Outline

1 Introduction

2 Resource Agent Logics

3 Verification of Resource-Bounded Systems
   Problem and Overview
   Undecidability
   Decidability

4 General Quantitative Reasoning Framework

5 Conclusion
1 Introduction

2 Resource Agent Logics

3 Verification of Resource-Bounded Systems
   Problem and Overview
   Undecidability
   Decidability

4 General Quantitative Reasoning Framework

5 Conclusion
Motivation

Logics for MAS: specification and verification

- **Strategic logics**
  - What can teams of agents achieve?
  - Can a set of interacting processes ensure correct functioning?

  \[\text{alternating-time temporal logic (ATL)}\left[\text{Alur et al., 2002}\right]\]

- **Resources**
  present in and crucial for many multi-agent systems
  - Do agents have **sufficient energy** to achieve a task?
  - Can a team of **robots defend the base with the given energy status**?
  - Do agents have enough **resources and capabilities** to complete a task?

  \[\text{many variants with resources: Resource Agent Logics (RAL)}\]
Be careful with resources:
- RAL + unbounded production/consumption of resources
  - (model checking over) Petri nets
  - (model checking over) vector addition systems
- rule of thumb: often undecidability if zero-test can be encoded
- but: **decidable model checking** possible... when?

Today’s talk:
1. introduce general resource-bounded framework
2. review some undecidability results
3. review some decidable cases
4. motivate general quantitative, game theoretic setting ↝ Valentin

**focus of talk:** *key concepts and techniques*
1 Introduction

2 Resource Agent Logics

3 Verification of Resource-Bounded Systems
   Problem and Overview
   Undecidability
   Decidability

4 General Quantitative Reasoning Framework

5 Conclusion
Resource-Bounded Models

- Computational systems often need a notion of resource.
- **Resource-bounded agents**
- Actions *consume* / *produce* resources.
- Non-empty set $\mathcal{R} = \{r_1, \ldots, r_\rho\}$ of *resources.*
A Single Agent Example

Example (Resource-Bounded Tree Logic [Bulling and Farwer, 2010a])

RTL replaces CTL’s path operator: $E \gamma \rightsquigarrow \langle \rho \rangle \gamma$

$\mathcal{M}, q \models \langle \rho \rangle \varphi$ iff $\exists \rho$-feasible path $\lambda$ such that $\mathcal{M}, \lambda \models \varphi$

- feasible path:
  $(q_0, (\infty, 1))(q_1, (\infty, 2))(q_0, (\infty, 4)) \ldots$

- resources $\geq 0$

- $\mathcal{M}, q_0 \models \langle (\infty, 1) \rangle G \top$

- $\mathcal{M}, q_0 \models \langle (1, \infty) \rangle G (p \lor t)$

- Note: nested operators re-set resources $\langle \rho_1 \rangle F \langle \rho_2 \rangle F p$.

Main result: Model checking RTL is decidable (open for RTL$^*$) (reduction to Petri net reachability)
Related Work on Resource Agent Logics

- **Resource-Bounded Coalition Logic** [Alechina et al., 2009]
  - only consumption, Coalition Logic

- **Resource-Bounded Alternating Time Temporal Logic** [Alechina et al., 2014, Alechina et al., 2010]
  - only consumption (**RB-ATL**), axiomatization, model checking, consumption & production, resource flat, proponent restricted, (**RB+-ATL**) ATL -based

- **Resource Agent Logic**
  - [Bulling and Farwer, 2010b, Alechina et al., 2015]
  - consumption & production **RAL**, undecidability & decidability, shared resources, ATL -based

- **Resources and money** [Della Monica et al., 2011]
  - decidability, bounded shared resources, **ATL** -based

**What makes settings (un)decidable?**
Concurrent Game Structures and ATL

Agents:
- execute actions
- cooperate
- model: concurrent game structure

Strategic logic ATL (Alur et al. 1997-2002):
- $\langle A \rangle \gamma$ “Group $A$ has a strategy to guarantee $\gamma$”

ATL: $\varphi ::= p \mid \neg \varphi \mid \varphi \land \varphi \mid \langle A \rangle X \varphi \mid \langle A \rangle G \varphi \mid \langle A \rangle \varphi U \varphi$

ATL*: Allows arbitrary combinations of cooperation and temporal modalities (e.g. $\langle A \rangle GF \varphi$).

Example: $M, q_0 \models \langle 1 \rangle G \neg \text{pos}_1$  $M, q_0 \not\models \langle 1 \rangle F \text{pos}_1$
Variants of Resource agent logics

- transitions have costs (or rewards) and the syntax can **express resource requirements for a strategy**, e.g.:

  *agents* \( A \) *can enforce outcome* \( \varphi \) *if they have at most* \( b_1 \) *units of resource* \( r_1 \) *and* \( b_2 \) *units of resource* \( r_2 \)

In the following:

- consumption & production
- unbounded resources
- all agents may act under resource constraints
**Definition (Resource Agent Logic RAL [Bulling and Farwer, 2010b])**

RAL-formulae are defined by:

\[ \phi ::= p \mid \neg \phi \mid \phi \land \phi \mid \langle\langle A \rangle\rangle_B X \varphi \mid \langle\langle A \rangle\rangle_B ^\eta X \varphi \mid \langle\langle A \rangle\rangle_B \varphi \psi \mid \langle\langle A \rangle\rangle_B ^\eta \varphi \psi \mid \langle\langle A \rangle\rangle_B ^\eta G \varphi \mid \langle\langle A \rangle\rangle_B ^\gamma \varphi \psi \mid \langle\langle A \rangle\rangle_B ^\eta \varphi U \psi \mid \langle\langle A \rangle\rangle_B ^\eta \varphi \psi \]

where \( p \in \Pi \) is a proposition, \( A, B \subseteq \mathbb{Agt} \) are sets of agents, and \( \eta \) is a resource endowment.

\( \langle\langle A \rangle\rangle_B ^\eta \varphi \): agents \( A \) have a strategy compatible with the endowment \( \eta \) to enforce \( \varphi \) whatever the opponent agents do (opponents in \( B \) also act under resource bound \( \eta \))

\( \langle\langle A \rangle\rangle_B ^\gamma \varphi \): agents \( A \) have a strategy compatible with the current resource endowment to enforce \( \varphi \) whatever the opponent agents do (opponents in \( B \) also act under the current resource bound)

**Computational costs:**

\[ \sim \sim \quad \langle\langle A \rangle\rangle_B ^{\eta_1} X \langle\langle A \rangle\rangle_B ^{\eta_2} \gamma \quad \text{vs.} \quad \langle\langle A \rangle\rangle_B ^{\eta_1} X \langle\langle A \rangle\rangle_B ^{\gamma} \]

Nils Bulling (TU Delft)  
Verifying Resource Bounded Agents  
August 11, 2015
Important fragments

**rfRAL:** resource-flat RAL, each nested ATL operator has a fresh assignment of resources ($\langle \langle A \rangle \rangle_B^\uparrow \varphi$ is not allowed):

*given their initial fuel, rescue robots $A$ can safely get to a position from which they can refuel and perform rescue while in visual contact with the base*

$$\langle \langle A \rangle \rangle_A^{\eta_{\text{init}}} (\text{safe } U (\langle \langle A \rangle \rangle_A^{\eta_{\text{refuel}}} (\text{visual } U \text{ rescue})))$$

contrast: $$\langle \langle A \rangle \rangle_A^{\eta_{\text{init}}} (\text{safe } U (\langle \langle A \rangle \rangle_A^\downarrow (\text{visual } U \text{ rescue})))$$

**prRAL:** proponent-restricted RAL, only the strategy of the proponent agents is resource bounded—the opponent agents have no resource bound $\langle \langle A \rangle \rangle_A^\eta \varphi$, $\langle \langle A \rangle \rangle_A^\downarrow \varphi$

**rfprRAL:** combination
Strategies and Their Outcome

- Perfect information perfect recall strategy for agent $a$ (IR-strategy):

  $$s_a : Q^+ \rightarrow Act$$

- Perfect information memoryless strategy for agent $a$ (Ir-strategy):

  $$s_a : Q \rightarrow Act$$

- ATL: it is known that memory does not matter [Alur et al., 2002]

  *if agents can win with memory they can also do so without!*

- RAL: memory does (usually) matter!
3 Verification of Resource-Bounded Systems

Problem and Overview
Undecidability
Decidability

4 General Quantitative Reasoning Framework

5 Conclusion
3 Verification of Resource-Bounded Systems

3.1 Problem and Overview

Undecidability
Decidability

4 General Quantitative Reasoning Framework

5 Conclusion
Model Checking

**Problem**
(e.g. mobile phone) +
(Safety) **Property**
(e.g. deadlock free)

Let's model check...

\[ M \models \varphi = \langle \{1, 2\} \rangle \square \Diamond \top \]

**Computational Complexity?**

**Logical (formal) specification**
Overview: (Un)Decidability

- variants of RTL: language, memory, models
- unbounded production ⇒ mostly undecidable
- overview results of [Bulling and Farwer, 2010b]:

<table>
<thead>
<tr>
<th></th>
<th>$\mathcal{L}_{RAL}^*$</th>
<th>$\mathcal{L}_{RAL}^+$</th>
<th>$\mathcal{L}_{RAL}$</th>
<th>$pr-\mathcal{L}_{RAL}^*$</th>
<th>$pr-\mathcal{L}_{RAL}^+$</th>
<th>$pr-\mathcal{L}_{RAL}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>=R$</td>
<td>$U^1$</td>
<td>$U^1$</td>
<td>$U^1$</td>
<td>$U^1$</td>
<td>$U^1$</td>
</tr>
<tr>
<td>$</td>
<td>=r$</td>
<td>$U^1$</td>
<td>$U^1$</td>
<td>$U^1$</td>
<td>$U^2$</td>
<td>$U^2$</td>
</tr>
<tr>
<td>$rf+_</td>
<td>=R$ / $</td>
<td>=\infty$</td>
<td>$U^2$</td>
<td>$U^2$</td>
<td>$U^2$</td>
<td>$U^2 / U_\infty^2$</td>
</tr>
<tr>
<td>$</td>
<td>=k$/$</td>
<td>=k_R$, $</td>
<td>=k_r$</td>
<td>$D$</td>
<td>$D$</td>
<td>$D$</td>
</tr>
</tbody>
</table>

- Decidability with unbounded production:
  - $RB\pm$ATL [Alechina et al., 2014]:
    (1) resource-flat, (2) proponent restricted, (3) idle action
  - $prRAL^r$ [Alechina et al., 2015]:
    (1) proponent restricted, (2) idle action, (3) positive fragment
  - 1-shared unbounded resource [Bulling and Nguyen, 2015]
3 Verification of Resource-Bounded Systems

3.2 Undecidability
**Undecidability of rfRAL over iRBMs**

<table>
<thead>
<tr>
<th>Models</th>
<th>RAL</th>
<th>rfRAL</th>
<th>prRAL</th>
<th>rfprRAL</th>
</tr>
</thead>
</table>

**RBM** Resource Bounded Models (infinite semantics)

**iRBM** Resource Bounded Models with *idle actions*

We also show undecidability wrt. 1 resource type

1. Bulling & Farwer 2010
2. Alechina et al. 2014
3. Alechina et al. 2015

An aside: **RBM + finitary semantics = iRBM + std. semantics**
High-Level Idea of Reduction

1. reduce halting problem for two counter machines (pushdown automaton with two stacks)

2. encode transition table as an iRBM
two counters simulated by two resource types

3. two agents:
   - (1) simulator agent selects transitions of the automaton
   - (2) the spoiler agent is used to ensure that only valid transitions are selected by the simulator agent

   spoiler agent used to encode zero-test

Observation
Proponent restrictedness is essential for decidability, even over iRBMs
Two-counter automaton [Hopcroft and Ullman, 1979]

**Two-counter automaton** is essentially a PDA with 2 stacks.

Transitions depend on
- state,
- symbol read,
- counters zero or non-zero.

Counters:
- $+1$
- $-1$

Crucial: The logic is used to implement the zero/emptiness test

Does $A$ halt on empty input? $\sim$ undecidable
Counters $\leadsto$ Resource types
Transitions $\leadsto$ Actions
Runs $\leadsto$ strategies/paths + validity condition
Accepting run $\leadsto$ strategies which ensure $F_{halt}$

**Transition relation:**

$$(s, E_1, E_2) \Delta (s', C_1, C_2)$$

$E_i \in \{0, 1\} \quad C_i \in \{-1, 0, +1\}$
\[ \Delta = \{ ((s, E_1, E_2), (s', C_1, C_2)), ((s, E_1, E_2), (s'', C_1, C_2)), ((s', E_1, E_2), (s'', C_1, C_2)) \} \]
Two agents:

- 1: simulate transitions
- 2: ‘spoil” execution in states $sE_1E_2$

$A$ halts on $\varepsilon$ iff

$M^A, s^{\text{init}}, \eta \models_R \langle 1 \rangle^0 \{1,2\} Fp$

Theorem

Model checking rfRAL over iRBMs is undecidable even with 2 agent and 2 resource types.
What about the **single agent** case?

\[
\begin{align*}
A \text{ halts on } \varepsilon & \iff M_1^A, s_{\text{init}}, \eta_0 \models_R \llbracket \{1\} \rrbracket^0 \left( \neg \llbracket \{1\} \rrbracket^\downarrow X \text{ fail} \right) \cup \text{ halt} \\
& \text{Test in error state}
\end{align*}
\]

**Theorem**

*Model checking prRAL over iRBMs is undecidable even with 1 agent and 2 resource types.*
Single Resource Setting

We can adapt the reduction to work with 1 resource type only. Introduce more agents and coordinate their actions.

\[ A \text{ halts on } \varepsilon \text{ iff } M^A_2, s_{\text{init}}, \vec{0} \models_R \langle 1, 2 \rangle^{\vec{0}} \{1, 2, 3, 4\} \text{ Fhalt} \]

Theorem (forthcoming)

Model checking rfRAL over iRBMs is undecidable even with 4 agent and 1 resource type.
A halts on $\varepsilon$ iff $M_1^A$, $s_{\text{init}}$, $\vec{0} \models \langle\{1, 2\}\rangle^\vec{0}(\neg\langle\{1, 2\}\rangle^\downarrow X \text{fail}) U \text{halt}$

**Theorem (forthcoming)**

*Model checking prRAL over iRBMs is undecidable even with 2 agent and 1 resource type.*
3 Verification of Resource-Bounded Systems

Problem and Overview
Undecidability
Decidability

4 General Quantitative Reasoning Framework

5 Conclusion
Decidable Fragments

Formula used in the reduction of prRAL:

\[ M_1^A, s_{\text{init}}, \bar{0} \models \langle\{1, 2\}\rangle^0((\neg\langle\{1, 2\}\rangle^\dagger X \text{fail}) U \text{halt}) \]

Definition (prRAL\(^r\))

prRAL\(^r\) is the positive fragment of prRAL, more precisely, at no coalition modality is under the scope of a negation.

<table>
<thead>
<tr>
<th>Models</th>
<th>RAL</th>
<th>rfRAL</th>
<th>prRAL</th>
<th>rfprRAL</th>
<th>prRAL(^r)</th>
<th>1 shared</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBM</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>D [4]</td>
</tr>
</tbody>
</table>

prRAL$^r$ vs rfprRAL

- given their initial battery charge, rescue robots $A$ can safely get to a position from which they can perform rescue while in visual contact with the base

$$\langle\langle A\rangle\rangle^{\eta_{\text{init}}}(\text{safe } U(\langle\langle A\rangle\rangle \downarrow (\text{visual } U \text{ rescue})))$$

i.e., the robots cannot recharge their batteries after reaching the position from which they can perform rescue

- given their initial fuel and battery, booster (1) & satellite (2) can safely reach a position from which satellite can monitor indefinitely

$$\langle\langle 1, 2\rangle\rangle^{\eta_{\text{init}}}(\text{safe } U(\langle\langle 2\rangle\rangle \downarrow G \text{ monitor}))$$

i.e., satellite has an action to recharge its batteries
Decidability of prRAL' over iRBMs

The algorithm requires as input $M$, $q$, $\eta$, $\phi$ and returns true or false.

1. **Algorithm performs an and-or search of the model**

2. $\langle\langle A \rangle\rangle \downarrow \phi$: propagate the current endowment to the nested search

3. $\langle\langle A \rangle\rangle \eta \phi$: start a new search with endowment $\eta$

4. **Termination**: check for loops with comparable endowments introduce $arb$ if there is a productive loop, finite but arbitrary amount of resources
   - important that no negation is allowed
     $\langle\langle 1 \rangle\rangle F \neg \langle\langle 1 \rangle\rangle \downarrow Fp$: if $arb$ is introduced, 1 has too much power
   - important that only proponent restricted
     $\langle\langle A \rangle\rangle_B Fp$: interplay between $A$ and $B$ tricky when introducing $arb$
   - important that iRBMs are used
     introduction of $arb$ not sufficient $\Rightarrow$ existence of infinite path
Shared Resources

- we consider **shared resources**: common pool
- opponents **always have priority** (similar to [Della Monica et al., 2011])

**Example**

Departmental travel budget. All agents **compete** for the same resources.

**Theorem** ([Bulling and Nguyen, 2015])

RAL over **k-unbounded iRBMs** is **decidable** for \( k \leq 1 \) and **undecidable** otherwise.

Reduction to CTL over alternating Büchi pushdown systems.
4 General Quantitative Reasoning Framework
The following topics are related (conceptually or technically):

- resource logics
- Petri nets
- vector addition systems
- (infinite) games (with quantitative aspects)
- quantitative reasoning tools

Can a **unified framework** help to understand such systems?

- also: resource consumption/production may depend on action profiles \(\leadsto\) closer to game theory
Quantitative Reasoning

Expressing specifications in QATL*[Bulling and Goranko, 2013]:

- QATL* extends ATL*, **qualitative properties**: $\langle\langle A \rangle\rangle(Gp \land qUq)

- Purely **quantitative properties**:
  - $\langle\langle \{a\} \rangle\rangle G(\nu_a > 0)$ “Player a has a strategy to maintain his accumulated payoff positive”,
  - $\langle\langle A \rangle\rangle (\nu_a \geq 3)$ “The coalition A has a strategy to guarantee the value (i.e., limit payoff) of the play for player a to be at least 3”.

- Combined **qualitative and quantitative properties**:
  - $\langle\langle \{a\} \rangle\rangle((a \text{ is happy}) U (\nu_a \geq 100))$
  - $\langle\langle \{a, b\} \rangle\rangle((\nu_a + \nu_b \geq \nu_c) U G(a \text{ is happy})))$

- In general easily **undecidable**
Example

<table>
<thead>
<tr>
<th></th>
<th>I</th>
<th>II</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>2</td>
<td>-3</td>
</tr>
<tr>
<td>D</td>
<td>-3</td>
<td>-1</td>
</tr>
</tbody>
</table>

Prisoners Dilemma

<table>
<thead>
<tr>
<th></th>
<th>I</th>
<th>II</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>-1</td>
<td>2</td>
</tr>
</tbody>
</table>

Battle of Sexes

<table>
<thead>
<tr>
<th></th>
<th>I</th>
<th>II</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>D</td>
<td>-1</td>
<td>1</td>
</tr>
</tbody>
</table>

Coordination Game

\[ u > 0 \Rightarrow \text{any action} \quad u = 0 \Rightarrow C \quad u < 0 \Rightarrow \text{max min payoff} \]

1. \( \langle \langle \{ I, II \} \rangle \rangle \mathbf{F}(p_1 \land v_I > 100 \land v_{II} > 100) \Rightarrow \)
   \( (s_1, (0, 0)), (s_1, (2, 2)), (s_1(4, 4)) \ldots \)

2. \( \langle \langle \{ I, II \} \rangle \rangle \mathbf{XXX} \langle \langle II \rangle \rangle \mathbf{(G}(p_2 \land v_I = 0) \land \mathbf{F} v_{II} > 100) \Rightarrow \)
   \( (s_1, (0, 0)), (s_1, (2, 2)), (s_2, (1, 1)), (s_2(0, -1)), (s_2, (0, 1)), (s_2(0, 3)) \)
1 Introduction

2 Resource Agent Logics

3 Verification of Resource-Bounded Systems
   Problem and Overview
   Undecidability
   Decidability

4 General Quantitative Reasoning Framework

5 Conclusion
Conclusions

• Extensions of ATL

• Main Interest: *What can be verified?*

• decidability depends on many design choices

Future work:

• *implementation* of prRAL$^r$ in MCMAS

• practical settings

• other *decidable fragments* of RAL

• computational complexity
Thank you for your attention.

Questions?
On the boundary of (un)decidability: Decidable model-checking for a fragment of resource agent logic.

A logic for coalitions with bounded resources.

Resource-bounded alternating-time temporal logic.
(to appear).

Decidable model-checking for a resource logic with production of resources.

Alternating-time Temporal Logic.
Journal of the ACM, 49:672–713.
References II

Bulling, N. and Farwer, B. (2010a).
Expressing properties of resource-bounded systems: The logics RBTL and RBTL*.

Bulling, N. and Farwer, B. (2010b).
On the (Un-)Decidability of Model-Checking Resource-Bounded Agents.

How to be both rich and happy: Combining quantitative and qualitative strategic reasoning about multi-player games (extended abstract).
In Proc. of the 1st International Workshop on Strategic Reasoning, EPTCS, pages 33–41, Rome, Italy.

Model checking resource bounded systems with shared resources via alternating Büchi pushdown systems (to appear).
In Proc. of the 18th International Conference on Principles and Practice of Multi-Agent Systems (PRIMA 2015), Bertinoro, Italy.

On a logic for coalitional games with priced-resource agents.

Introduction to Automata Theory, Languages, and Computation.
Addison-Wesley, Reading, Massachusetts.