

Expressing Properties of Coalitional Ability under Resource Bounds ^{*}

Natasha Alechina Brian Logan Nguyen Hoang Nga Abdur Rakib

School of Computer Science
University of Nottingham
Nottingham, UK
{nza,bsl,hnn,rza}@cs.nott.ac.uk

Abstract. We introduce Coalition Logic for Resource Games (CLRG) which extends Coalition Logic by allowing explicit reasoning about resource endowments of coalitions of agents and resource bounds on strategies. We show how to express interesting properties of coalitional ability under resource bounds in this logic, including properties of Coalitional Resource Games introduced by Wooldridge and Dunne in [12]. We also give an efficient model-checking algorithm for CLRG which makes it possible to verify the properties automatically.

1 Introduction

There are many problems in multi-agent systems which are not reducible to the abilities of individual agents in the system, and which can only be usefully analysed in terms of the combined abilities of groups of agents. For example, it may be that no single agent has a strategy to reach a particular state on its own, but two agents cooperating with each other are capable of achieving this outcome. It is often natural to formulate reasoning about the abilities of coalitions of agents in terms of games, and there is a considerable amount of work on coalitional games and logics in the literature, e.g., [8, 9, 7, 11, 1, 2, 10, 6]. For example, *Coalition Logic* can be used to reason about the coalitional ability of agents in strategic games. Coalition Logic generalises the notion of a strategic game, in that it is interpreted over state transition systems where each state has an associated strategic game. It can be seen either as a way of reasoning about a sequence of strategic games or as reasoning about a single complex game where each move corresponds to a transition to a new state [8]. *Alternating-Time Temporal Logic* (ATL) was originally developed to reason about computational processes in adversarial environments, and has been shown to generalise Coalition Logic [7]. Instead of talking about the outcome of a strategic game in the next state, ATL can express properties holding in arbitrary future states, or maintained during

^{*} An earlier version of this work was presented at the workshop on Logics for Agents and Mobility 2009.

a series of moves. Using these logics we can express properties such as ‘coalition C has a strategy to bring about a state satisfying ϕ ’ (no matter what the other agents in the system do) where ϕ characterises, e.g., the solution to a problem or the successful execution of a protocol.

However, none of the existing coalition logics can express properties of coalitional abilities under resource restrictions, that is, directly express properties such as ‘coalition C has a strategy to bring about ϕ under resource bound b_1 , but not under a tighter resource bound b_2 ’. Examples of situations where such properties are critical are numerous; e.g., whether a team of agents can achieve a task under a given allocation of resources [12], or whether a group of distributed reasoning agents can answer a query given specified memory, communication and time resources [3].

In this paper we present *Coalition Logic for (Strategic) Resource Games* (CLRG). CLRG allows us to express properties such as ‘coalition C can enforce a state satisfying ϕ under a resource bound b ’, that is, the cost of their joint action is at most b (where b is a tuple of resource value pairs $\langle r_1:b_1, \dots, r_j:b_j \rangle$ from a fixed set of resources R). We extend strategic games to *strategic resource games* by adding resources, endowments and costs, and define Coalition Logic for (Strategic) Resource Games in much the same way as Coalition Logic is defined with respect to strategic games. As illustrations of how CLRG can be used to express the abilities of coalitions under resource bounds, we show how to express some of the decision problems for Coalitional Resource Games (CRGs) given in [12], and how to express properties of agents in a simple multi-shot game. Finally, we show how to automatically verify properties expressed in CLRG, in particular properties of CRGs, using a standard model-checker, and give a more efficient model-checking algorithm specifically for CLRG formulas.

2 Coalition Logic

In this section we briefly describe Coalition Logic (CL) introduced by Pauly in [9] and state the main results on the complexity of model-checking CL.

Coalition Logic is used for reasoning about coalitional ability in strategic games. The language of coalition logic contains modalities, $[C]$, for each possible set of agents (coalition) C . The meaning of $[C]\phi$ is that the coalition C can choose a tuple of actions so that whatever actions are chosen by the other agents in the system, the outcome state satisfies ϕ (C can enforce an outcome state satisfying ϕ).

Definition 1. A strategic game form is a tuple $(A, \{Act_i \mid i \in A\}, S, o)$ where

- $A = \{1, \dots, n\}$ is a set of agents,
- $\{Act_i \mid i \in A\}$ is a set of sets of actions (or strategies) for each agent $i \in A$,
- S is a non-empty set of states,
- $o : \prod_{i \in A} Act_i \rightarrow S$ is an outcome function which associates with every tuple of actions, by all agents in parallel, an outcome state in S .

The set of all game forms for a set of players A and a set of states S will be denoted $\Gamma(A, S)$. For a set of agents $C \subseteq A$, we will denote a tuple of actions by the agents in C by a_C , and a tuple of actions by all agents where agents in C execute actions a_C and agents $\bar{C} = A \setminus C$ execute actions $a_{\bar{C}}$ as $(a_C, a_{\bar{C}})$.

Definition 2. A game model for a set of players A over a set of propositions $Prop$ is a triple (S, γ, V) , where

- S is a non-empty set of states,
- $\gamma : S \rightarrow \Gamma(A, S)$ is a mapping associating a strategic game form with each state in S (we will use an extra argument s to distinguish components of $\gamma(s)$, for example $o(s, a_1, \dots, a_n)$ for the outcome function in state s),
- $V : S \rightarrow 2^{Prop}$ is a valuation function which labels every state in S with a set of propositions that are true in that state.

The language L of CL (parameterised by a set of propositional variables $Prop$ and a set of agents A) is as follows:

$$p \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid [C]\phi$$

where $p \in Prop$ and $C \subseteq A$.

Formulas of L are evaluated with respect to game models as follows:

- $M, s \models p$ iff $p \in V(s)$
- $M, s \models \neg\phi$ iff $M, s \not\models \phi$
- $M, s \models \phi \wedge \psi$ iff $M, s \models \phi$ and $M, s \models \psi$
- $M, s \models [C]\phi$ iff there exists a_C such that for every $a_{A \setminus C}$, $M, o(s, a_C, a_{A \setminus C}) \models \phi$.

The model-checking problem for CL (given a formula ϕ and a model M , return the set $[\phi]_M$ of states satisfying ϕ) can be reduced to the model-checking problem for ATL using the embedding into ATL given in [7], and therefore can be done in time $O(|\phi| \times m)$ where m is the number of transitions in the model.

Coalition logic allows us to express interesting properties both of strategic games and of multi-shot games, using nested $[C]$ operators. For example, we can express that a group of agents C has a three-step winning strategy in a game by saying $[C][C][C]$ win. However CL does not allow us to express properties of coalitional abilities under resource restrictions.

3 Strategic resource game forms

In this section we extend strategic games to strategic resource games (or rather game forms, since we do not have a notion of preference over outcomes) by adding resources to strategic games. Strategic resource game forms will be used to define models of coalition logic for resource games in the next section.

Definition 3. A strategic resource game form is a tuple $(A, \{Act_i \mid i \in A\}, S, o, R, en, c)$ where

- $A = \{1, \dots, n\}$ is a set of agents,
- $\{Act_i \mid i \in A\}$ is a family of sets of actions for each agent $i \in A$,
- S is a non-empty set of states,
- $o : \prod_{i \in A} Act_i \rightarrow S$ is an outcome function,
- $R = \{r_1, \dots, r_t\}$ is a set of resources,
- $en : A \times R \rightarrow \mathbb{N}$ is a resource endowment function,
- $c : \cup_{i \in A} Act_i \times R \rightarrow \mathbb{N}$ is an action cost function.

We extend strategic games with resources, endowments and costs. Each action an agent can perform requires the consumption of zero or more units of each of a set of resources R (the cost of the action), and each agent is given an endowment of each resource $en(i, r)$ (which may be zero). An action can only be executed by an agent if the endowment of the agent is greater than or equal to the cost of the action: $a \in Act_i$ iff $c(a, r) \leq en(i, r)$ for every resource $r \in R$. The set of all resource game forms for a set of players A , a set of resources R and a set of states S will be denoted $\Gamma(A, S, R)$.

We will use $en(C, r)$ for the sum of $en(i, r)$ for all $i \in C$ and $c(a_C, r)$ for the sum of costs of all actions in a_C : $\sum_{i \in C} c(a_i, r)$. It is sometimes convenient to talk about resource endowment of an agent as a single value, namely a vector $en(i) = \langle r_1 : n_1, \dots, r_t : n_t \rangle$, where $en(i, r_j) = n_j$. The sum of vectors is defined in the usual way (pointwise), so $en(C) = \langle r_1 : n_1, \dots, r_t : n_t \rangle$ where $n_j = \sum_{i \in C} en(i, r_j)$. Similarly, we can talk about a cost of an action $c(a_i)$ as a vector of values, one for each resource, and a cost of a joint action by a coalition C , $c(a_C)$, as a vector corresponding to a pointwise sum of vectors $c(a_i)$ where $i \in C$.

Similarly, it is sometimes convenient to talk about tuples of resource values b of the form $\langle r_1 : n_1, \dots, r_j : n_j \rangle$ (where the indices r_j do not necessarily cover all the values in R). For two tuples of resource values over the same resource indices (referring to the same resources) $b = \langle r_1 : b_1, \dots, r_j : b_j \rangle$ and $v = \langle r_1 : v_1, \dots, r_j : v_j \rangle$ we will also use the usual pointwise comparisons and operations, for example $b \leq v$ iff $b_1 \leq v_1, \dots, b_j \leq v_j$ and $b+v = \langle r_1 : b_1+v_1, \dots, r_j : b_j+v_j \rangle$.

4 Resource game models

We can now define models of the logic corresponding to strategic resource game forms:

Definition 4. A resource game model for a set of players A and resources R over a set of propositions $Prop$ is a triple (S, γ, V) , where

- S is a non-empty set of states,
- $\gamma : S \rightarrow \Gamma(A, S, R)$ is a mapping associating a strategic resource game form with each state in S ,
- $V : S \rightarrow 2^{Prop}$ is a valuation function which labels every state in S with a set of propositions that are true in that state.

Endowments in reachable states reflect the resources which were required to obtain that state: if $s' = o(s, a_1, \dots, a_i)$, then for every resource r , $en(s', i, r) = en(s, i, r) - c(a_i, r)$. Actions are only executable by an agent in a state if the agent's endowment in that state is greater than or equal to the cost of the action. We assume one of the actions is *noop* such that for all resources r , $c(noop, r) = 0$ (this simply allows other agents to execute actions when some agents have run out of resources). A *noop* action by all agents does not change the state: $o(s, noop, \dots, noop) = s$. We assume that all other actions cost some non-zero amount for at least one resource.

5 Coalition Logic for Resource Games

In this section we extend Coalition Logic to a logic for reasoning about resource games. We want to be able to express and verify properties such as ‘coalition C can enforce ϕ under a resource bound b ’, that is, the cost of their joint action is less than b (where b is a tuple of resource value pairs $\langle r_1 : b_1, \dots, r_j : b_j \rangle$, from a fixed set of resources R). While endowments constrain the abilities of individual agents to perform actions, resource bounds constrain the abilities of a coalition of agents to perform a joint action. With this in mind, we introduce quantifiers $[C^b]\phi$ where b is a tuple of resource values. It is also useful to be able to refer to the endowments of agents in each state; for this purpose we introduce nullary modalities (endowment counters) $e^=(C, b)$ where C is a coalition and b is a tuple of resource values, which means that the coalition's endowment for the given resources in a given state is equal to b .

The language $L^{b,e}$ of CLRG (parameterised by a set of propositional variables $Prop$, a set of agents A and a set of resources R) is as follows:

$$p \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid [C]\phi \mid [C^b]\phi \mid e^=(C, b)$$

where $p \in Prop$, $C \subseteq A$ and b is a tuple of resource values. Note that $e^{\leq}(C, b)$ can be defined as $\bigvee_{b' \leq b} e^=(C, b')$ and $e^{\geq}(C, b)$ as $\neg e^{\leq}(C, b) \vee e^=(C, b)$. We will write $e^=(i, b)$ for $e^=(\{i\}, b)$.

We will refer to the language obtained from $L^{b,e}$ by omitting modalities $[C^b]$ by L^e , the language obtained from $L^{b,e}$ by omitting endowment counters $e^=(C, b)$ as L^b , and the language of coalition logic as L .

Formulas of $L^{b,e}$ are evaluated with respect to resource game models as follows:

- $M, s \models p$ iff $p \in V(s)$
- $M, s \models \neg\phi$ iff $M, s \not\models \phi$
- $M, s \models \phi \wedge \psi$ iff $M, s \models \phi$ and $M, s \models \psi$
- $M, s \models [C]\phi$ iff there exists a_C such that for every $a_{A \setminus C}$, $M, o(s, a_C, a_{A \setminus C}) \models \phi$.
- $M, s \models [C^b]\phi$ iff there exists a_C with $c(a_C) \leq b$ such that for every $a_{\bar{C}}$, the outcome of the resulting tuple of actions executed in s satisfies ϕ : $M, o(s, a_C, a_{A \setminus C}) \models \phi$
- $M, s \models e^=(C, b)$ iff $en(s, C) = b$.

6 Expressing properties in CLRG

In this section we show how to express some properties of coalitional ability under resource restrictions in the language of CLRG.

6.1 Properties of CRGs

We first show how to express properties of Coalitional Resource Games (CRGs) introduced by Wooldridge and Dunne in [12]. A coalitional resource game Γ is defined as a tuple $(A, G, R, G_1, \dots, G_n, en, req)$ where

- $A = \{1, \dots, n\}$ is a set of agents,
- $G = \{g_1, \dots, g_m\}$ is a set of goals,
- $R = \{r_1, \dots, r_t\}$ is a set of resources,
- $G_i \subseteq G$ is the set of goals for agent i ,
- $en : A \times R \rightarrow \mathbb{N}$ is the resource endowment function (how many units of a given resource is allocated to an agent),
- $req : G \times R \rightarrow \mathbb{N}$ is the resource requirement function (how many units of a particular resource is required to achieve a goal). It is assumed that each goal requires a non-zero amount for at least one resource.

In CRGs, the endowment of a coalition is equal to the sum of the endowments of its members: $en(C, r) = \sum_{i \in C} en(i, r)$. Similarly, the resource requirement for a set of goals is the sum of the requirements for each of the goals in the set: $req(X, r) = \sum_{g \in X} req(g, r)$. A set of goals satisfies a coalition if the intersection with each of the members' goal sets is non empty: X satisfies C if for every $i \in C$, $X \cap G_i \neq \emptyset$. The set of such sets for a coalition C is denoted by $sat(C)$. A set of goals is feasible for a coalition if the coalition has sufficient resources to achieve it: X is feasible for C if for every resource r , $req(X, r) \leq en(C, r)$. The set containing all feasible sets of goals for C is denoted by $feas(C)$. The effectivity function sf returns, for each coalition, the set of sets of goals which both satisfy a coalition and are feasible for it, namely $sat(C) \cap feas(C)$.

Coalitional resource games are defined in terms of goals whereas CLRG is defined in terms of actions. However we show that we can encode a CRG without loss of information in the initial state of a CLRG model, i.e., in a strategic resource game form.

For each CRG Γ , we define a corresponding CLRG model M_Γ . Let $Prop = G \cup \{sat_i \mid i \in A\}$. Intuitively, $g \in G$ holds in a state if this goal has been achieved in this state and sat_i holds in a state if one of agent i 's goals has been achieved in this state. Given a CRG $\Gamma = (A, G, R, G_1, \dots, G_n, en, req)$, we define $M_\Gamma = (S, \gamma, V)$ as follows:

- $S = 2^G$. Intuitively, the initial state is $s_0 = \emptyset$, where no goals have been achieved.
- $V : S \rightarrow 2^{Prop}$ assigns to a state $s = Q \subseteq G$ the set of goals which 'hold' in that state, namely $g \in V(s)$ iff $g \in s$; $sat_i \in V(s)$ if for some $g \in G_i$, $g \in V(s)$.

- for s_0 , $\gamma(s_0)$ is as follows:
 - $Act_i(s_0)$: the actions of each agent are vectors of the form $\langle g_1 : x_1, \dots, g_m : x_m \rangle$ where $g_j : x_j$ means that the agent contributes a vector of resources x_j to the goal g_j . $\langle g_1 : x_1, \dots, g_m : x_m \rangle \in Act_i(s_0)$ iff $\sum_j x_j \leq en(i)$ and $x_j \leq req(g_j)$. In other words, the agents don't contribute more than their endowment or more than the requirement for the goal.
 - $o(s_0, \langle g_1 : x_1^1, \dots, g_m : x_m^1 \rangle, \dots, \langle g_1 : x_1^n, \dots, g_m : x_m^n \rangle) = s$ where $g_j \in s$ iff $\sum_i x_j^i \geq req(g_j)$, that is, g_j is achieved in s iff the agents together contributed sufficient resources to achieve g_j , and otherwise g_j is false in s .
 - $en(s_0, i) = en(i)$.
 - $c(\langle g_1 : x_1, \dots, g_m : x_m \rangle) = \sum_j x_j$, that is the cost of an action is the (vector) sum of resources committed to all the goals in this action.
- for $s \neq s_0$, $Act_i(s) = \{noop\}$ for all i .

Proposition 1. *Given M_Γ as defined above, there is a unique (up to isomorphism) CRG Γ corresponding to M_Γ .*

To show that we can recover Γ from M_Γ , observe that s_0 is the state with the maximal endowment. The set of goals Γ is given, but we can also reconstruct it from the fact that $S = 2^G$. Once we know s_0 , the outcome function in s_0 , and Γ , we can compute the requirements for each goal: the minimal cost of a joint action which achieves (exactly) that goal. Finally, sat_i allows us to compute G_i for each i .

We now show how to express some CRG decision problems introduced in [12] in CLRG. Below for a set $G' \subseteq G$, we define G'^{\vee} and G'^{\wedge} as the disjunction and conjunction respectively of goals in G' .

Successful coalition (Γ, C) : in a CRG Γ , C is a successful coalition. An agent is successful if it achieves at least one of its goals. A coalition is successful if each agent in the coalition is successful. A coalition C is successful in a CRG Γ iff the following formula is true in M_Γ, s_0 :

$$[C] \bigwedge_{i \in C} G_i^{\vee}$$

Maximal coalition (Γ, C) : in a CRG Γ , C is a maximal coalition if any larger coalition is not successful. A coalition C is maximal in a CRG Γ iff the following formula is true in M_Γ, s_0 :

$$\bigwedge_{C' : C \subseteq C'} \neg[C'] \bigwedge_{i \in C'} G_i^{\vee}$$

Maximally successful coalition (Γ, C) : in a CRG Γ , C is a maximally successful coalition if it is maximal and successful. A coalition C is maximally successful in Γ iff the conjunction of the two previous properties holds in M_Γ .

Coalition successful under resource bound b (Γ, C, b) : in a CRG Γ , C is a coalition successful under resource bound b if it can be successful while staying within a resource bound b . A coalition C is successful under resource bound b in Γ iff the following formula is true in M_Γ, s_0 :

$$[C^b] \bigwedge_{i \in C} G_i^\vee$$

Necessary resource (Γ, C, r) : in a CRG Γ , for a coalition C , r is a necessary resource if for every $G' \in sf(C)$, $req(G', r) > 0$. For readability assume that $C = \{1, \dots, k\}$. This property holds iff the following formula is true in M_Γ, s_0 :

$$\bigwedge_{G' \in sat(C)} ([C]G' \rightarrow \neg[C^{(r:0)}]G')$$

where $sat(C) = \{Q_1^\wedge \wedge \dots \wedge Q_k^\wedge \mid i \in C, Q_i \subseteq G_i\}$ (intuitively, a set of formulas saying that at least one of the goals of each agent in C is satisfied).

Strictly necessary resource (Γ, C, r) : r is a strictly necessary resource for C in Γ if $sf(C) \neq \emptyset$ and for every $G' \in sf(C)$, $req(G', r) > 0$. This can be expressed as a conjunction of the successful coalition formula and the necessary resource formula.

G' is optimal for C (Γ, C, G', r) : a set of goals G' is optimal for C in Γ if $G' \in sf(C)$ and for every $G'' \in sf(C)$, $req(G'', r) \geq req(G', r)$. This property holds iff the following formula is true in M_Γ, s_0 :

$$[C]G'^\wedge \wedge \bigwedge_{G'' \in sat(C)} ([C^b]G'' \rightarrow [C^b]G'^\wedge)$$

where $sat(C)$ is as above and $G'^\wedge \in sat(C)$.

G' is Pareto-efficient for C (Γ, C, G') : G' is Pareto-efficient for C in Γ if for every $G'' \in sf(C)$, if for some r_1 , $req(G'', r_1) < req(G', r_1)$, then there exists r_2 such that $req(G'', r_2) > req(G', r_2)$. This property holds iff the following formula is true in M_Γ, s_0 :

$$\bigwedge_{G'' \in sat(C)} ([C^{r_1:v}]G'' \wedge \neg[C^{r_1:v}]G'^\wedge \rightarrow \bigvee_{r \neq r_1} \bigvee_{v \leq en(C,r)} ([C^{r:v}]G'^\wedge \wedge \neg[C^{r:v}]G''))$$

where $sat(C)$ is as above.

Conflicting coalitions (Γ, C_1, C_2, b) : C_1 and C_2 are conflicting coalitions if for every $G_1 \in sf(C_1)$, $G_2 \in sf(C_2)$, G_1 and G_2 are achievable under the resource bound b , but $G_1 \cup G_2$ is not achievable under b . This property holds iff the following formula is true in M_Γ, s_0 :

$$\bigwedge_{G_1 \in sat(C_1), G_2 \in sat(C_2)} ([C_1]G_1 \wedge [C_2]G_2 \rightarrow ([A^b]G_1 \wedge [A^b]G_2 \wedge \neg[A^b](G_1 \wedge G_2)))$$

Positive goal set(Γ, G') : G' is a positive goal set in Γ if there exists a coalition C such that $G' \in sf(C)$. This holds iff the following formula is true in M_Γ, s_0 :

$$\bigvee_{C \subseteq A: G' \wedge \in sat(C)} [C]G' \wedge$$

6.2 Properties of multi-shot games

In addition to expressing properties of CRGs, CLRG can express properties of multi-shot resource games. Consider a simple example. Three agents 1, 2 and 3 are playing two consecutive games in which there is only one resource which is money m . In the first game, they decide whether to get into town (which costs 1 dollar) or stay at home (*noop* action). Once an agent is in town, it can decide to eat in a restaurant X , where dinner costs 25 dollars, or in a restaurant Y , where dinner costs 50 dollars (or skip dinner). The goal of the agents is that two or more agents have dinner together. Assume that their endowments are above 26 dollars. Then although no single agent can enforce the goal, each two-agent coalition can, and so can the grand coalition: $[1, 2][1, 2]$ **12dinner** and $[1, 2, 3][1, 2, 3]$ **123dinner** (where **12dinner** stands for agents 1 and 2 having a joint dinner, similarly for **123dinner**). However if the dinner costs at most 50 dollars (for all participants), the agents can claim the money as project meeting expenses. This can be expressed as achieving a joint dinner under the resource bound of 50 in the second game. Now $[1, 2][1, 2^{(m:50)}]$ **12dinner** but not $[1, 2, 3][1, 2, 3^{(m:50)}]$ **123dinner**.

7 Automated verification of CLRG formulas

While CLRG allows us to express properties of resource games, there are no existing automated verification tools which accept $L^{b,e}$ formulas as specifications. However, in this section we show that it is possible to use a standard Alternating-Time Temporal Logic (ATL) model-checker (such as MOCHA [?]) to automatically verify a property ψ stated in the language $L^{b,e}$ of CLRG by translating it into a formula of L^e which is equivalent to ψ with respect to a fixed model.

Note that under our assumption that each action apart from *noop* has a non-zero cost, each model has a state with the maximal endowment value for each agent (intuitively, the initial state of the system), and all other states have a lower endowment value on at least one resource for one of the agents. This has several implications. First of all, it implies that each model has a finite number of non-identical states. Second, if we know the endowment in the initial state and the cost of each action, we can calculate the set of all possible endowments for each agent for each resource. Let us denote the set of resource endowment tuples (one value for each resource) which agent i can have in a model M by $Q(M, i)$. As a trivial example, if the set of resources $R = \{r_1, r_2\}$, the highest endowment of agent i is $\langle r_1 : 4, r_2 : 3 \rangle$ and in addition to *noop* agent i can execute a single action a with cost $\langle r_1 : 2, r_2 : 2 \rangle$, then the possible endowments of i are $\langle r_1 : 4, r_2 : 3 \rangle$ and $\langle r_1 : 2, r_2 : 1 \rangle$.

Given a finite set $Q(M, i)$ for each i , we can eliminate $[C^b]$ modalities from any formula of CLRG:

Theorem 1. *Given a model M such that $Q(M, i)$ is finite for every i , and a formula ϕ in the language $L^{b,e}$, there exists a formula ϕ' in the language L^e such that for every state s in M , $M, s \models \phi$ iff $M, s \models \phi'$.*

Proof. We define a translation function t which takes a formula $\phi \in L^{b,e}$ and returns $t(\phi) \in L^e$:

- $t(p) = p$, $t(e^=(C, r)) = e^=(C, r)$,
- t commutes with the booleans and $[C]$,
- $t([C^b]\psi) = \bigwedge_{v_1, \dots, v_k \in Q(M, i)} (e^=(1, v_1) \wedge \dots \wedge e^=(k, v_k) \rightarrow [C](t(\psi) \wedge \bigvee_{b_1 + \dots + b_k \leq b} e^=(1, v_1 - b_1) \wedge \dots \wedge e^=(k, v_k - b_k)))$ where, for readability, we assume that $C = \{1, \dots, k\}$

It is straightforward to translate formulas of L^e into ATL. The endowment modalities can be encoded as propositional variables (which are true in a state s if and only if the value of the resource counter in the state satisfies the corresponding condition), and formulas of the form $[C]\phi$ are translated as $\langle\langle C \rangle\rangle X\phi$ (see, for example, [7]).

As an illustration, we show how to translate the following example CRG from [12] into ATL.

$A = \{1, 2, 3\}$; $G = \{g_1, g_2\}$; $R = \{r_1, r_2\}$; $G_1 = \{g_1\}$, $G_2 = \{g_2\}$, $G_3 = \{g_1, g_2\}$; $en(1, r_1) = 2$, $en(1, r_2) = 0$, $en(2, r_1) = 0$, $en(2, r_2) = 1$, $en(3, r_1) = 1$, $en(3, r_2) = 2$; $req(g_1, r_1) = 3$, $req(g_1, r_2) = 2$, $req(g_2, r_1) = 2$, and $req(g_2, r_2) = 1$.

The corresponding model $M_\Gamma = (S, \gamma, V)$ is as follows:

- $S = \{s_0 = \emptyset, s_1 = \{g_1\}, s_2 = \{g_2\}, s_3 = \{g_1, g_2\}\}$
- $V(s_0) = \emptyset$, $V(s_1) = \{g_1, sat_1, sat_3\}$, $V(s_2) = \{g_2, sat_2, sat_3\}$, $V(s_3) = \{g_1, g_2, sat_1, sat_2, sat_3\}$;
- $Act_1(s_0) = \langle g_1 : \langle 0, 0 \rangle, g_2 : \langle 0, 0 \rangle \rangle$, $\langle g_1 : \langle 1, 0 \rangle, g_2 : \langle 0, 0 \rangle \rangle$, $\langle g_1 : \langle 2, 0 \rangle, g_2 : \langle 0, 0 \rangle \rangle$, $\langle g_1 : \langle 0, 0 \rangle, g_2 : \langle 1, 0 \rangle \rangle$, $\langle g_1 : \langle 0, 0 \rangle, g_2 : \langle 2, 0 \rangle \rangle$, $\langle g_1 : \langle 1, 0 \rangle, g_2 : \langle 1, 0 \rangle \rangle$ (the actions available to agents 2 and 3 can be enumerated in a similar way);
- $o(s_0, \langle g_1 : \langle 0, 0 \rangle, g_2 : \langle 0, 0 \rangle \rangle, \langle g_1 : \langle 0, 0 \rangle, g_2 : \langle 0, 0 \rangle \rangle, \langle g_1 : \langle 0, 0 \rangle, g_2 : \langle 0, 0 \rangle \rangle) = \emptyset$ (other action transitions can be enumerated similarly);
- $c(\langle g_1 : \langle 0, 0 \rangle, g_2 : \langle 0, 0 \rangle \rangle) = \langle 0, 0 \rangle$, for agent 1; the cost of actions 2 and 4 are $\langle 1, 0 \rangle$ and the costs of actions 3, 5 and 6 are $\langle 2, 0 \rangle$ (costs of actions by other agents can be computed in a similar way).

The endowment in s_0 is as in Γ .

Using this translation, we can state properties such as the coalition of agents 1 and 3 can achieve g_1 under the resource bound corresponding to the sum of their endowments: $[1, 3^{(r_1:3, r_2:2)}]g_1$. In ATL, to simplify the translation, we just give the conjunct for the endowment values which evaluate to true:

$$e^=(1, \langle r_1:2, r_2:0 \rangle) \wedge e^=(3, \langle r_1:1, r_2:2 \rangle) \rightarrow$$

$$\langle\langle 1, 3 \rangle\rangle X(g_1 \wedge \bigvee_{b_1+b_2 \leq \langle r_1:3, r_2:2 \rangle} e^=(1, \langle r_1:2, r_2:0 \rangle - b_1) \wedge e^=(3, \langle r_1:1, r_2:2 \rangle - b_2))$$

The possible values of b_1 and b_2 above are all b_1 and b_2 such that $b_1 + b_2 \leq \langle r_1 : 3, r_2 : 2 \rangle$, but the values which make the disjunction in the consequent true are $b_1 = \langle r_1:2, r_2:0 \rangle$ and $b_2 = \langle r_1:1, r_2:2 \rangle$.

It is straightforward to encode a CLRG model for an ATL model checker. As an example, we sketch an encoding for the *reactive modules* system description language used by MOCHA model checker.

States of the CLRG models correspond to an assignment of values to state variables in the model checker. The agent's resource endowments are encoded as vectors of state variables (of range type) and the goal propositions are encoded as a set of boolean state variables. We also define a set of vectors of 'contribution variables' for each agent and each goal, which represent the amount of each resource the agent has contributed to achieving the goal. The actions available to the agents are encoded as MOCHA atoms which describe the initial condition and transition relation for a set of related state variables. An action is enabled if the agent's remaining endowment of a resource is greater than or equal to the cost of the action on that resource, i.e., if $en(i, r) \geq c(a_i, r)$. Performing an action decrements the endowment variables for the agent by the cost of the action on each resource, and increments the contribution variables for the proposition(s) affected by the action by the same amount.

Each agent is encoded as a MOCHA module. A module is a collection of atoms and a specification of which of the state variables updated by those atoms are visible from outside the module. For technical reasons, it is convenient to associate the state variables encoding the goal propositions with a separate 'valuation' module. The valuation module aggregates the effects of each individual agent's actions as encoded in the contribution variables and determines the truth value of the goal propositions affected by the actions. A particular game is then simply a parallel composition of the appropriate agent modules and the valuation module.

Using the translation above, CRG problems can be translated from CLRG into ATL with resource counters without increasing the size of the formula (since the CLRG formulas contain no nested modalities). However the translation may lead to an exponential increase in the size of the ATL formula if the CLRG formula contains nested $[C^b]$ modalities. This naturally raises the question of whether there is a special purpose model-checking algorithm for $L^{b,e}$ formulas which avoids the exponential blowup.

8 Model-checking problem for CLRG

In this section we describe a model-checking algorithm for CLRG, which is very similar to the model-checking algorithm for ATL [5] except that it only considers 'next state' formulas. We show that bounds on coalition modalities and endowment counters do not increase the complexity of the algorithm which has the same complexity as the ATL algorithm (namely linear).

Theorem 2. *The model-checking problem for CLRG is solvable in linear time. Given a model $M = (S, V, \gamma)$ and an $L^{b,e}$ formula ϕ , there is an algorithm which returns the set of states $[\phi]_M$ satisfying ϕ : $[\phi]_M = \{s \mid M, s \models \phi\}$, which runs in time $O(|\phi| \times m)$ where m is the number of transitions in M .*

Proof. Consider the following symbolic model-checking algorithm:

for every ϕ' in the set of subformulas of ϕ :
 case $\phi' == p$: $[\phi']_M = \{s \mid p \in V(s)\}$
 case $\phi' == e^=(C, b)$: $[\phi']_M = \{s \mid \sum_{i \in C} en(i) = b\}$
 case $\phi' == \neg\psi$: $[\phi']_M = S \setminus [\psi]_M$
 case $\phi' == \psi_1 \wedge \psi_2$: $[\phi']_M = [\psi_1]_M \cap [\psi_2]_M$
 case $\phi' == [C]\psi$: $[\phi']_M = Pre(C, [\psi]_M)$
 case $\phi' == [C^b]\psi$: $[\phi']_M = Pre^b(C, [\psi]_M)$

where Pre and Pre^b can be computed in $O(m)$ as follows:

Computation of $Pre(C, [\psi]_M)$:

1. bucket sort the set of transitions possible from a state. In other words, iterate through the set of all possible transitions $(a_C, a_{A \setminus C})$ from a state s collecting all transitions with the same a_C (i.e., the same set of actions by the agents in C) in the same ‘bucket’ (note that given the mapping γ from states to strategic game forms, the number of buckets is known in advance);¹
2. iterate through the set of buckets and put s in $Pre(C, [\psi]_M)$ if there is a bucket where for all the transitions (joint actions) in the bucket, $o(s, a_C, a_{A \setminus C}) \in [\psi]_M$.

Computation of $Pre^b(C, [\psi]_M)$:

1. bucket sort the set of transitions possible from a state: iterate through the set of all possible transitions $(a_C, a_{A \setminus C})$ from a state s collecting all transitions with the same a_C and where $c(a_C) \leq b$ in the same bucket;
2. iterate through the set of buckets and put s in $Pre^b(C, [\psi]_M)$ if there is a bucket where for all the transitions (joint actions) in the bucket, $o(s, a_C, a_{A \setminus C}) \in [\psi]_M$.

We might hope that the resource bounds in CLRG might reduce the complexity of the model checking problem for formulas with bounds, but this is not the case. In ATL, for an explicitly enumerated model with m states, the maximum length of the pre-image computation for computing $\langle\langle C \rangle\rangle \Box \phi$ is bounded by m . In our case, if $b/d < m$, where d is the smallest resource cost for any action possible by agents in C and any resource, then the maximum length of the pre-image computation may be shorter than for ATL. However the saving cannot be greater than m for each subformula of the property to be checked, and so the overall saving can be no better than linear in the size of the property, which doesn’t change the complexity class.

¹ The buckets serve essentially the same role as the C-move states in [5].

9 Conclusion and further work

We propose a logic CLRG for reasoning about resource limitations on coalitional ability, and show how to express some example problems—including CRG decision problems—in CLRG. As far as we are aware, there are no other automatic tools for solving CRG problems. We show how to use a standard ATL model-checker for verifying CLRG properties. However for nested modalities with resource bounds the translation into ATL may cause an exponential blow-up. We show that this can be avoided with a special purpose model-checking algorithm for CLRG which is (as expected) linear in the size of the formula and the number of transitions in the model. This gives an automatic verification procedure for CRG properties which is EXPTIME (since the model is exponential in the size of a CRG). For NP- and co-NP complete properties this is the best that can be expected. In other work [4] we provide a complete and sound axiomatisation for a related logic (without endowment counters and unbounded $[C]$ operators). Axiomatisation of CLRG is a subject of future work.

Acknowledgements

This work was supported by the Engineering and Physical Sciences Research Council [grant number EP/E031226/1].

References

1. Thomas Ågotnes, Wiebe van der Hoek, and Michael Wooldridge. On the logic of coalitional games. In Hideyuki Nakashima, Michael P. Wellman, Gerhard Weiss, and Peter Stone, editors, *5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006), Hakodate, Japan, May 8-12, 2006*, pages 153–160. ACM, 2006.
2. Thomas Ågotnes, Wiebe van der Hoek, and Michael Wooldridge. Temporal qualitative coalitional games. In Hideyuki Nakashima, Michael P. Wellman, Gerhard Weiss, and Peter Stone, editors, *5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006), Hakodate, Japan, May 8-12, 2006*, pages 177–184. ACM, 2006.
3. Natasha Alechina, Brian Logan, Nguyen Hoang Nga, and Abdur Rakib. Verifying time, memory and communication bounds in systems of reasoning agents. In Lin Padgham, David Parkes, Jörg Müller, and Simon Parsons, editors, *Proceedings of the Seventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, volume 2, pages 736–743, Estoril, Portugal, May 2008. IFAAMAS, IFAAMAS.
4. Natasha Alechina, Brian Logan, Nguyen Hoang Nga, and Abdur Rakib. A logic for coalitions with bounded resources. In *Proceedings of the Twenty First International Joint Conference on Artificial Intelligence*, 2009. To appear.
5. Rajeev Alur, Thomas Henzinger, and Orna Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49(5):672–713, September 2002.

6. Paul E. Dunne, Wiebe van der Hoek, Sarit Kraus, and Michael Wooldridge. Cooperative boolean games. In Lin Padgham, David C. Parkes, Jörg Müller, and Simon Parsons, editors, *7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008), Estoril, Portugal, May 12-16, 2008, Volume 2*, pages 1015–1022. IFAAMAS, 2008.
7. Valentin Goranko. Coalition games and alternating temporal logics. In *Proceeding of the Eighth Conference on Theoretical Aspects of Rationality and Knowledge (TARK VIII)*, pages 259–272. Morgan Kaufmann, 2001.
8. Marc Pauly. *Logic for Social Software*. Ph.D. thesis, ILLC, University of Amsterdam, 2001.
9. Marc Pauly. A modal logic for coalitional power in games. *J. Log. Comput.*, 12(1):149–166, 2002.
10. Michael Wooldridge, Thomas Ågotnes, Paul E. Dunne, and Wiebe van der Hoek. Logic for automated mechanism design - a progress report. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada*, pages 9–16. AAAI Press, 2007.
11. Michael Wooldridge and Paul E. Dunne. On the computational complexity of qualitative coalitional games. *Artif. Intell.*, 158(1):27–73, 2004.
12. Michael Wooldridge and Paul E. Dunne. On the computational complexity of coalitional resource games. *Artif. Intell.*, 170(10):835–871, 2006.