An Outline of Parameterised Resource-Bounded ATL

Natasha Alechina[1], Stéphane Demri[2] and Brian Logan[3]

[1]Utrecht University, [2]CNRS, [3]University of Nottingham

LAMAS 2020 8-10 May 2020, online

Context

- strategy logics such as ATL are useful for expressing properties of MAS and synthesising strategies for groups of agents
- we consider strategies that require (and produce) resources
- previous work in strategy logics with resources such as RB$\pm$ATL has focussed on checking whether a specified amount of resources is sufficient to execute a strategy

Strategic modalities with and without resources

- without: do the booster (agent 1) & satellite (agent 2) have a strategy to reach an orbit from which satellite can monitor indefinitely (*m*)

$$\langle\langle\{1,2\}\rangle\rangle \top \, \mathcal{U} \, \langle\langle\{2\}\rangle\rangle \Box \langle\langle\{2\}\rangle\rangle \top \, \mathcal{U} \, m$$

- with: given fuel 20 and battery 15, booster (agent 1) & satellite (agent 2) can launch, and the satellite can stay in orbit using fuel 10 and battery 5 from which it can monitor indefinitely with 5 units of battery

$$\langle\langle\{1,2\}^{20,15}\rangle\rangle \top \, \mathcal{U} \, \langle\langle\{2\}^{10,5}\rangle\rangle \Box \langle\langle\{2\}^{0,5}\rangle\rangle \top \, \mathcal{U} \, m$$

The problem

- we are solving a different problem: what are the constraints on assignments to resource variables (fuel & battery) that make the property true?

$$\langle\langle\{1,2\}^{x_1,x_2}\rangle\rangle \top \,\mathcal{U}\, \langle\langle\{2\}^{y_1,y_2}\rangle\rangle \Box \langle\langle\{2\}^{z_1,z_2}\rangle\rangle \top \,\mathcal{U}\, m$$

- in general, this problem cannot be solved by iteratively checking ever increasing values for $x_1, x_2, y_1, y_2, z_1, z_2$, unless we know the maximal bound on the values

- function form of the model checking problem: return the constraints on resource variables for which a property holds, for example

$$((x_1, x_2) \geq (15, 5) \vee (x_1, x_2) \geq (10, 7)) \wedge (y_1, y_2) = (z_1, z_2) \geq (0, 0)$$

Contribution

- in this paper, we show how to compute the constraints on amounts of resources required to satisfy a formula
- we introduce a new strategy logic ParRB±ATL(n,r) (where *n* is the number of agents and *r* the number of resources) with variables for amounts of resources
- we provide a 2EXPTIME complexity result for extracting the resources required for satisfying a ParRB±ATL(n,r) formula
- for positive formulas, we can extract *Pareto optimal* resource vectors required to achieve coalition goals

ParRB±ATL

- Parameterised Resource-Bounded Alternating Time Temporal Logic (ParRB±ATL) allows us to reason about resource-bounded agents executing joint actions that produce and consume resources:
    - $\langle\langle A^{\mathbf{x}}\rangle\rangle \bigcirc \psi$ means that a coalition A has a strategy executable within resource bound **x** (where **x** is a tuple of variables) to ensure that the next state satisfies ψ
    - $\langle\langle A^{\mathbf{x}}\rangle\rangle \psi_1 \,\mathcal{U}\, \psi_2$ means that A has a strategy executable within resource bound **x** to ensure ψ_2 while maintaining ψ_1
    - $\langle\langle A^{\mathbf{x}}\rangle\rangle \Box \psi$ means that A has a strategy executable within resource bound **x** to ensure that ψ is always true

Strategies

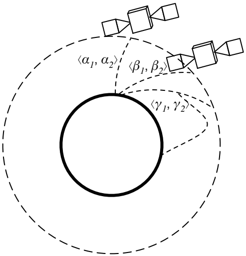
- a strategy for a coalition A is a mapping from finite sequences of states (histories) to joint actions by agents in A
- a strategy achieves $\langle\langle A^{\mathbf{x}} \rangle\rangle \bigcirc \psi$ if all computations generated by the strategy satisfy ψ in the next state and do not consume more than **x** resources
- a strategy achieves $\langle\langle A^{\mathbf{x}} \rangle\rangle \psi_1 \mathcal{U} \psi_2$ if all computations generated by the strategy eventually satisfy ψ_2 and until then satisfy ψ_1, and do not consume more than **x** resources
- a strategy achieves $\langle\langle A^{\mathbf{x}} \rangle\rangle \Box \psi$, if all computations generated by the strategy never consume more than **x** resources and satisfy ψ at every step

Example: what can be expressed in ParRB±ATL

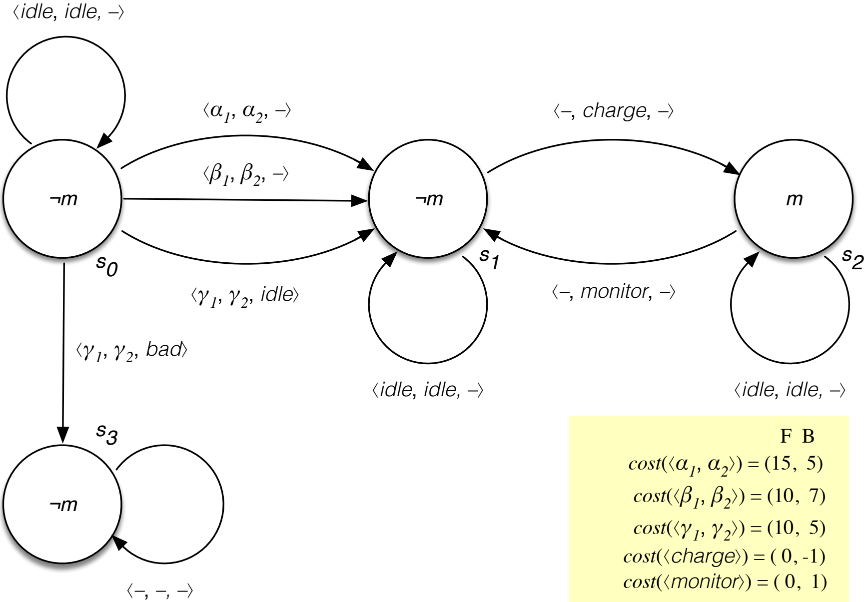
- for which values of fuel and battery can the booster (agent 1) & satellite (agent 2) reach an orbit from which satellite can monitor indefinitely (*m*)

$$\langle\langle \{1,2\}^{x_1,x_2} \rangle\rangle \top \, \mathcal{U} \, \langle\langle \{2\}^{y_1,y_2} \rangle\rangle \Box \langle\langle \{2\}^{y_1,y_2} \rangle\rangle \top \, \mathcal{U} \, m$$

agent 1	agent 2	fuel	battery
α_1	α_2	15	5
β_1	β_2	10	7
γ_1	γ_2	10	5
	charge		-1
	monitor		1



Models of ParRB±ATL



F B

$cost(\langle \alpha_1, \alpha_2 \rangle) = (15, 5)$

$cost(\langle \beta_1, \beta_2 \rangle) = (10, 7)$

$cost(\langle \gamma_1, \gamma_2 \rangle) = (10, 5)$

$cost(\langle charge \rangle) = (0, -1)$

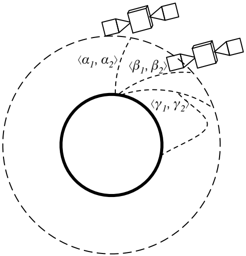
$cost(\langle monitor \rangle) = (0, 1)$

Back to the example

$$\langle\langle\{1,2\}^{x_1,x_2}\rangle\rangle \top\, \mathcal{U}\, \langle\langle\{2\}^{y_1,y_2}\rangle\rangle \Box \langle\langle\{2\}^{y_1,y_2}\rangle\rangle \top\, \mathcal{U}\, m$$

- this property holds in s_0 if (x_1, x_2) are assigned $(15, 5)$ or $(10, 7)$, and (y_1, y_2) are $(0, 0)$, and for all larger values.

agent 1	agent 2	fuel	battery
α_1	α_2	15	5
β_1	β_2	10	7
γ_1	γ_2	10	5
	charge		-1
	monitor		1



Model checking ParRB±ATL

Definition (*function form of the model-checking problem* for ParRB±ATL(n,r))

Input: $n, r \geq 1$ (in unary), a ParRB±ATL(n,r) formula φ, a finite model M, and a state s

Output: a formula γ describing the set of assignments v such that $M, s, v \models \varphi$, where γ is of the form:

$$\gamma := x \geq b \mid \top \mid \bot \mid \neg\gamma \mid \gamma \wedge \gamma \mid \gamma \vee \gamma$$

for $b \in \mathbb{N}$

Complexity of model checking ParRB±ATL

The main result of our paper is the following upper bound:

Theorem

the function form of the model-checking problem for ParRB±ATL can be solved in 2EXPTIME

- 2EXPTIME lower bound is an easy consequence of the result for non-parameterised in RB±ATL:

  Alechina, N., Bulling, N., Demri, S. and Logan, B. (2018). On the complexity of resource-bounded logics. *Theoretical Computer Science* 750:69–100

Upper bound proof idea

- builds on results from games on multi-weighted graphs
- we show that there is a bound on the amount of resource needed to reach a particular state or to enter and execute a non-resource-consuming loop, and this bound depends only on the model and not on the property to be checked
- the bound is exponential in the number of resource types and the largest action cost (max_M) in the model
- we a result from

  Jurdzinski, M., Lazić, R. and Schmitz, S. (2015). Fixed-dimensional energy games are in pseudo-polynomial time. In *Proceedings of ICALP 2015*, 260–272.

  for $\langle\langle A^{\mathbf{x}} \rangle\rangle \Box$ modalities, and extend this to $\langle\langle A^{\mathbf{x}} \rangle\rangle \mathcal{U}$ modalities and the whole language

Finite witnesses for a strategy

- a key in developing a model checking algorithm for the whole language is that of a non-dominated witness for a strategy
- intuitively, a witness for $\langle\langle A^{\mathbf{x}} \rangle\rangle \Box \psi$ is a finite tree describing paths to, and representations of, non-resource consuming loops, and where every state in the tree satisfies ψ
- for $\langle\langle A^{\mathbf{x}} \rangle\rangle \psi_1 \, \mathcal{U} \, \psi_2$ formulas, a witness is a finite tree where the leaves satisfy ψ_2 and all other nodes satisfy ψ_1, and the resource allocation at the root allows *A* to execute all actions in the strategy without ever going negative on any resource type
- such a witness can be extended to an infinite strategy by executing *idle* in all state sequences extending paths from ψ_2 states

2EXPTIME upper bound

- a witness is non-dominated if there is no other witness where the largest resource vector anywhere in the tree (the norm) is strictly less, and no other norm-minimal witness with a strictly smaller value at the root
- the set of non-dominated witnesses for each subformula ϕ' of ϕ_0 can be computed by an alternating Turing machine in EXPSPACE (AEXSPACE)
- hence the function form of the model checking problem can be solved in AEXSPACE
- since ASPACE($f(n)$) = DTIME($2^{O(f(n))}$), AEXPSPACE=2EXPTIME

Pareto optimal bounds

- for positive formulas, the constraint on the assignment is of the form

$$\bigvee_i \mathbf{x} \geq \mathbf{b}_i$$

- since witnesses are non-dominated, the $\mathbf{b}_i$ are Pareto optimal values of resource variables

$$\gamma = ((x_1, x_2) \geq (15, 5) \vee (x_1, x_2) \geq (10, 7)) \wedge (y_1, y_2) = (z_1, z_2) \geq (0, 0)$$

- so the Pareto optimal values are

$$((x_1, x_2) = (15, 5) \vee (x_1, x_2) = (10, 7)) \wedge (y_1, y_2) = (z_1, z_2) = (0, 0)$$

Future work

- implement the algorithm
- study a fragment of parameterised Resource Agent Logic (RAL) where resource values are not 'refreshed' for nested modalities
- the inner strategy (for a nested modality) must use the resources remaining from the outer strategy, e.g.,

$$\langle\langle A^{x} \rangle\rangle \top \mathcal{U} (\phi \wedge \langle\langle A^{\downarrow} \rangle\rangle \top \mathcal{U} \psi)$$

  means that there is a value for x, such that if A have this amount of resources, then they could enforce a state where ϕ holds, and *with remaining resources*, they could enforce ψ