Model Checking Coalitional Games with Priced-Resource Agents

Dario Della Monica\textsuperscript{1}, Margherita Napoli\textsuperscript{2}, and Mimmo Parente\textsuperscript{2}

\textsuperscript{1} ICE-TCS, School of Computer Science, Reykjavik University, Iceland, 
dariodm@ru.is

\textsuperscript{2} Dept. of Computer Science, Università di Salerno, Fisciano, Italy, 
\{napoli|parente\}@unisa.it

Automated verification of multi-agent systems is a significant topic in the recent literature in artificial intelligence [1]. The need of modeling this kind of systems has inspired logical formalisms, the most famous being the Alternating-time Temporal Logics [4] and the Coalition Logic (CL) [13, 14], oriented towards the description of collective behaviors.

The idea of such logics is that agents can join together in teams (or coalitions) and share resources to accomplish a task (reach a goal). In particular, Alternating-time Temporal Logics have been introduced in [4], where the full alternating-time temporal language, denoted by ATL\textsuperscript{+}, has been presented, along with two significant fragments, namely, ATL and ATL\textsuperscript{†}. These logics are natural specification languages for open system, that is, systems whose behavior depends on the interactions with an external entity, usually called the environment.

In [12], Goranko has studied the relationship between the (expressive power of the) two formalisms. In particular, he has shown that CL can be embedded into ATL. Recently these two logics have been used for the verification of multi-agent systems (MAS), where the agents are equipped with a limited amount of resources to reach their goal [2, 3, 6, 7] (more on this in the Related works section below).

The framework we present here hinges on these approaches and represents a further step towards the formalization of such complex systems: multi-agent systems in which agents can cooperate to perform a task and are subject to a limited availability of resources, that is an intrinsic feature of most real-world systems. In particular formulae of the formalisms proposed in [2, 3, 6, 7] allow one to assign an endowment of resources to the agents by means of the so-called team operators (borrowed from ATL). The problem is then to determine whether the agents in the proponent team have a strategy to carry out the assigned goals with that bounded amount of resources, whatever the agents in the opponent team do. Anyway, the treatment of this boundedness presents some weakness, as we will point out below.

Based on the natural observation that, in order to acquire a resource, there is a price to be paid, usually depending also on the availability of the resource on the market, we propose to consider bounded resources that have each a price to be paid by the agents for their use in reaching the goal. Thus differently from the existing approaches, agents are equipped with an amount of money instead of an endowment of resources. Money is in a sense a meta-resource. On one hand, in [2, 3], actions can only consume resources; on the other hand, in [7], the authors state that whenever actions can produce resources the model checking problem is undecidable. It can be easily argued that the undecidability...
comes from the unboundedness production of resources, thus we naturally constrain the way in which actions can produce resources: it is possible for an action to produce a resource in a quantity that is not greater than the amount that has already been consumed so far. Such a notion makes sense as, in practical terms, it allows one to model significant real-world scenarios, such as, acquiring memory by a program, leasing a car during a travel, and, in general, any scenario in which an agent is releasing resources previously acquired.

Finally, we also tackle the problem of coalition formation. How and why agents should aggregate is not a new issue and has been deeply investigated, in past and recent years, in various frameworks, as for example in algorithmic game theory, argumentation settings, and logic-based knowledge representation (see [11, 5]). We face this problem in the setting which an agent is releasing resources previously acquired.

World scenarios, such as, acquiring memory by a program, leasing a car during a travel, and, in general, any scenario in which an agent is releasing resources previously acquired.

related works. In [2], Alechina et al. introduce the logic RBCL, whose language extends the one of CL with explicit representation of resource bounds. In [3], the same authors propose an analogous extension for ATL, called RB-ATL, and give a model checking procedure that runs in time \( O(|\varphi|^2 \times r, S) \), where \( \varphi \) is the formula to be checked, \( S \) is the model, and \( r \) is the number of resources. Thus, if the number of resources is treated as constant, the model checking problem for RB-ATL is in PTIME. However, the problem of determining a lower bound to the model checking problem and, in particular, whether a PTIME algorithm exists even if the number of resources is not treated as a constant factor is left open.

In [7], Bulling and Farwer introduce the logics RAL and RAL⁺. The former represents a generalization of Alechina et al.’s RB-ATL, the latter is ATL⁺ extended with bounded resources. The authors study several syntactic and semantic variants of RAL and RAL⁺ with respect to the (un)decidability of the model checking problem. In particular, while previous approaches only conceive actions consuming resources, they introduce the notion of actions producing resources. It turned out that such a new notion makes the model checking problem undecidable.

The present work is based on [9, 10], where the logic is introduced and the upper bound (EXPTIME) for the model checking problem is given. Here we complete the complexity characterization of the model checking problem by also showing the EXPTIME-hardness. Finally, it is worth pointing out that a further extension of the logic, based on \( \mu \)-calculus, is discussed in [8].

References