A Resolution Calculus for Coalition Logic

Cláudia Nalon
Departament of Computer Science
University of Brasília (Brazil)
nalon@unb.br

Lan Zhang
Information School
Capital University of Economics and Business (China)
lan@cueb.edu.cn

Clare Dixon Ullrich Hustadt
Department of Computer Science
University of Liverpool (UK)
{CLDixon,U.Hustadt}@liverpool.ac.uk

We report on a recent work on a resolution-based calculus for Coalition Logics, a non-normal modal logic that can be used for reasoning about cooperative agency. The resolution method works on problems, which are essentially sets of clauses where the different contexts for reasoning are separated. We introduce a normal form and a set of inference rules to solve the satisfiability problem in Coalition Logics. The calculus is sound, complete, and terminating.

1 Introduction

Coalition Logic CL is a formalism intended to describe the ability of groups of agents to achieve an outcome in a strategic game [13]. CL is a multi-modal logic with modal operators of the form $[\mathscr{A}]$, where $\mathscr{A}$ is a set of agents. The formula $[\mathscr{A}]\varphi$ reads as *the coalition $\mathscr{A}$ has a strategy to achieve φ*, where φ is a formula. We note that CL is a non-normal modal logic, as the schema that represents *additivity*, $[\mathscr{A}]\varphi \wedge [\mathscr{A}]\psi \Rightarrow [\mathscr{A}](\varphi \wedge \psi)$, is not valid. However, *monotonicity*, $[\mathscr{A}](\varphi \wedge \psi) \Rightarrow [\mathscr{A}]\varphi \wedge [\mathscr{A}]\psi$, holds.

Coalition Logic is equivalent to the next-time fragment of *Alternating-Time Temporal Logic* (ATL) [1, 4], where $[\mathscr{A}]\varphi$ translates into $\langle\langle\mathscr{A}\rangle\rangle\bigcirc\varphi$ (read as *the coalition $\mathscr{A}$ can ensure φ at the next moment in time*). The satisfiability problems for ATL and CL are EXPTIME-complete [15] and PSPACE-complete [13], respectively. Proof methods for these logics include, for instance, tableau-based methods for ATL [15, 5] and a tableau-based method for CL [7].

In this paper, we report on a recently developed resolution-based calculus for CL, $\mathsf{RES}_{\mathsf{CL}}$ [11]. As to the best of our knowledge, there are no other resolution-based methods for either ATL or CL. Providing such a method for CL gives the user a choice of proof methods. Several comparisons of tableau algorithms and resolution methods [9, 6] indicate that there is no overall best approach: for some classes of formulae tableau algorithms perform better whilst on others resolution performs better. So, with a choice of different provers, for the best result the user could run several in parallel or the one most likely to succeed depending on the type of the input formulae. $\mathsf{RES}_{\mathsf{CL}}$ is sound, complete, and terminating as shown in [11].

The paper is organised as follows. In the next section, we present the syntax, axiomatisation, and semantics of CL. In Section 3, we introduce the resolution-based method for CL, the main results, and provide a small example. Conclusions and future work are given in Section 4.

Submitted to:
SR 2014

2 Coalition Logic

As in [5], we define $\Sigma \subset \mathbb{N}$ to be a finite, non-empty set of agents. A **coalition** $\mathscr{A}$ is a subset of Σ. Formulae in CL are constructed from propositional symbols ($\Pi = \{p, q, r, \ldots, p_1, q_1, r_1, \ldots\}$) and constants (**true**, **false**), together with Boolean operators ($\neg$, for negation, and $\wedge$, for conjunction) and coalition modalities. Formulae whose main operator is classical are built in the usual way. A **coalition modality** is either of the form $[\mathscr{A}]\varphi$ or $\langle\mathscr{A}\rangle\varphi$, where φ is a well-formed CL formula. The coalition operator $\langle\mathscr{A}\rangle$ is the dual of $[\mathscr{A}]$, that is, $\langle\mathscr{A}\rangle\varphi$ is an abbreviation for $\neg[\mathscr{A}]\neg\varphi$, for every coalition $\mathscr{A}$ and formula φ. We denote by $\mathsf{WFF}_{\mathsf{CL}}$ the set of CL well-formed formulae. Parentheses will be omitted if the reading is not ambiguous. We also omit the curly brackets within modalities. For instance, we write $[1,2]\varphi$ instead of $[\{1,2\}]\varphi$. Formulae of the form $\bigvee \varphi_i$ (resp. $\bigwedge \varphi_i$), $1 \leq i \leq n$, $n \in \mathbb{N}$, $\varphi_i \in \mathsf{WFF}_{\mathsf{CL}}$, represent arbitrary disjunctions (resp. conjunctions) of formulae. If $n = 0$, $\bigvee \varphi_i$ (resp. $\bigwedge \varphi_i$) is called the **empty disjunction** (resp. **empty conjunction**), denoted by **false** (resp. **true**).

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**. A **positive coalition formula** (resp. **negative coalition formula**) is a formula of the form $[\mathscr{A}]\varphi$ (resp. $\langle\mathscr{A}\rangle\varphi$), where $\varphi \in \mathsf{WFF}_{\mathsf{CL}}$. A **coalition formula** is either a positive or a negative coalition formula.

Coalition logic can be axiomatised by the following schemata (where $\mathscr{A}, \mathscr{A}'$ are coalitions and φ, φ_1, φ_2 are well-formed formulae) [13]:

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

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

Lemma 1 *The formula* $[\mathscr{A}]\psi_1 \wedge \langle\mathscr{B}\rangle\psi_2 \Rightarrow \langle\mathscr{B} \setminus \mathscr{A}\rangle(\psi_1 \wedge \psi_2)$ *where* $\mathscr{A}$ *and* $\mathscr{B}$ *are coalitions,* $\mathscr{A} \subseteq \mathscr{B}$*, and* $\psi_1, \psi_2 \in \mathsf{WFF}_{\mathsf{CL}}$*, is valid.*

Proof.

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

□

The semantics of CL is given in terms of *Concurrent Game Structures* (CGS) [2] and it is *positional*, that is, agents have no memory of their past decisions and, thus, those decisions are made by taking into account only the current state. We note that the semantics of CL is often presented in terms of *Multiplayer Game Models* (MGMs) [12]. Note also that MGMs yield the same set of validities as CGSs [4]. As we intend to extend the proof method given here, the correctness proofs are based on the tableau procedure for full ATL [5] and we follow the semantics presentation given there.

Def. 1 A **Concurrent Game Frame** (CGF) is a tuple $\mathscr{F} = (\Sigma, \mathscr{S}, s_0, d, \delta)$, where

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

In the following, let $\mathscr{F} = (\Sigma, \mathscr{S}, s_0, d, \delta)$ be a CGF with $s, s' \in \mathscr{S}$. We say that s' is a **successor** of s (an s-successor) if $s' = \delta(s,\sigma)$, for some $\sigma \in D(s)$. If κ is a tuple, then κ_n (or $\kappa(n)$) denotes the n-th element of κ. Let $|\Sigma| = k$ and let $\mathscr{A} \subseteq \Sigma$ be a coalition. An $\mathscr{A}$**-move** $\sigma_{\mathscr{A}}$ at $s \in \mathscr{S}$ is a k-tuple such that $\sigma_{\mathscr{A}}(a) \in D(a,s)$ for every $a \in \mathscr{A}$ and $\sigma_{\mathscr{A}}(a') = *$ (i.e. an arbitrary move) for every $a' \notin \mathscr{A}$. We denote by $D(\mathscr{A},s)$ the set of all $\mathscr{A}$-moves at state s. A move vector σ **extends** an $\mathscr{A}$-move vector $\sigma_{\mathscr{A}}$, denoted by $\sigma_{\mathscr{A}} \sqsubseteq \sigma$ or $\sigma \sqsupseteq \sigma_{\mathscr{A}}$, if $\sigma(a) = \sigma_{\mathscr{A}}(a)$ for every $a \in \mathscr{A}$. Let $\sigma_{\mathscr{A}} \in D(\mathscr{A},s)$ be an $\mathscr{A}$-move. The **outcome** of $\sigma_{\mathscr{A}}$ at s, denoted by $out(s,\sigma_{\mathscr{A}})$, is the set of all states $s' \in \mathscr{S}$ for which there exists a move vector $\sigma \in D(s)$ such that $\sigma_{\mathscr{A}} \sqsubseteq \sigma$ and $\delta(s,\sigma) = s'$.

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

Def. 3 Let $\mathscr{M} = (\Sigma, \mathscr{S}, s_0, d, \delta, \Pi, \pi)$ be a CGM with $s \in \mathscr{S}$. The satisfaction relation, denoted by $\models$, is inductively defined as follows.

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

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

In this work, we consider *tight satisfiability*, i.e. the evaluation of a formula φ depends only on the agents occurring in φ [15]. We denote by Σ_φ, where $\Sigma_\varphi \subseteq \Sigma$, the set of agents occurring in a well-formed formula φ. If Φ is a set of well-formed formulae, $\Sigma_\Phi \subseteq \Sigma$ denotes $\bigcup_{\varphi\in\Phi} \Sigma_\varphi$. Let $\varphi \in \mathsf{WFF}_{\mathsf{CL}}$ and $\mathscr{M} = (\Sigma_\varphi, \mathscr{S}, s_0, d, \delta, \Pi, \pi)$ be a CGM. Formulae are interpreted with respect to the distinguished world s_0. Thus, a formula φ is said to be **satisfiable in** $\mathscr{M}$, denoted by $\mathscr{M} \models \varphi$, if $\langle \mathscr{M}, s_0 \rangle \models \varphi$; it is said to be **satisfiable** if there is a model $\mathscr{M}$ such that $\langle \mathscr{M}, s_0 \rangle \models \varphi$; and it is said to be **valid** if for all models $\mathscr{M}$ we have $\langle \mathscr{M}, s_0 \rangle \models \varphi$. A finite set $\Gamma \subset \mathsf{WFF}_{\mathsf{CL}}$ is **satisfiable in a state** s **in** $\mathscr{M}$, denoted by $\langle \mathscr{M}, s \rangle \models \Gamma$, if for all $\gamma_i \in \Gamma$, $0 \leq i \leq n, n \in \mathbb{N}$, $\langle \mathscr{M}, s \rangle \models \gamma_i$; Γ is **satisfiable in a model** $\mathscr{M}$, $\mathscr{M} \models \Gamma$, if $\langle \mathscr{M}, s_0 \rangle \models \Gamma$; and Γ is **satisfiable**, if there is a model $\mathscr{M}$ such that $\mathscr{M} \models \Gamma$.

3 Resolution Calculus

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

Def. 4 A **coalition problem** is a tuple $(\mathscr{I}, \mathscr{U}, \mathscr{N})$, where $\mathscr{I}$, the set of initial formulae, is a finite set of propositional formulae; $\mathscr{U}$, the set of global formulae, is a finite set of formulae in $\mathsf{WFF}_{\mathsf{CL}}$; and $\mathscr{N}$, the set of coalition formulae, is a finite set of coalition formulae, i.e. those formulae in which a coalition modality occurs.

The semantics of coalition problems assumes that initial formulae hold at the initial state; and that global and coalition formulae hold at every state of a model.

Def. 5 Given a coalition problem $\mathscr{C} = (\mathscr{I}, \mathscr{U}, \mathscr{N})$, we denote by $\Sigma_{\mathscr{C}}$ the set of agents $\Sigma_{\mathscr{U} \cup \mathscr{N}}$. If $\mathscr{C} = (\mathscr{I}, \mathscr{U}, \mathscr{N})$ is a coalition problem and $\mathscr{M} = (\Sigma_{\mathscr{C}}, \mathscr{S}, s_0, d, \delta, \Pi, \pi)$ is a CGM, then $\mathscr{M} \models \mathscr{C}$ if, and only if, $\langle \mathscr{M}, s_0 \rangle \models \mathscr{I}$ and $\langle \mathscr{M}, s \rangle \models \mathscr{U} \cup \mathscr{N}$, for all $s \in \mathscr{S}$. We say that $\mathscr{C} = (\mathscr{I}, \mathscr{U}, \mathscr{N})$ is **satisfiable**, if there is a model $\mathscr{M}$ such that $\mathscr{M} \models \mathscr{C}$.

In order to apply the resolution method, we further require that formulae within each of those sets are in *clausal form*: **initial clauses** and **global clauses** are of the form $\bigvee_{j=1}^{n} l_j$; **positive coalition clauses** are of the form $\bigwedge_{i=1}^{m} l'_i \Rightarrow [\mathscr{A}] \bigvee_{j=1}^{n} l_j$; and **negative coalition clauses** are of the form $\bigwedge_{i=1}^{m} l'_i \Rightarrow \langle \mathscr{A} \rangle \bigvee_{j=1}^{n} l_j$; where $m, n \geq 0$ and l'_i, l_j, for all $1 \leq i \leq m$, $1 \leq j \leq n$, are literals or constants. We assume that clauses are kept in the simplest form by means of usual Boolean simplification rules. Tautologies are removed from the set of clauses as they cannot contribute to finding a contradiction. A **coalition problem in** $\mathsf{DSNF}_{\mathsf{CL}}$ is a coalition problem $(\mathscr{I}, \mathscr{U}, \mathscr{N})$ such that $\mathscr{I}$ is a set of initial clauses, $\mathscr{U}$ is a set of global clauses, and $\mathscr{N}$ is a set of positive and negative coalition clauses.

The transformation of a coalition logic formula into a coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$ is analogous to the approach taken in [3], where first-order temporal formulae are transformed into a *Divided Separated Normal Form* (DSNF) by means of renaming and rewriting of temporal operators by simulating their fix-point representation. The transformation of a formula into a **coalition problem in** $\mathsf{DSNF}_{\mathsf{CL}}$, which are given in [10, 11], reduces the number of operators and separates the contexts to which the resolution inference rules are applied.

The set of inference rules for $\mathsf{RES}_{\mathsf{CL}}$ are given as follows. Let $(\mathscr{I}, \mathscr{U}, \mathscr{N})$ be a coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$; C, C' be conjunctions of literals; D, D' be disjunctions of literals; l, l_i be literals; and $\mathscr{A}, \mathscr{B} \subseteq \Sigma$ be coalitions (where Σ is the set of all agents). The first rule, **IRES1**, is classical resolution applied to clauses which are true at the initial state. The next inference rule, **GRES1**, performs resolution on clauses which are true in all states.

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

Soundness of **IRES1** and **GRES1** follow from the semantics of coalition problems and the soundness result for classical propositional resolution [14]. The following rules perform resolution on positive and negative coalition clauses.

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

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

Soundness of the inference rules **CRES1-4** follow from the axiomatisation of CL, given in Section 2. We give sketches of the proofs here. Let $\mathscr{M}$ be a CGM and $s\in\mathscr{M}$ a state. Recall that coalition clauses are satisfied at any state in $\mathscr{M}$. For **CRES1**, if $\langle\mathscr{M},s\rangle\models C\wedge C'$, by the semantics of conjunction and implication, we have that $\langle\mathscr{M},s\rangle\models C\wedge C'\Rightarrow[\mathscr{A}](D\vee l)\wedge[\mathscr{B}](D'\vee\neg l)$. By axiom **S**, we have that $[\mathscr{A}](D\vee l)\wedge[\mathscr{B}](D'\vee\neg l)$ implies $[\mathscr{A}\cup\mathscr{B}]((D\vee l)\wedge(D'\vee\neg l))$. Therefore, $\langle\mathscr{M},s\rangle\models C\wedge C'\Rightarrow[\mathscr{A}\cup\mathscr{B}]((D\vee l)\wedge(D'\vee\neg l))$. By classical resolution applied within the successor states, we obtain that $\langle\mathscr{M},s\rangle\models C\wedge C'\Rightarrow[\mathscr{A}\cup\mathscr{B}](D\vee D')$. For **CRES3**, by Lemma 1, we have that $[\mathscr{A}](D\vee l)\wedge\langle\mathscr{B}\rangle(D'\vee\neg l)\Rightarrow\langle\mathscr{B}\setminus\mathscr{A}\rangle((D\vee l)\wedge(D'\vee\neg l))$, with $\mathscr{A}\subseteq\mathscr{B}$, is valid. If $\langle\mathscr{M},s\rangle\models C\wedge C'$, by the semantics of implication, we have that $\langle\mathscr{M},s\rangle\models C\wedge C'\Rightarrow\langle\mathscr{B}\setminus\mathscr{A}\rangle((D\vee l)\wedge(D'\vee\neg l))$. Applying classical resolution within the successor states, we obtain that $\langle\mathscr{M},s\rangle\models C\wedge C'\Rightarrow\langle\mathscr{B}\setminus\mathscr{A}\rangle(D\vee D')$. Soundness of the inference rules **CRES2** and **CRES4** follow from the above and the semantics of coalition problems: as a formula φ in $\mathscr{U}$ is satisfied at all states, we have that **true** $\Rightarrow[\emptyset]\varphi$ is also satisfied at all states.

The next two inference rules are justified by the axioms $\bot$ and $\top$, given by $\neg[\mathscr{A}]$**false** and $[\mathscr{A}]$**true**, respectively, which imply that the consequent in both rewriting rules cannot be satisfied.

$$\textbf{RW1}\quad \begin{array}{cl} \bigwedge_{i=1}^{n} l_i\Rightarrow[\mathscr{A}]\textbf{false} & \in\mathscr{N}\\ \hline \bigvee_{i=1}^{n}\neg l_i & \in\mathscr{U} \end{array} \qquad \textbf{RW2}\quad \begin{array}{cl} \bigwedge_{i=1}^{n} l_i\Rightarrow\langle\mathscr{A}\rangle\textbf{false} & \in\mathscr{N}\\ \hline \bigvee_{i=1}^{n}\neg l_i & \in\mathscr{U} \end{array}$$

As sketched above, the resolution-based calculus for Coalition Logic is sound.

Theorem 1 (Soundness) *Let $\mathscr{C}$ be a coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$. Let $\mathscr{C}'$ be the coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$ obtained from $\mathscr{C}$ by applying any of the inference rules* **IRES1**, **GRES1**, **CRES1-4** *or* **RW1-2** *to $\mathscr{C}$. If $\mathscr{C}$ is satisfiable, then $\mathscr{C}'$ is satisfiable.*

A **derivation** from a coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$ $\mathscr{C}=(\mathscr{I},\mathscr{U},\mathscr{N})$ by $\mathsf{RES}_{\mathsf{CL}}$ is a sequence $\mathscr{C}_0,\mathscr{C}_1,\mathscr{C}_2,\ldots$ of problems such that $\mathscr{C}_0=\mathscr{C}$, $\mathscr{C}_i=(\mathscr{I}_i,\mathscr{U}_i,\mathscr{N}_i)$, and $\mathscr{C}_{i+1}$ is either $(\mathscr{I}_i\cup\{D\},\mathscr{U}_i,\mathscr{N}_i)$, where D is the conclusion of **IRES1**; $(\mathscr{I}_i,\mathscr{U}_i\cup\{D\},\mathscr{N}_i)$, where D is the conclusion of **GRES1**, **RW1**, or **RW2**; or $(\mathscr{I}_i,\mathscr{U}_i,\mathscr{N}_i\cup\{D\})$, where D is the conclusion of **CRES1**, **CRES2**, **CRES3**, or **CRES4**; and D is not a tautology.

A **refutation** for a coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$ $\mathscr{C}=(\mathscr{I},\mathscr{U},\mathscr{N})$ (by $\mathsf{RES}_{\mathsf{CL}}$) is a derivation from $\mathscr{C}$ such that for some $i\geq 0$, $\mathscr{C}_i=(\mathscr{I}_i,\mathscr{U}_i,\mathscr{N}_i)$ contains a contradiction, where a contradiction is given by either **false** $\in\mathscr{I}_i$ or **false** $\in\mathscr{U}_i$. A derivation *terminates* if, and only if, either a contradiction is derived or no new clauses can be derived by further application of resolution rules of $\mathsf{RES}_{\mathsf{CL}}$.

The completeness proof for $\mathsf{RES}_{\mathsf{CL}}$ is based on the tableau construction given in [5]. Given an unsatisfiable coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$ $\mathscr{C}$, an initial tableau is obtained by this construction which is then reduced to an empty tableau via a sequence of deletion steps. We show that each deletion step corresponds to an application of the resolution inference rules to (sub)sets of clauses in $\mathscr{C}$ or clauses previously derived from $\mathscr{C}$. The derivation constructed in this way is shown to be a refutation of $\mathscr{C}$.

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

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

Theorem 3 *Let $\mathscr{C} = (\mathscr{I}, \mathscr{U}, \mathscr{N})$ be a coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$. Then any derivation from $\mathscr{C}$ by $\mathsf{RES}_{\mathsf{CL}}$ terminates.*

Example 1 *We show a simple example, adapted from [8], of the application of $\mathsf{RES}_{\mathsf{CL}}$ to a problem involving the cooperation of agents. There are two agents (1 and 2) and two toggle switches. For each agent $a = 1,2$, there are two possible actions: $[a]tog_a \wedge [a]\neg tog_a$, where tog_a denotes that the agent a can toggle the switch (clauses 3, 9–13). The light is initially off, i.e. we have that $t_0 \Rightarrow \neg l$ (clauses 1 and 2). If the light is off and the switch is toggled, then at the next moment the light is on: $tog_a \wedge \neg l \Rightarrow [a]l$ (clauses 5 and 6). Similarly, if the light is on and the agent toggles the switch, then at the next moment the light is off: $tog_a \wedge l \Rightarrow [a]\neg l$ (clauses 7 and 8). We prove that the agents can cooperate to turn on the light, that is, we introduce the clauses 4 and 14, which corresponds to the negation of $[1,2]l$.*

1.			t_0	$[\mathscr{I}]$	*14.*	t_4	$\Rightarrow$	$[\emptyset]\neg l$	$[\mathscr{N}]$
2.			$\neg t_0 \vee \neg l$	$[\mathscr{U}]$	*15.*			$\neg t_0 \vee t_4$	$[\mathscr{U}$,**gres1**,*3, 4*$]$
3.			$\neg t_0 \vee t_1$	$[\mathscr{U}]$	*16.*	$t_4 \wedge tog_1 \wedge \neg l$	$\Rightarrow$	$[1]$**false**	$[\mathscr{N}$,**cres1**,*5, 14*$]$
4.			$\neg t_1 \vee t_4$	$[\mathscr{U}]$	*17.*	t_1	$\Rightarrow$	$[\emptyset]t_4$	$[\mathscr{N}$,**cres2**,*13, 4*$]$
5.	$tog_1 \wedge \neg l$	$\Rightarrow$	$[1]l$	$[\mathscr{N}]$	*18.*			$l \vee \neg t_4 \vee \neg tog_1$	$[\mathscr{U}$,**rw1**,*16*$]$
6.	$tog_2 \wedge \neg l$	$\Rightarrow$	$[2]l$	$[\mathscr{N}]$	*19.*	t_1	$\Rightarrow$	$[\emptyset]l \vee \neg tog_1$	$[\mathscr{N}$,**cres2**,*17, 18*$]$
7.	$tog_1 \wedge l$	$\Rightarrow$	$[1]\neg l$	$[\mathscr{N}]$	*20.*	t_1	$\Rightarrow$	$[1]l$	$[\mathscr{N}$,**cres1**,*19, 9*$]$
8.	$tog_2 \wedge l$	$\Rightarrow$	$[2]\neg l$	$[\mathscr{N}]$	*21.*	$t_1 \wedge t_4$	$\Rightarrow$	$[1]$**false**	$[\mathscr{N}$,**cres1**,*20, 14*$]$
9.	t_1	$\Rightarrow$	$[1]tog_1$	$[\mathscr{N}]$	*22.*			$\neg t_1 \vee \neg t_4$	$[\mathscr{U}$,**rw1**,*21*$]$
10.	t_1	$\Rightarrow$	$[2]tog_2$	$[\mathscr{N}]$	*23.*			$\neg t_0 \vee \neg t_1$	$[\mathscr{U}$,**gres1**,*22, 15*$]$
11.	t_1	$\Rightarrow$	$[1]\neg tog_1$	$[\mathscr{N}]$	*24.*			$\neg t_0$	$[\mathscr{U}$,**gres1**,*23, 3*$]$
12.	t_1	$\Rightarrow$	$[2]\neg tog_2$	$[\mathscr{N}]$	*25.*			**false**	$[\mathscr{I}$,**ires1**,*1, 24*$]$
13.	t_1	$\Rightarrow$	$[\emptyset]t_1$	$[\mathscr{N}]$					

4 Conclusion

The resolution-based calculus for the Coalition Logic CL is applied to a coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$, which separates the dimensions to which the resolution rules are applied. The transformation into the normal form is satisfiability preserving and polynomially bounded by the size of the original formula. Soundness of the method follows from the axiomatisation of CL. Completeness is proved with respect to the tableau procedure given in [5]: if a tableau for a coalition problem is closed, there is a refutation based on the calculus given here. Termination is ensured by the fact that number of propositional symbols and agents is finite, so there are only a finite number of clauses that can be generated.

The decision procedure based on $\mathsf{RES}_{\mathsf{CL}}$ is in EXPTIME, as shown in [11]. This is optimal, as the satisfiability problem for coalition problems in $\mathsf{DSNF}_{\mathsf{CL}}$ is EXPTIME-hard, thus more expressive than the language of CL. This result follows from [15, Lemma 4.10, page 785] and the fact that an extension of CL with positive occurrences of ATL's $\langle\langle\emptyset\rangle\rangle\,\square$ operator can be translated into $\mathsf{DSNF_{CL}}$. It also follows that $\mathsf{DSNF_{CL}}$ is more expressive than CL.

$\mathsf{RES_{CL}}$ is very simple in structure, so an implementation can be obtained in a quite straightforward way by extending existing resolution provers for either PTL or CTL, for instance, and it is left as future work. Future work also includes the extension of this calculus to the full language of ATL, which can be achieved by designing a set of resolution-like inference rules to deal with eventualities, that is, formulae which hold at some future time of a run.

References

[1] R. Alur, T. A. Henziger & O. Kupferman (1997): *Alternating-Time Temporal Logic*. In: *Proceedings of the 38th IEEE Symposium on Foundations of Computer Science*, pp. 100–109, doi:10.1109/SFCS.1997.646098.

[2] R. Alur, T. A. Henziger & O. Kupferman (2002): *Alternating-Time Temporal Logic*. *Journal of the ACM* 49(5), pp. 672–713, doi:10.1145/585265.585270.

[3] A. Degtyarev, M. Fisher & B. Konev (2006): *Monodic temporal resolution*. *ACM Trans. Comput. Log* 7(1), pp. 108–150, doi:10.1145/1119439.1119443.

[4] V. Goranko (2001): *Coalition games and alternating temporal logics*. In: *TARK '01*, Morgan Kaufmann, San Francisco, CA, USA, pp. 259–272. Available at `http://dl.acm.org/citation.cfm?id=1028128.1028157`.

[5] V. Goranko & D. Shkatov (2009): *Tableau-Based Decision Procedures for Logics of Strategic Ability in Multiagent Systems*. *ACM Transactions on Computational Logic* 11(1), pp. 3:1–3:51, doi:10.1145/1614431.1614434.

[6] R. Goré, J. Thomson & F. Widmann (2011): *An Experimental Comparison of Theorem Provers for CTL*. In C. Combi, M. Leucker & F.. Wolter, editors: *TIME 2011,Germany, September 12-14*, IEEE, pp. 49–56, doi:10.1109/TIME.2011.16.

[7] H. Hansen (2004): *Tableau Games for Coalition Logic and Alternating-Time Temporal Logic – theory and implementation*. Master's thesis, University of Amsterdam.

[8] A. Herzig (2007): *Logics for Agency and Multi-Agent Systems*. Slides. ESSLLI. Available at `http://www.staff.science.uu.nl/~broer110/ESSLLI07/`.

[9] U. Hustadt & R. A. Schmidt (2002): *Scientific Benchmarking with Temporal Logic Decision Procedures*. In D. Fensel, F. Giunchiglia, D. McGuinness & M-A. Williams, editors: *KR'2002*, M. Kaufmann, pp. 533–544.

[10] C. Nalon, L. Zhang, C. Dixon & U. Hustadt (2013): *A resolution-based calculus for Coalition Logic (Extended Version)*. Technical Report ULCS-13-004, University of Liverpool, Liverpool, UK. Available at http://intranet.csc.liv.ac.uk/research/techreports/?id=ULCS-13-004.

[11] C. Nalon, L. Zhang, C. Dixon & U. Hustadt (2014): *A resolution-based calculus for Coalition Logic*. *Journal of Logic and Computation*, doi:10.1093/logcom/ext074. To appear.

[12] M. Pauly (2001): *Logic for Social Software*. Ph.D. thesis, University of Amsterdam. Dissertation Series 2001-10.

[13] M. Pauly (2002): *A Modal Logic for Coalitional Power in Games*. *Journal of Logic and Computation* 12(1), pp. 149–166, doi:10.1093/logcom/12.1.149.

[14] J.A. Robinson (1965): *A Machine–Oriented Logic Based on the Resolution Principle*. *Journal of the ACM* 12(1), pp. 23–41, doi:10.1145/321250.321253.

[15] D. Walther, C. Lutz, F. Wolter & M. Wooldridge (2006): *ATL Satisfiability is Indeed ExpTime-complete*. *Journal of Logic and Computation* 16(6), pp. 765–787, doi:10.1093/logcom/exl009.