+1} = \mathcal{N}'_j \cup \{\gamma_{j+1}\}$, $\mathcal{M}'_{j+1} = \mathcal{M}'_j \cup \{\gamma_{j+1}\}$, $\Gamma_{j+1} = \Gamma_j$, $\kappa_{(C \vee D)} = \gamma_{j+1}$.

   We need to establish that these definitions satisfy the conditions for our construction:

   - By induction hypothesis, $\mathcal{M}'_j \subseteq \mathcal{N}'_j$ and since by our definition $\mathcal{N}'{}_{j+1} = \mathcal{N}'{}_j \cup \{\gamma_{j+1}\}$ and $\mathcal{M}'{}_{j+1} = \mathcal{M}'{}_j \cup \{\gamma_{j+1}\}$, we have $\mathcal{M}'_{j+1} \subseteq \mathcal{N}'_{j+1}$.
   - Also, we have $\mathcal{S}_{j+1} = \mathcal{S}_j \cup \{(C \vee D)\}$ and, by induction hypothesis, $\{\kappa_C \mid C \in \mathcal{S}_j\} \subseteq \mathcal{M}'_j \cup \mathcal{U}_j$. So, $\{\kappa_C \mid C \in \mathcal{S}_{j+1}\} = \{\kappa_C \mid C \in \mathcal{S}_j\} \cup \{(C \vee D)\} \subseteq \mathcal{M}'_j \cup \{\kappa_{(C \vee D)}\} \cup \mathcal{U}_j = \mathcal{M}'_{j+1} \cup \mathcal{U}_j = \mathcal{M}'_{j+1} \cup \mathcal{U}_{j+1}$.
   - By induction hypothesis, Q is a conjunction of elements of $\mathcal{L}^+_0 \cup \mathcal{L}^-_0$.
   - By induction hypothesis $\{\vec{c} \mid P \to \vec{c}C \in \mathcal{M}'_j\} \subseteq \Gamma_j$. So, $\{\vec{c} \mid P \to \vec{c}C \in \mathcal{M}'_{j+1}\} = \{\vec{c} \mid P \to \vec{c}C \in \mathcal{M}'_j\} \subseteq \Gamma_j = \Gamma_{j+1}$.
   - By induction hypthosis, Γ_j is a pairwise mergeable set of coalition vectors and so is $\Gamma_{j+1} = \Gamma_j$.

3. There case where $\kappa_{(C \vee l)} = P \to \vec{c}_1(C \vee l)$ and $\kappa_{(D \vee \neg l)} = (D \vee \neg l) \in \mathcal{U}_j$ can be treated analogously to the previous case.

4. Assume $\kappa_{(C \vee l)} = (C \vee l) \in \mathcal{U}_j$ and $\kappa_{(D \vee \neg l)} = (D \vee \neg l) \in \mathcal{U}_j$. Using **GRES1** we can derive the clause $\gamma_{j+1} = (C \vee D)$. Let $\mathcal{C}'_{j+1} = (\mathcal{I}, \mathcal{U}_{j+1}, \mathcal{N}'_{j+1})$ with $\mathcal{U}_{j+1} = \mathcal{U}_j \cup \{\gamma_{j+1}\}$ and $\mathcal{N}'{}_{j+1} = \mathcal{N}'_j$, $\mathcal{M}'_{j+1} = \mathcal{M}'_j$, $\Gamma_{j+1} = \Gamma_j$, $\kappa_{(C \vee D)} = \gamma_{j+1}$.

   We need to establish that these definitions satisfy the conditions for our construction:

- By induction hypothesis, $\mathcal{M}'_j \subseteq \mathcal{N}'_j$ and since by our definition $\mathcal{N'}_{j+1} = \mathcal{N'}_j$ and $\mathcal{M'}_{j+1} = \mathcal{M'}_j$ we have $\mathcal{M}'_{j+1} \subseteq \mathcal{N}'_{j+1}$.
- Also, we have $\mathcal{S}_{j+1} = \mathcal{S}_j \cup \{(C \vee D)\}$ and, by induction hypothesis, $\{\kappa_C \mid C \in \mathcal{S}_j\} \subseteq \mathcal{M}'_j \cup \mathcal{U}_j$. So, $\{\kappa_C \mid C \in \mathcal{S}_{j+1}\} = \{\kappa_C \mid C \in \mathcal{S}_j\} \cup \{(C \vee D)\} \subseteq \mathcal{M}'_j \cup \mathcal{U}_j \cup \{\kappa_{(C\vee D)}\} = \mathcal{M}'_{j+1} \cup \mathcal{U}_j \cup \{\kappa_{(C\vee D)}\} = \mathcal{M}'_{j+1} \cup \mathcal{U}_{j+1}$.
- By induction hypothesis $\{\vec{c} \mid P \to \vec{c}C \in \mathcal{M}'_j\} \subseteq \Gamma_j$. So, $\{\vec{c} \mid P \to \vec{c}C \in \mathcal{M}'_{j+1}\} = \{\vec{c} \mid P \to \vec{c}C \in \mathcal{M}'_j\} \subseteq \Gamma_j = \Gamma_{j+1}$.
- By induction hypthosis, Γ_j is a pairwise mergeable set of coalition vectors and so is $\Gamma_{j+1} = \Gamma_j$.

Thus, there is a derivation $\mathcal{C}'_0, \ldots, \mathcal{C}'_n$, which uses only the inference rules **GRES1** and **VRES1** such that $\mathcal{C}'_n$ contains a coalition clause of the form $P \to \vec{c}\,\mathbf{false}$. As $P \to \vec{c}\,\mathbf{false} \in \mathcal{M}'_n$, P is a conjunction of elements of $\mathcal{L}_0^+ \cup \mathcal{L}_0^-$, that is, $P = P_1 \wedge \ldots \wedge P_k$ with $P_i \in \mathcal{L}_0^+ \cup \mathcal{L}_0^-$ for each i, $1 \leq i \leq k$.

Let $\mathcal{C}'_{n+1}$ be the coalition problem in $\mathsf{DSNF_{VCL}}$ obtained from $\mathcal{C}'_n$ by adding the result of an application of **RW1** $P \to \vec{c}\,\mathbf{false}$ to $\mathcal{C}'_n$, that is, $\mathcal{C}'_{n+1} = (\mathcal{I}, \mathcal{U}_n \cup \{\neg P\}, \mathcal{N}'_n)$.

We claim that $\neg P \notin \mathcal{U}$. Assume the opposite. From Lemma 24, global clauses are in every state of $\mathcal{T}_+^{\mathcal{C}}$. So, the state Δ contains $\neg P = \neg P_1 \vee \ldots \vee \neg P_k$. As Δ is a minimal downward saturated set, Δ entails $\neg P_i$ for some i, $1 \leq i \leq k$. Δ also contains all formulae in $\mathcal{N}_0^+$, $\mathcal{E}_0^+$ $\mathcal{N}_0^-$, and $\mathcal{E}_0^-$. By construction, either $P_i \to [\mathcal{A}_i]C_i \in \mathcal{N}_0^+$ or $P_i \to \langle\mathcal{A}_i\rangle C_i \in \mathcal{N}_0^-$. By definition of $\mathcal{N}_0^+$ and $\mathcal{N}_0^-$, $\Delta \models P_i$. However, $\Delta \models \neg P_i$ and $\Delta \models P_i$ implies that Δ is inconsistent and should have been removed by an application of rule **E1**. This contradicts our assumption that Δ is in $\mathcal{T}_+^{\mathcal{C}}$. □

Theorem 11 (Completeness of $\mathsf{RES}^{\succ}_{\mathsf{CL}}$)**.** *Let $\mathcal{C} = (\mathcal{I}, \mathcal{U}, \mathcal{N})$ be an unsatisfiable coalition problem in $\mathsf{DSNF_{CL}}$ and let $\mathcal{C}' = (\mathcal{I}, \mathcal{U}, \mathcal{N}')$ be the unsatisfiable coalition problem in $\mathsf{DSNF_{VCL}}$ resulting from exhaustively applying the rules of rewriting system R_4 to $\mathcal{C}$. Then there is a refutation for $\mathcal{C}'$ by $\mathsf{RES}^{\succ}_{\mathsf{CL}}$.*

Proof. First, by Lemma 8, if $\mathcal{C}$ is unsatisfiable then $\mathcal{C}'$ is unsatisfiable. Second, if $\mathcal{C}$ is unsatisfiable and only if $\mathcal{C}$ is unsatisfiable, by Theorem 9, we have that $\mathcal{T}^{\mathcal{C}}$ is closed. In the following we construct $\mathcal{C}' = \mathcal{C}'_0, \ldots, \mathcal{C}'_n$, for some $n \in \mathbb{N}$, of $\mathcal{C}'$ using $\mathsf{RES}^{\succ}_{\mathsf{CL}}$.

Let $\mathcal{C}_{0,0} = \mathcal{C}$. If $\mathcal{T}_+^{\mathcal{C}_{0,0}}$ is closed, then all initial states in $\mathcal{T}_+^{\mathcal{C}_{0,0}}$ have been removed by applications of **E1** which means that $\mathcal{I} \cup \mathcal{U}$ is unsatisfiable. As $\mathcal{C}' = (\mathcal{I}, \mathcal{U}, \mathcal{N}')$, by Lemma 27 there exists a refutation $\mathcal{C}' = \mathcal{C}'_{0,0}, \ldots, \mathcal{C}'_{0,m_0}$ of $\mathcal{C}'_{0,0}$ using only the inference rules **IRES1** and **GRES1** of $\mathsf{RES}^{\succ}_{\mathsf{CL}}$. If $\mathcal{T}_+^{\mathcal{C}_{0,0}}$ is not closed, then by Lemma 28, we can construct a derivation $\mathcal{C}' = \mathcal{C}'_{0,0}, \ldots, \mathcal{C}'_{0,m'_0} = \mathcal{C}'_{1,0} = (\mathcal{I}, \mathcal{U}_1, \mathcal{N}'_1)$ such that there is a global clause γ with $\gamma \in \mathcal{U}_1$ but $\gamma \notin \mathcal{U}$. Let $\mathcal{C}_{1,0} = (\mathcal{I}, \mathcal{U}_1, \mathcal{N})$. We call $\mathcal{C}_{1,0}$ the corresponding coalition problem to $\mathcal{C}'_{1,0}$. As $\mathcal{U} \subset \mathcal{U}_1$, $\mathcal{C}_{1,0}$ is unsatisfiable. Depending on whether $\mathcal{T}^{\mathcal{C}_{1,0}}$ is closed, we proceed as for $\mathcal{C}_{0,0}$ in the construction of the derivation.

We continue this construction until we derive a coalition problem $\mathcal{C}'_{i,0}$ in $\mathsf{DSNF_{VCL}}$ with corresponding coalition problem $\mathcal{C}_{i,0}$ in $\mathsf{DSNF_{CL}}$ for which $\mathcal{T}_+^{\mathcal{C}_{i,0}}$ is closed and for which we can then complete the construction of the refutation using Lemma 27. We know that we will eventually derive such a coalition problem $\mathcal{C}'_{i,0}$ as the number of global clauses is finite, that is, it cannot indefinitely be the case that $\mathcal{T}_+^{\mathcal{C}_{i,0}}$ is open while the final tableau $\mathcal{T}^{\mathcal{C}_{i,0}}$ is closed. On the other hand if we were to derive a coalition problem $\mathcal{C}'_{i,0}$ in $\mathsf{DSNF_{VCL}}$ with corresponding coalition problem $\mathcal{C}_{i,0}$ in $\mathsf{DSNF_{CL}}$ such that both $\mathcal{T}_+^{\mathcal{C}_{i,0}}$ and $\mathcal{T}^{\mathcal{C}_{i,0}}$ are open, then by Theorem 9 $\mathcal{C}_{i,0} = (\mathcal{I}, \mathcal{U}_i, \mathcal{N}_i)$ is satisfiable. As $\mathcal{U} \subseteq \mathcal{U}_i$ and $\mathcal{N} \subseteq \mathcal{N}$ this contradicts the assumption that $\mathcal{C} = (\mathcal{I}, \mathcal{U}, \mathcal{N})$ is unsatisfiable. □

Theorem 12. *Let $\varphi \in \mathsf{WFF_{CL}}$. Let $\mathcal{C}'_0 = (\mathcal{I}, \mathcal{U}, \mathcal{N}')$ be the coalition problem in $\mathsf{DSNF_{VCL}}$ resulting from exhaustively applying the rules of rewriting system R_4 to $\mathcal{C}' = (\{t_\varphi\}, \{t_\varphi \to \tau_0(\varphi)\}, \emptyset)$. Let $\mathcal{C}'_0, \ldots, \mathcal{C}'_n$ be a derivation from $\mathcal{C}'_0$ by $\mathsf{RES}^{\succ}_{\mathsf{CL}}$.*

(a) If $\mathcal{C}'_i$, for some i, $1 \leq i \leq n$, contains a contradiction, then φ is unsatisfiable.

(b) If $\mathcal{C}'_n$ does not contain a contradition and any inference by $\mathsf{RES}^{\succ}_{\mathsf{CL}}$ with premises in $\mathcal{C}'_n$ only derives a clause already in $\mathcal{C}'_n$, then φ is satisfiable.

Proof. Statement (a) follows from Theorems 6 and 7: If $\mathcal{C}'_0$ is satisfiable, then $\mathcal{C}'_i$ is satisfiable. Since $\mathcal{C}'_i$ contains a contradiction, it is obviously not satisfiable and therefore, by Theorem 7, $\mathcal{C}'_0$ is not satisfiable. By Theorem 6, if $\mathcal{C}'_0$ is not satisfiable then φ is not satisfiable.

Regarding statement (b), let us first consider a derivation $\mathcal{C}'_0 = \mathcal{C}''_0, \ldots, \mathcal{C}''_m$ such that any inference by $\mathsf{RES}^{\succ}_{\mathsf{CL}}$ with premises in $\mathcal{C}''_m$ only derives a clause already in $\mathcal{C}''_m$. Then $\mathcal{C}''_m = \mathcal{C}'_n$, there is one unique coalition problem with this closure property derivable from $\mathcal{C}'_0$. Assume the opposite, that is, $\mathcal{C}''_m \neq \mathcal{C}'_n$. Without loss of generality assume that $\mathcal{C}''_m$ contains a clause γ that is not in $\mathcal{C}'_n$. Clearly, γ has been derived by a sequence of inference steps from claues in $\mathcal{C}''_0$. However, since $\mathcal{C}''_0 = \mathcal{C}'_0$ these clauses are also present in $\mathcal{C}'_0$ and γ is derivable from $\mathcal{C}'_0$. So, γ must be present in $\mathcal{C}'_n$.

Statement (b) then follows from Theorems 6 and 11: Since any inference by $\mathsf{RES}^{\succ}_{\mathsf{CL}}$ with premises in $\mathcal{C}'_n$ only derives a clause already in $\mathcal{C}'_n$ and $\mathcal{C}'_n$ is unique and contains no contradiction, there is no refutation of $\mathcal{C}'_n$. Therefore, there is also no refutation of $\mathcal{C}'_0$ which by Theorem 11 implies that $\mathcal{C}'_0$ is satisfiable. By Theorem 6, if $\mathcal{C}'_0$ is satisfiable then φ is satisfiable. ☐

5 Conclusion

We have described a calculus $\mathsf{RES}^{\succ}_{\mathsf{CL}}$ based on ordered resolution for Coalition Logic and sketched proofs of its soundness and completeness. We have also shown that any derivation by $\mathsf{RES}^{\succ}_{\mathsf{CL}}$ terminates. The prover CLProver++ provides an implementation of $\mathsf{RES}^{\succ}_{\mathsf{CL}}$. Our evaluation of CLProver++ indicates that the ordering refinement improves performance by several orders of magnitude compared to unrefined resolution as implemented in CLProver. Our evaluation also shows that similar improvements can be gained by optimising the normal form transformation that is used to obtain coalition problems from CL formulae.

Our work on Coalition Logic is a first step towards the development of resolution calculi for more expressive logics for reasoning about the strategic abilities of coalitions of agents. A wide variety of such logics can be found in the literature, starting with Alternating-Time Temporal Logic ATL. The notion of coalition vectors that we have introduced in this paper are closely related to the notion of k-actions in Coalition Action Logic [5] and to the notion of commitment functions in ATLES [29]. We believe that the combination of the techniques developed in this paper with the techniques for temporal logics with eventualities provide a good basis for the development of effective calculi for logics such as ATL, Coalition Action Logic and ATLES.

Acknowledgments

The second author was supported by the EPSRC vacation bursary scheme.

References

[1] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. In *Proc. FOCS '97*, pages 100–109. IEEE, 1997.

[2] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. In *Revised Lectures of COMPOS '97*, volume 1536 of *LNCS*, pages 23–60. Springer, 1998.

[3] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *J. ACM*, 49(5):672–713, 2002.

[4] L. Bachmair and H. Ganzinger. Resolution theorem proving. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, pages 19–99. Elsevier, 2001.

[5] S. Borgo. Coalitions in action logic. In *Proc. IJCAI '07*, pages 1822–1827, 2007.

[6] S. Cerrito, A. David, and V. Goranko. Optimal tableaux-based decision procedure for testing satisfiability in the alternating-time temporal logic ATL+. In *Proc. IJCAR 2014*, volume 8562 of *LNCS*, pages 277–291. Springer, 2014.

[7] A. David. TATL: Implementation of ATL tableau-based decision procedure. In *Proc. TABLEAUX 2013*, volume 8123 of *LNCS*, pages 97–103. Springer, 2013.

[8] V. Goranko. Coalition games and alternating temporal logics. In *Proc. TARK '01*, pages 259–272. Morgan Kaufmann, 2001.

[9] V. Goranko and D. Shkatov. Tableau-based decision procedures for logics of strategic ability in multiagent systems. *ACM Trans. Comput. Log.*, 11(1):1–51, 2009.

[10] V. Goranko and G. van Drimmelen. Complete axiomatization and decidability of alternating-time temporal logic. *Theor. Comput. Sci.*, 353(1):93–117, 2006.

[11] H. H. Hansen. Tableau games for coalition logic and alternating-time temporal logic – theory and implementation. Master's thesis, University of Amsterdam, The Netherlands, Oct. 2004.

[12] I. Horrocks, U. Hustadt, U. Sattler, and R. A. Schmidt. Computational modal logic. In P. Blackburn, J. van Benthem, and F. Wolter, editors, *Handbook of Modal Logic*, pages 181–245. Elsevier, 2006.

[13] U. Hustadt and B. Konev. TRP++: A temporal resolution prover. In M. Baaz, J. Makowsky, and A. Voronkov, editors, *Collegium Logicum*, pages 65–79. Kurt Gödel Society, 2004.

[14] L. Kovács and A. Voronkov. First-order theorem proving and Vampire. In *Proc. CAV 2013*, volume 8044 of *LNCS*, pages 1–35. Springer, 2013.

[15] R. Kowalski and P. J. Hayes. Semantic trees in automatic theorem-proving. In B. Meltzer and D. Mitchie, editors, *Machine Intelligence 4*, pages 87–101. Edinburgh University Press, UK, 1969.

[16] R. C. T. Lee. *A completeness theorem and a computer program for finding theorems derivable from given axioms.* PhD thesis, University of California, Berkeley, USA, 1967.

[17] C. Nalon, L. Zhang, C. Dixon, and U. Hustadt. A resolution-based calculus for coalition logic. *J. Logic Comput.*, 24(4):883–917, 2014.

[18] C. Nalon, L. Zhang, C. Dixon, and U. Hustadt. A resolution prover for coalition logic. In *Proc. SR2014*, volume 146 of *Electron. Proc. Theor. Comput. Sci.*, pages 65–73, 2014.

[19] M. Pauly. *Logic for Social Software.* PhD thesis, University of Amsterdam, The Netherlands, 2001.

[20] M. Pauly. A modal logic for coalitional power in games. *Journal of Logic and Computation*, 12(1):149–166, 2002.

[21] D. A. Plaisted and S. Greenbaum. A structure-preserving clause form translation. *J. Symb. Comput.*, 2:293–304, 1986.

[22] J. A. Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM*, 12:23–41, 1965.

[23] S. Schulz. The E theorem prover, 2013. http://wwwlehre.dhbw-stuttgart.de/~sschulz/E/E.html.

[24] S. Schulz. System description: E 1.8. In *Proc. LPAR-19*, volume 8312 of *LNCS*, pages 735–743. Springer, 2013.

[25] The SPASS Team. Automation of logic: Spass, 2010. http://www.spass-prover.org/.

[26] G. van Drimmelen. Satisfiability in alternating-time temporal logic. In *Proc. LICS '03*, pages 208–207. IEEE, 2003.

[27] A. Voronkov. Vampire. http://www.vprover.org/index.cgi.

[28] D. Walther, C. Lutz, F. Wolter, and M. Wooldridge. ATL satisfiability is indeed ExpTime-complete. *J. Logic Comput.*, 16(6):765–787, 2006.

[29] D. Walther, W. van der Hoek, and M. Wooldridge. Alternating-time temporal logic with explicit strategies. In *Proc. TARK '07*, pages 269–278. ACM, 2007.

[30] C. Weidenbach, D. Dimova, A. Fietzke, R. Kumar, M. Suda, and P. Wischnewski. SPASS version 3.5. In R. A. Schmidt, editor, *Proc. CADE-22*, volume 5663 of *LNCS*, pages 140–145. Springer, 2009.

[31] P. Wolper. The tableau method for temporal logic: An overview. *Logique et Analyse*, 28:119–136, 1985.

[32] L. Zhang, U. Hustadt, and C. Dixon. A refined resolution calculus for CTL. In *Proc. CADE-22*, volume 5663 of *LNCS*, pages 245–260. Springer, 2009.

[33] L. Zhang, U. Hustadt, and C. Dixon. A resolution calculus for branching-time temporal logic CTL. *ACM Trans. Comput. Log.*, 15(1):10:1–38, 2014.