Collective resource bounded reasoning in concurrent multi-agent systems

Valentin Goranko
Stockholm University
(based on joint work with **Nils Bulling**)

Workshop on Logics for Resource-Bounded Agents
ESSLLI'2015, Barcelona, August 14, 2015

Overview of the talk

- Collective agency and resource boundedness
- Concurrent game models with resource costs and action guards
- A running example: robot team on a mission
- QATL*: a quantitative extension of the logic ATL* and its use in collective resource bounded reasoning.
- Concluding remarks.

Introduction: Collective agency and resource sharing

When acting towards achievement of their (qualitative) objectives, agents often act as teams.

As such, they share resources and their collective actions bring about collective updates or redistribution of these resources.

The cost / resource consumption may depend not only on the objective but also on the team of agents acting together.

Some examples:

- A family sharing household and budget
- Joint venture business. Industrial plants.
- Conference fees and organising expenses
- Petrol consumption per passenger in a car, sharing a taxi, paying for a dinner party, etc.

Collective agency and resource sharing: formal modelling and logical reasoning

Here we propose abstract models and logical systems for reasoning about resource sharing collective agency.

The models extend concurrent game models with resource update mechanism, associating with every action profile a table of resource costs for all agents.

Thus, resource consumption and updates are (generally) collective.

The logic extends ATL* with 'resource counters', one per agent. (Extension to multiple resources is quite straightforward.)

Thus, in the course of the play, resources change dynamically.

The available actions for a given agent at every given state depend on the current resource credit of that agent.

All these lead to combined quantitative-qualitative reasoning.

Arithmetic constraints over resources

A simple formal language for dealing with resource updates:

- $R_{\mathbb{A}} = \{r_{\mathbf{a}} \mid \mathbf{a} \in \mathbb{A}\}$:
  set of special variables to refer to the accumulated resources;
- Given sets X and $A \subseteq \mathbb{A}$, the set $T(X, A)$ of terms over X and A is built from $X \cup R_A$ by applying addition.
- Terms are evaluated in domain of payoffs D (usually, $\mathbb{Z}$ or $\mathbb{R}$).
- The set $\mathsf{AC}(X, A)$ of arithmetic constraints over X and A:

$$\{t_1 * t_2 \mid * \in \{<, \leq, =, \geq, >\} \text{ and } t_1, t_2 \in T(X, A)\}$$

- Arithmetic constraint formulae:
  $\mathsf{ACF}(X, A)$: the set of Boolean formulae over $\mathsf{AC}(X, A)$.

Concurrent game models with resource guards and updates

A CGM with guards and resources (CGMGR) is a tuple

$$\mathfrak{M} = (\mathcal{S}, \text{resource}, \{g_{\mathbf{a}}\}_{\mathbf{a}\in\mathbb{A}},$$

where $\mathcal{S} = (\mathbb{A}, \mathsf{St}, \{\mathsf{Act}_{\mathbf{a}}\}_{\mathbf{a}\in\mathbb{A}}, \{\mathsf{act}_{\mathbf{a}}\}_{\mathbf{a}\in\mathbb{A}}, \mathsf{out}, \mathsf{Prop}, \mathsf{L})$ is a CGM and:

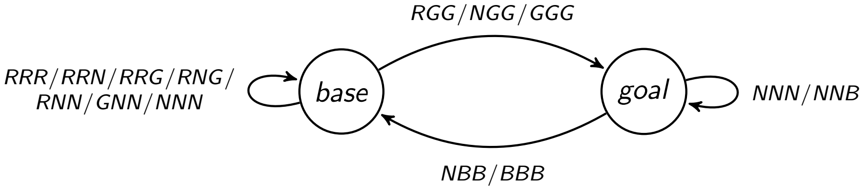
- resource : $\mathbb{A} \times S \times \mathsf{Act}_{\mathbb{A}} \to \mathrm{D}$ is a resource update function.
- accumulated resource of a player **a** at a state of a play: the sum of the initial resource credit and all **a**'s resource updates incurred in the play so far.
- $g_{\mathbf{a}} : S \times \mathsf{Act}_{\mathbf{a}} \to \mathsf{ACF}(X, \{a\})$, for $\mathbf{a} \in \mathbb{A}$, is a guard function such that $g_{\mathbf{a}}(s, \alpha)$ is an ACF for each $s \in \mathsf{St}$ and $\alpha \in \mathsf{Act}_{\mathbf{a}}$.

  $\triangleright$ The action α is available to **a** at s iff the current accumulated resource of **a** satisfies $g_{\mathbf{a}}(s, \alpha)$.

  The guard must enable at least one action for **a** at s.

Example: robots on a mission

Scenario: a team of 3 robots is on a mission. The team must accomplish a certain task, e.g., formalized as ‘reaching state *goal*’.



The robots work on batteries which need to be charged in order to provide the robots with sufficient energy to be able to function.

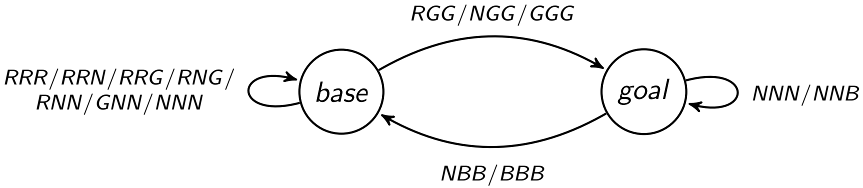
We assume the robots’ energy levels are non-negative integers.

Every action of a robot consumes some of its energy.

Collective actions of all robots may, additionally, increase or decrease the energy level of each of them.

V Goranko

Robots on a mission: agents and states



For every collective action: an 'energy update table' is associated, representing the net change – increase or decrease – of the energy level after that collective action is performed at the given state.

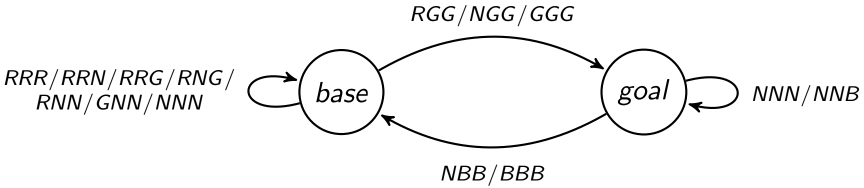
In this example the energy level of a robot may never go below 0.

Here are the detailed descriptions of the components of the model:

Agents: The 3 robots: $\mathbf{a}, \mathbf{b}, \mathbf{c}$.

States: The 'base station' state (*base*) and the target state *goal*.

Robots on a mission: actions and transitions



Actions. The possible actions are:
R: 'recharge', *N*: 'do nothing', *G*: 'go to goal', *B*: 'return to base'.

All robots have the same functionalities and abilities to perform actions, and their actions have the same effect.

Each robot has the following actions possibly executable at the different states: $\{R, N, G\}$ at state *base* and $\{N, B\}$ at state *goal*.

Transitions. The transition function is specified in the figure.
NB: since the robots abilities are assumed symmetric, it suffices to specify the action profiles as multisets, not as tuples.

Robots on a mission: some constraints

- The team has one recharging device which can recharge at most 2 batteries at a time and produces a total of 2 energy units in one recharge step.

  So if 1 or 2 robots recharge at the same time they receive a pro rata energy increase, but if all 3 robots try to recharge at the same time, the device does not charge any of them.

- Transition from one state to the other consumes a total of 3 energy units. If all 3 robots take the action which is needed for that transition (G for transition from *base* to *goal*, and B for transition from *goal* to *base*), then the energy cost of the transition is distributed equally amongst them.

  If only 2 of them take that action, then each consumes 2 units and the extra unit is transferred to the 3rd robot.

- An attempt by a single robot to reach the other state fails and costs that robot 1 energy unit.

Robots on a mission: resource updates

Resource updates. Resource updates are given below as vectors with components that correspond to the order of the actions in the triple, not to the order of the agents who have performed them.

From state *base*:

Actions	Successor	Payoffs
RRR	*base*	(0,0,0)
RRN	*base*	(1,1,0)
RRG	*base*	(1,1,-1)
RNN	*base*	(2,0,0)
RNG	*base*	(2,0,-1)
RGG	*goal*	(3,-2,-2)
NNN	*base*	(0,0,0)
NNG	*base*	(0,0,-1)
NGG	*goal*	(1,-2,-2)
GGG	*goal*	(-1,-1,-1)

From state *goal*:

Actions	Successor	Payoffs
NNN	*goal*	(0,0,0)
NNB	*goal*	(0,0,-1)
NBB	*base*	(1,-2,-2)
BBB	*base*	(-1,-1,-1)

Robots on a mission: guards

At state *base*:

Action	Guard
R	$v \leq 2$
N	*true*
G	$v \geq 2$
B	*false*

At state *goal*:

Action	Guard
R	*false*
N	*true*
G	*false*
B	$v \geq 2$

Guards. The same for each robot. The variable v denotes the current resource of the respective robot. Some explanations:

- Action B is disabled at state *base* and actions R and G are disabled at state *goal*.
- No requirements for the 'do nothing' action N.
- R can only be attempted if the current energy level is ≤ 2.
- For a robot to attempt a transition to the other state, that robot must have a minimal energy level 2.
- Any set of at least two robots can ensure transition from one state to the other, but no single robot can do that.

Configurations, plays and histories in a CGMGR

Configuration in $\mathfrak{M} = (\mathcal{S}, \text{resource}, \{g_{\mathbf{a}}\}_{\mathbf{a}\in\mathbb{A}}, \{d_{\mathbf{a}}\}_{\mathbf{a}\in\mathbb{A}})$:
a pair $(s, \overrightarrow{u})$ of a state s and a vector $\overrightarrow{u} = (u_1, \ldots, u_k)$ of currently accumulated resources of the agents at that state.

The set of possible configurations: $\text{Con}(\mathfrak{M}) = S \times \mathrm{D}^{|\mathbb{A}|}$.

Partial configuration transition function:

$$\widehat{\text{out}} : \text{Con}(\mathfrak{M}) \times \text{Act}_{\mathbb{A}} \dashrightarrow \text{Con}(\mathfrak{M})$$

where $\widehat{\text{out}}((s, \overrightarrow{u}), \overrightarrow{\alpha}) = (s', \overrightarrow{u'})$ iff $\text{out}(s, \overrightarrow{\alpha}) = s'$ and:

(i) the value $u_{\mathbf{a}}$ assigned to $r_{\mathbf{a}}$ satisfies $g_{\mathbf{a}}(s, \alpha_{\mathbf{a}})$ for each $\mathbf{a} \in \mathbb{A}$
(ii) $u'_{\mathbf{a}} = u_{\mathbf{a}} + \text{resource}_{\mathbf{a}}(s, \overrightarrow{\alpha})$ for each $\mathbf{a} \in \mathbb{A}$

The configuration graph on $\mathfrak{M}$ with an initial configuration $(s_0, \overrightarrow{u_0})$ consists of all configurations in $\mathfrak{M}$ reachable from $(s_0, \overrightarrow{u_0})$ by $\widehat{\text{out}}$.

A play in $\mathfrak{M}$: an infinite sequence $\pi = c_0\overrightarrow{\alpha_0}, c_1\overrightarrow{\alpha_1}, \ldots$ from $(\text{Con}(\mathfrak{M}) \times \text{Act})^{\omega}$ such that $c_n \in \widehat{\text{out}}(c_{n-1}, \overrightarrow{\alpha}_{n-1})$ for all $n > 0$.

A history: any finite initial sequence of a play in $\text{Plays}_{\mathfrak{M}}$.

V Goranko

Some configurations and plays in the robots example

$$\begin{tikzcd}[column sep=huge]
base \arrow[loop left, "RRR/RRN/RRG/RNG/ RNN/GNN/NNN"] \arrow[r, bend left, "RGG/NGG/GGG"] & goal \arrow[l, bend left, "NBB/BBB"] \arrow[loop right, "NNN/NNB"]
\end{tikzcd}$$

Initial configuration: $(base, (0, 0, 0))$.

1. The robots do not coordinate and keep trying to recharge forever. The mission fails:

$(base; 0, 0, 0)(RRR), (base; 0, 0, 0)(RRR), (base; 0, 0, 0)(RRR), \ldots$

2. Now the robots coordinate on recharging, 2 at a time, until they each reach energy levels at least 3. Then they all take action G and the team reaches state *goal* and then succeeds to return to

$(base, 0, 0, 0)(RRN), (base, 1, 1, 0)(NRR), (base, 1, 2, 1)(RNR), (base, 2, 2, 2)(RRN),$
$(base, 3, 3, 2)(NNR), (base, 3, 3, 4)(GGG)(goal, 2, 2, 3)(BBB), (base, 1, 1, 2) \ldots$

More configurations and plays in the robots example

$$\begin{tikzcd}
\mathit{base} \arrow[loop left, "\mathit{RRR/RRN/RRG/RNG/}\atop \mathit{RNN/GNN/NNN}"] \arrow[rr, bend left, "\mathit{RGG/NGG/GGG}"] & & \mathit{goal} \arrow[ll, bend left, "\mathit{NBB/BBB}"] \arrow[loop right, "\mathit{NNN/NNB}"]
\end{tikzcd}$$

3. Again the robots coordinate on recharging, but after the first recharge Robot **a** goes out of order. Thereafter **a** does nothing while the other two robots try to accomplish the mission by each recharging as much as possible and then both taking action G. The team reaches state *goal* but cannot return to *base* and remains stuck at state *goal* forever, for one of the two functioning robots does not have enough energy to apply B:

$(\mathit{base}, 0, 0, 0)(\mathit{RRN}), (\mathit{base}, 1, 1, 0)(\mathit{NRR}), (\mathit{base}, 1, 2, 1)(\mathit{NRR}), (\mathit{base}, 1, 3, 2)(\mathit{NRR}), (\mathit{base}, 1, 3, 4)(\mathit{NGG}), (\mathit{goal}, 2, 1, 2)(\mathit{NNB}), (\mathit{goal}, 2, 1, 1)(\mathit{NNN}), \ldots$

4. As above, but now **b** and **c** apply a cleverer plan and succeed together to reach *goal* and then return to *base*:

$(\mathit{base}, 0, 0, 0)(\mathit{RRN}), (\mathit{base}, 1, 1, 0)(\mathit{NRR}), (\mathit{base}, 1, 2, 1)(\mathit{NRR}), (\mathit{base}, 1, 3, 2)(\mathit{NGR}), (\mathit{base}, 1, 2, 4)(\mathit{NRN}), (\mathit{base}, 1, 4, 4)(\mathit{NGG}), (\mathit{goal}, 2, 2, 2)(\mathit{NBB}), (\mathit{base}, 3, 0, 0) \ldots$

The logic of qualitative strategic abilities ATL*

Alternating-time Temporal Logic ATL* involves:

- *Coalitional strategic path operators:* $\langle\langle A\rangle\rangle$ for any coalition of agents A. We will write $\langle\langle \mathbf{i}\rangle\rangle$ instead of $\langle\langle \{\mathbf{i}\}\rangle\rangle$.
- *Temporal operators:* $\mathcal{X}$ (next time), $\mathcal{G}$ (forever), $\mathcal{U}$ (until)

Formulae:

$$\varphi := p \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \langle\langle A\rangle\rangle\varphi \mid \mathcal{X}\,\varphi \mid \mathcal{G}\,\varphi \mid \varphi_1\,\mathcal{U}\,\varphi_2$$

Semantics: in concurrent game models.
Extends the semantics for LTL with the clause:

$\langle\langle A\rangle\rangle\varphi$: "The coalition A has a collective strategy to guarantee the satisfaction of the goal φ" on every play enabled by that strategy.

The Quantitative ATL*: syntax and semantics

State formulae $\varphi ::= p \mid \mathsf{ac} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\!\langle A \rangle\!\rangle\gamma$
Path formulae: $\gamma ::= \varphi \mid \neg\gamma \mid \gamma \wedge \gamma \mid \mathcal{X}\,\gamma \mid \mathcal{G}\,\gamma \mid \gamma\,\mathcal{U}\,\gamma$
where $A \subseteq \mathbb{A}$, $\mathsf{ac} \in \mathsf{AC}$, and $p \in \mathsf{Prop}$.

Given: $\mathfrak{M}$ be a CGMGR, c a configuration, φ state formula, $\gamma, \gamma_1, \gamma_2$ path formulae, $\mathcal{S}^p$ and $\mathcal{S}^o$ two classes of strategies.

$\mathfrak{M}, c \models p$ iff $p \in \mathsf{L}(c^s)$;
$\mathfrak{M}, c \models \mathsf{ac}$ iff $c^u \models \mathsf{ac}$,

$\mathfrak{M}, c \models \langle\!\langle A \rangle\!\rangle\gamma$ iff there is a $\mathcal{S}^p$-strategy s_A such that for all $\mathcal{S}^o$-strategies $s_{\mathbb{A}\setminus A}$: $\mathfrak{M}, \mathsf{outcome\_play}^{\mathfrak{M}}(c, (s_A, s_{\mathbb{A}\setminus A})) \models \gamma$.

$\mathfrak{M}, \pi \models \varphi$ iff $\mathfrak{M}, \pi[0] \models \varphi$,
$\mathfrak{M}, \pi \models \mathcal{X}\,\gamma$ iff $\mathfrak{M}, \pi[1] \models \gamma$,
$\mathfrak{M}, \pi \models \mathcal{G}\,\gamma$ iff $\mathfrak{M}, \pi[i] \models \gamma$ for all $i \in \mathbb{N}$,
$\mathfrak{M}, \pi \models \gamma_1\,\mathcal{U}\,\gamma_2$ iff there is $j \in \mathbb{N}_0$ such that $\mathfrak{M}, \pi[j] \models \gamma_2$ and $\mathfrak{M}, \pi[i] \models \gamma_1$ for all $0 \leq i < j$.

Ultimately, we define $\mathfrak{M}, c \models \varphi$ iff $\mathfrak{M}, c, 0 \models \varphi$.

Expressing properties in QATL*: examples

Suppose the objective of the team of robots on mission, starting from state *base* where each robot has energy level 0, is to eventually reach the state *goal* and then return to the base station.

Below, ‘base’ is an atomic proposition true only at state *base* and, ‘goal’ is an atomic proposition true only at state *goal*.

The following QATL*-formulae are true at $(\mathit{base}, 0, 0, 0)$:

- $\langle\!\langle\rangle\!\rangle\mathcal{G}\,(r_{\mathbf{a}} \geq 0 \wedge r_{\mathbf{b}} \geq 0 \wedge r_{\mathbf{c}} \geq 0)$
- $\neg\langle\!\langle\mathbf{a}\rangle\!\rangle\mathcal{F}\,\mathrm{goal} \wedge \neg\langle\!\langle\mathbf{b}\rangle\!\rangle\mathcal{F}\,\mathrm{goal} \wedge \neg\langle\!\langle\mathbf{c}\rangle\!\rangle\mathcal{F}\,\mathrm{goal}.$
- $\langle\!\langle\mathbf{b},\mathbf{c}\rangle\!\rangle\mathcal{F}\,(\mathrm{goal} \wedge \langle\!\langle\mathbf{a},\mathbf{b},\mathbf{c}\rangle\!\rangle(r_{\mathbf{a}} > 0 \wedge r_{\mathbf{b}} > 0 \wedge r_{\mathbf{c}} > 0)\,\mathcal{U}\,\mathrm{base}).$
- $\langle\!\langle\mathbf{b},\mathbf{c}\rangle\!\rangle\mathcal{F}\,(\mathrm{goal} \wedge \langle\!\langle\mathbf{b},\mathbf{c}\rangle\!\rangle(r_{\mathbf{a}} > 0)\,\mathcal{U}\,(\mathrm{base} \wedge r_{\mathbf{a}} > 0)).$
- $\neg\langle\!\langle\mathbf{b},\mathbf{c}\rangle\!\rangle\mathcal{F}\,(\mathrm{goal} \wedge \langle\!\langle\mathbf{b},\mathbf{c}\rangle\!\rangle\mathcal{F}\,(\mathrm{base} \wedge (r_{\mathbf{b}} > 0 \vee r_{\mathbf{c}} > 0))).$

Model checking in QATL*

Model checking in QATL* on finite models is generally undecidable, even under very weak assumptions.

Still, there are several practically important decidable cases where the configuration space remains finite, e.g.:

- When resources are not created, but only consumed or re-distributed, but cannot become negative.
- When the possible accumulated amount of resource per agent is bounded above and below.

There are some non-trivial decidable cases with infinite configuration spaces, too.

Concluding remarks

Collective resource bounded reasoning in concurrent multi-agent systems is natural and important.

QATL* is a natural and expressive logic for such reasoning.

Model checking is generally undecidable, but there are practically important decidable cases.

Many open problems and directions for further work.
One of them: allow collective guards on the resources of coalitions.

Good case studies are wanted, too.