ESSAI 2024 course: Logic-based specification and verification of multi-agent systems

Lecture 5: How to be both rich and happy?

A logic for combined qualitative and quantitative strategic reasoning

Valentin Goranko
Stockholm University

2nd European Summer School on Artificial Intelligence
ESSAI 2024
Athens, July 15-19, 2024

Based on:

Nils Bulling and Valentin Goranko:
Combining quantitative and qualitative reasoning in concurrent multi-player games,
Journal of Autonomous Agents and Multiagent Systems, 2022, 36:2, 1–33.

Overview of the talk

- Introduction: strategic abilities in multi-player games
  – quantitative and qualitative aspects
- Multi-player concurrent game models
- Concurrent game models with payoffs and guards
- QATL*: a quantitative extension of the logic ATL*
- Model checking of QATL*: some (un)decidability results
- Concluding remarks

Introduction:
strategic abilities of agents in multi-player games

Two traditions:

Game theory: study of rational behavior of players aiming to achieve quantitative objectives: optimizing payoffs or, more generally, preferences on outcomes.

Typical models:
normal form games, repeated games, extensive games.

Logic (and CS): study of strategic abilities of players for achieving qualitative objectives: reaching or maintaining outcome states with desired properties, e.g., winning states, or safe states, etc.

Typical models:
multi-agent transition systems, a.k.a. concurrent game models.

Rich or happy?

In a slogan:

the game theory tradition is concerned with how a player can become maximally rich, or how to pay as little cost as possible,

while the logic tradition – with how a player can achieve a state of 'happiness', e.g. winning, or avoid a state of 'unhappiness' (losing).

So, rich or happy?

Rich or happy? Preferably, both!

In a slogan:

the game theory tradition is concerned with how a player can become maximally rich, or how to pay as little cost as possible,

while the logic tradition – with how a player can achieve a state of 'happiness', e.g. winning, or avoid a state of 'unhappiness' (losing).

Our objective: to bring these two perspectives together within a unifying logical framework.

Wide spectrum of related work:

- resource-bounded reasoning;
- concurrent games with omega-regular objectives;
- mean-payoff and energy parity games;
- counter automata, Petri nets and VASS, timed games; etc.

Concurrent game models recalled

$$(\mathbb{A}, \mathsf{St}, \{\mathsf{Act}_{\mathbf{a}}\}_{\mathbf{a}\in\mathbb{A}}, \{\mathsf{act}_{\mathbf{a}}\}_{\mathbf{a}\in\mathbb{A}}, \mathsf{out}, \mathsf{Prop}, \mathsf{L})$$

- $\mathbb{A} = \{1, \ldots, k\}$ is a fixed finite set of agents (players)
- a set of actions $\mathsf{Act}_{\mathbf{a}} \neq \emptyset$ for each $\mathbf{a} \in \mathbb{A}$.

  For any $A \subseteq \mathbb{A}$ we denote $\mathsf{Act}_A := \prod_{\mathbf{a}\in A} \mathsf{Act}_{\mathbf{a}}$.
- St is a set of system states.
- $\mathsf{act}_{\mathbf{a}} : \mathsf{St} \to \mathcal{P}(\mathsf{Act}_{\mathbf{a}})$ for each $\mathbf{a} \in \mathbb{A}$.

  $\mathsf{act}_{\mathbf{a}}(s)$ is the set of actions available to $\mathbf{a}$ at s.
- $\mathsf{out} : S \times \mathsf{Act}_{\mathbb{A}} \to S$ is a transition function.

  $\mathsf{out}(s, \overrightarrow{\alpha}_{\mathbb{A}})$ is the outcome state for every $q \in \mathsf{St}$ and action profile $\overrightarrow{\alpha}_{\mathbb{A}} = \langle \alpha_1, \ldots, \alpha_{\mathbf{k}} \rangle$ s.t. $\alpha_{\mathbf{a}} \in \mathsf{act}_{\mathbf{a}}(s)$ for each $\mathbf{a} \in \mathbb{A}$.
- Prop is the set of atomic propositions.
- $\mathsf{L} : \mathsf{St} \to \mathcal{P}(\mathsf{Prop})$ is the labeling function.

Towards quantitative reasoning: Concurrent game models with payoffs and guards

Concurrent game models with payoffs and guards (GGMPG): extend concurrent game models by associating with every state a strategic game with payoffs, which can also be interpreted as resources.

– at every state each player chooses an action; all actions are applied simultaneously and determine transition to successor state;

– the collective action also determines each player's payoff;

– same happens at the successor state, etc., thus eventually generating an infinite play;

– so, players accumulate utilities in the course of the play;

– the players' current utility values determine their available actions at the current state, through guards – arithmetical constraints over the current utilities.

Thus, CGMPGs are games with qualitative and quantitative objectives.

Towards quantitative reasoning: arithmetic constraints over payoffs

We need a simple formal language for dealing with payoffs/resources.

- $V_{\mathbb{A}} = \{v_{\mathbf{a}} \mid \mathbf{a} \in \mathbb{A}\}$:
  set of special variables to refer to the accumulated utilities;
- Given sets X and $A \subseteq \mathbb{A}$, the set $T(X, A)$ of terms over X and A is built from $X \cup V_A$ by applying addition.
- Terms are evaluated in domain of payoffs D (usually, $\mathbb{Z}$ or $\mathbb{R}$).
- The set $\mathrm{AC}(X, A)$ 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:
  $\mathrm{ACF}(X, A)$: the set of Boolean formulae over $\mathrm{AC}(X, A)$.

Concurrent game models with payoffs and guards

A guarded CGM with payoffs (GCMGP) is a tuple

$$\mathfrak{M} = (\mathcal{S}, \mathsf{payoff}, \{g_{\mathbf{a}}\}_{\mathbf{a}\in\mathbb{A}}, \{d_{\mathbf{a}}\}_{\mathbf{a}\in\mathbb{A}})$$

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:

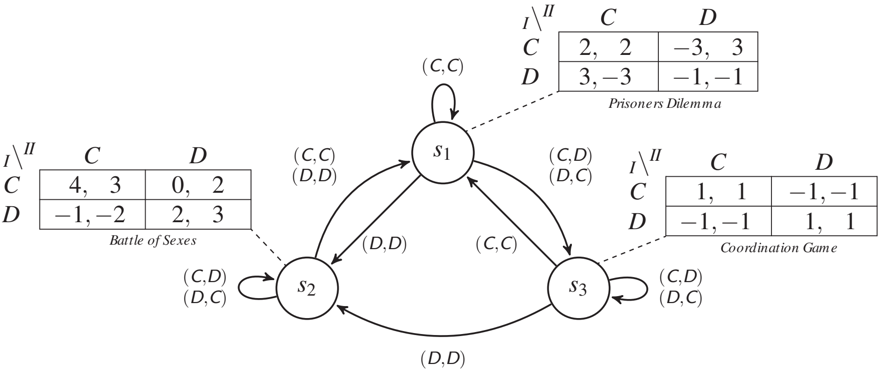
- $\mathsf{payoff} : \mathbb{A} \times S \times \mathsf{Act}_{\mathbb{A}} \to \mathrm{D}$ is a payoff function.
- $d_{\mathbf{a}} \in [0, 1]$ is a discount factor for each $\mathbf{a} \in \mathbb{A}$.
- accumulated utility of a player **a** at a state of a play: the (discounted) sum of all **a**'s payoffs collected in the play so far.

  All initial payoffs are assumed 0.
- $g_{\mathbf{a}} : S \times \mathsf{Act}_{\mathbf{a}} \to \mathsf{ACF}(X, \{a\})$, 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 $\alpha \in \mathsf{Act}_{\mathbf{a}}$.

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

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

CGM with payoffs and guards: a toy game example



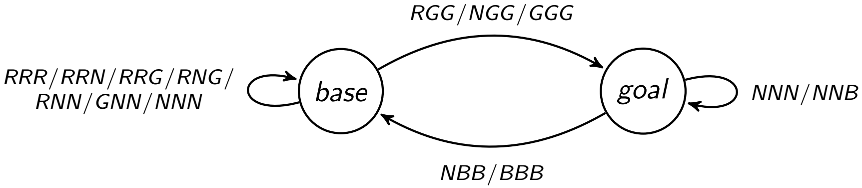
The guards for both players are defined at each state so that the player may:

- apply any action if she has a positive current accumulated utility,
- only apply action C if she has accumulated utility 0,
- must play an action maximizing her minimum payoff in the current game if she has a negative accumulated utility.

The discounting factors are 1 and the initial payoffs of both players are 0.

Example 2: 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*'.



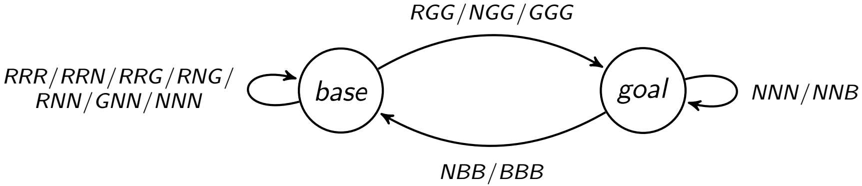
The robots work on batteries which need to be charged in order to provide the robots with sufficient energy to be able to function.

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.

Robots on a mission: agents and states



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

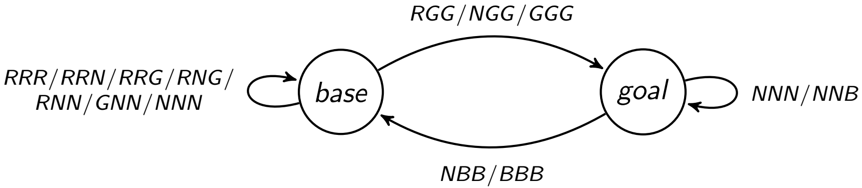
In this example the energy level of a robot may never go below 0.

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 GCMGP

Hereafter we ignore accumulated utilities and discounting.

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

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

Partial configuration transition function:

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

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

(i) the value $u_{\mathbf{a}}$ assigned to $v_{\mathbf{a}}$ satisfies $g_{\mathbf{a}}(s, \alpha_{\mathbf{a}})$ for each $\mathbf{a} \in \mathbb{A}$

(ii) $u'_{\mathbf{a}} = u_{\mathbf{a}} + \mathrm{payoff}_{\mathbf{a}}(s, \overrightarrow{\alpha})$ 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 $\widehat{\mathrm{out}}$.

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

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

Some configurations and plays in the toy example

I \ II	C	D
C	4, 3	0, 2
D	−1, −2	2, 3

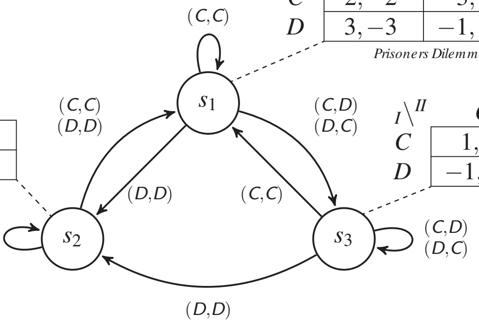
Battle of Sexes

I \ II	C	D
C	2, 2	−3, 3
D	3, −3	−1, −1

Prisoners Dilemma

I \ II	C	D
C	1, 1	−1, −1
D	−1, −1	1, 1

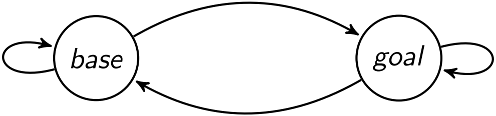
Coordination Game



- ▷ $(s_1, 0, 0)(C, C)(s_1, 2, 2)(C, C)(s_1, 4, 4), \ldots$
- ▷ $(s_1, 0, 0)(C, C)(s_1, 2, 2)(D, D)(s_2, 1, 1)(D, C)(s_2, 0, -1)(C, D)(s_2, 0, 1), (s_2, 0, 3)\ldots$
- ▷ $(s_1, 0, 0)(C, C)(s_1, 2, 2)(D, C)(s_3, 5, -2)(D, C)(s_3, 4, -3)(C, D)(s_3, 3, -4)\ldots$ $(s_3, 0, -7)(C, D)(s_3, -1, -8), \ldots$

NB: If player II has enough memory or can observe the accumulated utilities of I, she can coordinate with I at the round where $v_I = 0$ by cooperating, thus escaping the trap at s_3 and making a sure transition to s_2.

Some configurations and plays in the robots example



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

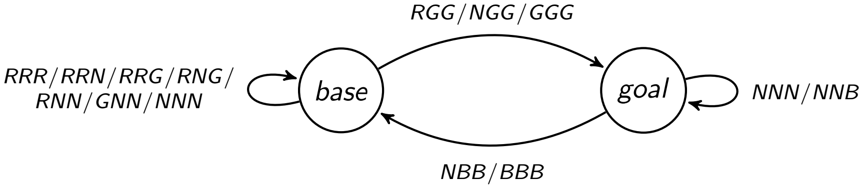
1. The robots do not coordinate and keep trying to recharge forever. The mission fails:

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

2. Now the robots coordinate on recharging, two 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*:

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

More configurations and plays in the robots example



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:

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

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

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

Strategies

A strategy of a player **a** is a function $s_{\mathbf{a}}$: Hist $\rightarrow$ Act that respects the guards, i.e., if $s_{\mathbf{a}}(h) = \alpha$ then $h^u[last]_{\mathbf{a}} \models g_{\mathbf{a}}(h^s[last], \alpha)$.

NB: strategy is based on histories of configurations and actions.

Typically considered in the study of repeated games, e.g., TIT-FOR-TAT or GRIM-TRIGGER in repeated Prisoners Dilemma.

Strategies depend on players' information, memory, observations.

Some natural restrictions: state-, action-, or configuration-based; memoryless, bounded memory, of perfect recall strategies.

We assume that two classes of strategies $\mathcal{S}^p$ and $\mathcal{S}^o$ are fixed as parameters, resp. for the proponents and opponents to select from.

A unique $\text{outcome\_play}_{\mathfrak{M}}(c, (s_A, s_{\mathbb{A}\setminus A}))$ emerges from the execution of any strategy profile $(s_A, s_{\mathbb{A}\setminus A})$ from configuration c.

Effective strategies: bounded memory strategies determined by transducers with transitions defined by arithmetical constraints on the current configurations.

QATL*: Quantitative extension of ATL*

Reminder: the logic of qualitative strategic abilities ATL*

The Alternating-time Temporal Logic 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 $\varphi ::= p \mid \mathrm{ac} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle A \rangle\rangle \gamma$
Path formulae: $\gamma ::= \varphi \mid \mathrm{apc} \mid \neg\gamma \mid \gamma \wedge \gamma \mid \mathcal{X}\gamma \mid \mathcal{G}\gamma \mid \gamma \mathcal{U} \gamma$
where $A \subseteq \mathbb{A}$, $\mathrm{ac} \in \mathrm{AC}$, $\mathrm{apc} \in \mathrm{APC}$, and $p \in \mathrm{Prop}$.

Given: $\mathfrak{M}$ be a GCMGP, 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 $p \in \mathrm{L}(c^s)$;
$\mathfrak{M}, c \models \mathrm{ac}$ iff $c^u \models \mathrm{ac}$,

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

$\mathfrak{M}, \pi \models \varphi$ 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 specifications in QATL*

▷ QATL* extends ATL*, so it can express all purely qualitative ATL* properties, like

$$\langle\langle A\rangle\rangle(\mathcal{G}p \wedge q\mathcal{U}r)$$

▷ QATL* can also express quantitative properties, e.g.:

$$\langle\langle\{\mathbf{a}\}\rangle\rangle\mathcal{G}(v_{\mathbf{a}} > 0)$$

"*Player* **a** *has a strategy to maintain his accumulated utility positive*",

or

$$\langle\langle A\rangle\rangle(w_a \geq 3)$$

"*The coalition A has a strategy to guarantee the value (i.t., limit payoff) of the play for player* **a** *to be at least 3*".

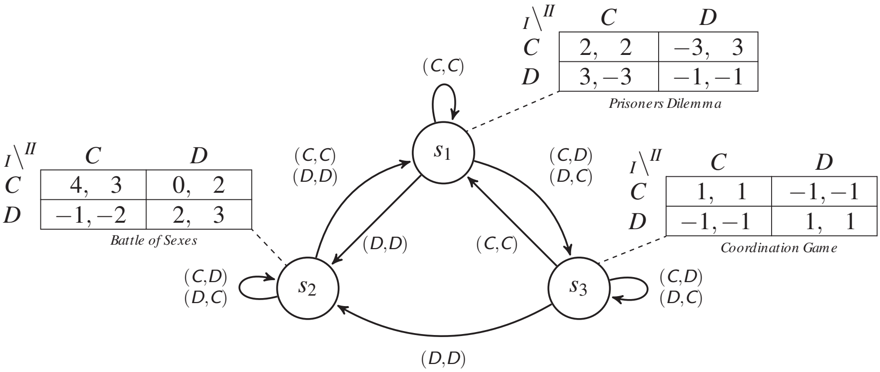
▷ Moreover, QATL* can naturally express combined qualitative and quantitative properties, e.g.

$$\langle\langle\{\mathbf{a}\}\rangle\rangle((\mathbf{a}\ \mathtt{is\ happy})\ \mathcal{U}\ (v_{\mathbf{a}} \geq 10^6))$$

or

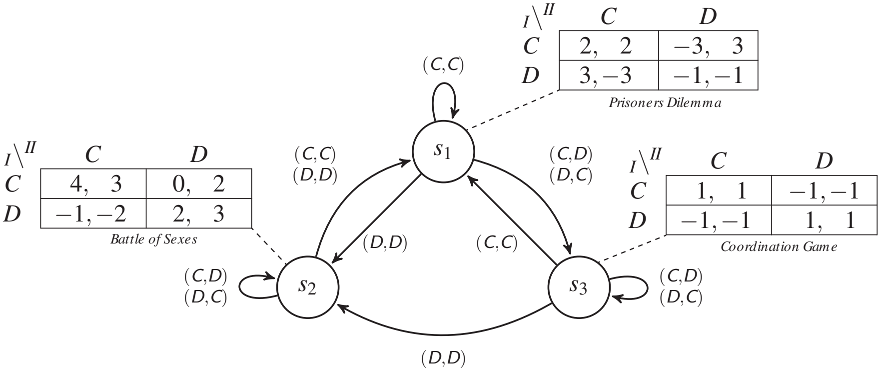
$$\langle\langle\{\mathbf{a},\mathbf{b}\}\rangle\rangle((v_{\mathbf{a}} + v_{\mathbf{b}} > 0)\ \mathcal{U}\ \mathcal{G}\mathtt{safe}))$$

Expressing properties in QATL* for the toy example



In the examples below p_i is true only at s_i, for each $i = 1, 2, 3$.

1. $\langle\langle\{I, II\}\rangle\rangle\mathcal{F}(p_1 \wedge v_I > 100 \wedge v_{II} > 100)$
2. $\langle\langle\{I, II\}\rangle\rangle\mathcal{X}\mathcal{X}\langle\langle\{II\}\rangle\rangle(\mathcal{G}(p_2 \wedge v_I = 0) \wedge \mathcal{F}\ v_{II} > 100)$.
3. $\neg\langle\langle\{I\}\rangle\rangle\mathcal{G}(p_1 \vee v_I > 0)$
4. $\neg\langle\langle\{I, II\}\rangle\rangle\mathcal{F}(p_3 \wedge \mathcal{G}(p_3 \wedge v_I + v_{II} > 0))$.



$u > 0 \Rightarrow$ any action $\quad u = 0 \Rightarrow C \quad u < 0 \Rightarrow$ max min p

1. $\langle\langle\{I, II\}\rangle\rangle\mathcal{F}(p_1 \wedge v_I > 100 \wedge v_{II} > 100)$

$(s_1, (0, 0)), (s_1, (2, 2)), (s_1(4, 4)) \ldots$

2. $\langle\langle\{I, II\}\rangle\rangle\mathcal{X}\mathcal{X}\mathcal{X}\langle\langle\{II\}\rangle\rangle(\mathcal{G}(p_2 \wedge v_I = 0) \wedge \mathcal{F}\ v_{II} > 100)$

$(s_1, (0, 0)), (s_1, (2, 2)), (s_2, (1, 1)), (s_2, (0, -1)), (s_2, (0, 1)), (s_2, (0, 3))$

Expressing properties in QATL* for the *Robots_on_a_mission* example

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 $(base, 0, 0, 0)$:

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

On model checking in QATL*: reduction from the Halting problem for Minsky machines

The framework of QATL* is very general and easily leads to undecidable model checking (on finite models) even under very weak assumptions.

Lemma (Reduction from the Halting problem for Minsky machines)
For any Minsky machine (2-counter automaton) A a finite 2-player GCMGP $\mathfrak{M}^A$ using a proposition `halt` can be constructed so that:

A halts on empty input iff
there is a play π in $\mathfrak{M}^A$ which reaches a `halt`-state.

Undecidability results about model checking QATL*

Using that reduction, undecidability of the model checking in QATL* can be proved under quite week assumptions:

– two players,

– simple temporal objectives, only of the type $\mathcal{X}\varphi, \mathcal{G}\varphi$, and $\varphi\mathcal{U}\psi$, for state formulae φ, ψ.

– no nesting of strategic operators,

– simple arithmetical constraints, only comparing players' utilities with constants, not with each other.

– and state-based guards.

Some decidability results and conjectures about QATL*

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

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

There are some non-trivial decidable cases with infinite configuration spaces, too, by reduction to VASS reachability and coverability problems or to energy parity games.

For further details, as well as some open problems and conjectures, see the full paper.

Concluding remarks

This is a long-term interdisciplinary project, involving Logic, Game Theory and CS. There is a wide spectrum of related work.

- Three perspectives of research agenda:
  - Game theory: solution concepts, equilibria, extending results from repeated games (e.g., folk theorems), etc.
  - Logic: Expressiveness, formal reasoning, deduction.
  - Computation: decidability, algorithms and complexity for model checking and synthesis, incl. solving games, computing winning strategies, optimizing payoffs, etc.
- Many still unexplored directions:
  - solution concepts and equilibria
  - games with imperfect information
  - stochastic games with probabilistic strategies, etc.

The end