How to be both rich and happy: Combining quantitative and qualitative strategic reasoning about multi-player games

Valentin Goranko
Centre International de Mathmatiques et Informatique de Toulouse
(currently on leave from the Technical University of Denmark)

joint work with **Nils Bulling**
Clausthal University of Technology, Germany

IBICS, Université d'Evry-Val-d'Essonne, 19 April 2013

Overview of the lecture

- Strategic abilities in multi-player games: quantitative and qualitative aspects
- Multi-agent transition systems (aka, concurrent game models)
- Concurrent game models with payoffs and guards
- Quantitative extension of the logic ATL*
- 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: 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:
(turn-based or concurrent) multi-agent transition systems
aka, 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).

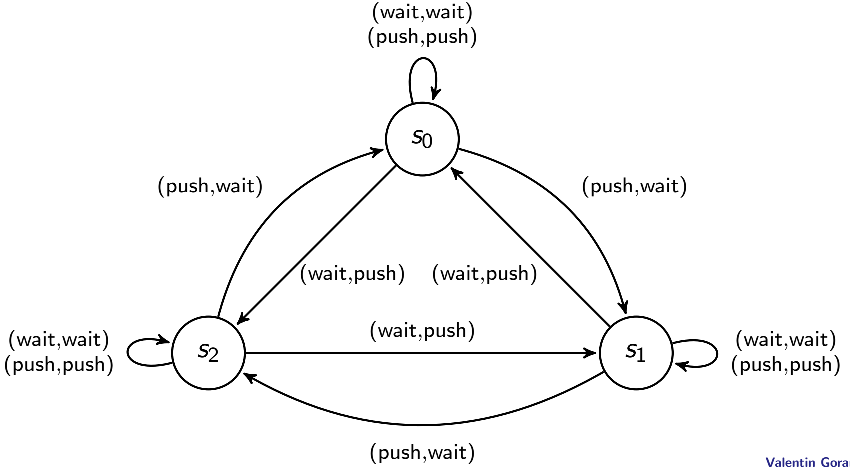
Our objective: to bring these two perspectives together within a unifying logical framework.

Wide spectrum of related work:

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

Multi-agent transition system: example

Two robots pushing a carriage. Robot 1 can only push clockwise and Robot 2 can only push anticlockwise, with the same force.



Game-theoretic perspective on multi-agent transition systems

In concurrent game models agents do not just take actions at every state of the system.

They collectively play games.

Unlike the usual normal form games, the outcomes of these games are not payoffs, but transitions to other games, etc.

Multi-player strategic game forms

A strategic game form is a tuple

$$\langle \mathbb{A}, W, \{\mathrm{Act}_{\mathbf{i}}\}_{\mathbf{i}\in\mathbb{A}}, \rho \rangle$$

where:

- $\mathbb{A}$ is a finite set of agents (players);
- W is a set of possible outcomes;
- $\mathrm{Act}_{\mathbf{i}}$ is a set of actions (moves, strategies) for player $\mathbf{i} \in \mathbb{A}$;
- $\rho : \prod_{\mathbf{i}\in\mathbb{A}} \mathrm{Act}_{\mathbf{i}} \to W$ is the outcome function.

Concurrent game models as multi-stage games

Concurrent game structure (CGS): a set of states S; every state is associated with a strategic game form with outcome states in S.

CGSs model extensive games where:

- at every stage the players play the associated NF game,
  - by making simultaneous moves,
    - each choosing from a set of available actions.

The collective action effects a transition into a successor state;

A play in a CGS is an infinite sequence of successor states.

Concurrent game model (CGM) is a CGS plus labeling of all states with sets of primitive propositions (true at the respective states).

This enables qualitative reasoning.

Concurrent game models formally

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

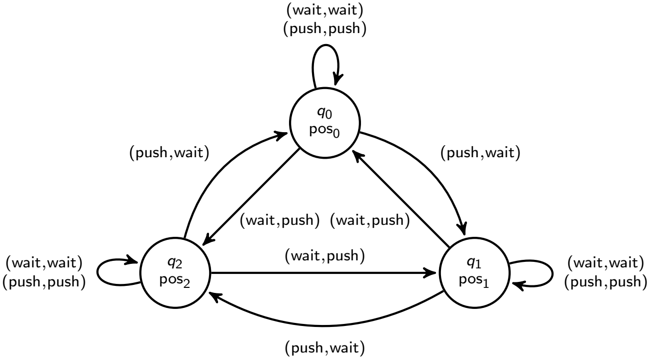
- $\mathbb{A} = \{1, \ldots, k\}$ is a fixed finite set of agents (players)
- a set of actions $\mathsf{Act}_{\mathbf{a}} \neq \emptyset$ for each $\mathbf{a} \in \mathbb{A}$.

  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 **a** at s.
- $\mathsf{out} : S \times \mathsf{Act}_{\mathbb{A}} \to S$ is a transition function.

  $\mathsf{out}(q, \overrightarrow{\alpha}_{\mathbb{A}})$ 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. $\alpha_{\mathbf{a}} \in \mathsf{act}_{\mathbf{a}}(q)$ 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.

The two-robot example as a concurrent game model



- $\mathbb{A} = \{\mathsf{Robot}_1, \mathsf{Robot}_2\}$; $S = \{q_0, q_1, q_2\}$; $\Pi = \{\mathsf{pos}_0, \mathsf{pos}_1, \mathsf{pos}_2\}$.
- $\pi : S \to \mathcal{P}(\Pi)$ defined by $\pi(q_i) = \{\mathsf{pos}_i\}$, for $i = 0, 1, 2$.
- $\mathrm{Act} = \{\mathsf{push}, \mathsf{wait}\}$.
- action function: both actions available to each agent at every state.
- outcome function: as on the figure.

Towards quantitative reasoning: adding payoffs and accumulated payoffs to CGMs

▷ Payoffs added to the games associated with every state.

They may represent profits/rewards or costs/penalties.

▷ In the process of the play, players accumulate payoffs, which determine or guide their further actions.

Payoffs may be discounted in the future.

▷ Players' objectives can involve their accumulated, mean, or limit payoffs.

▷ Players' abilities to perform actions and their strategies may depend on their current accumulated payoffs (available resources).

Towards quantitative reasoning: arithmetic constraints over payoffs

We need a simple formal language for dealing with payoffs.

- $V_{\mathbb{A}} = \{v_{\mathbf{a}} \mid \mathbf{a} \in \mathbb{A}\}$:
  set of special variable to refer to the accumulated payoffs;
- 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 (GCGMP) 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:

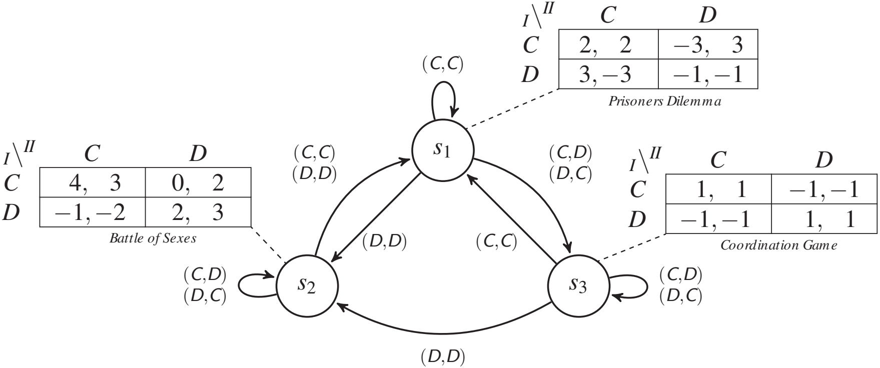
- $\mathsf{payoff} : \mathbb{A} \times S \times \mathsf{Act}_{\mathbb{A}} \to \mathrm{D}$ is a payoff function.
- $d_{\mathbf{a}} \in [0, 1]$ is a discount factor for each $\mathbf{a} \in \mathbb{A}$.
- accumulated payoff of a player **a** at a state of a play: the (discounted) sum of all **a**'s payoffs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

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

Guarded concurrent game model with payoffs: example



I\II	C	D
C	2, 2	−3, 3
D	3, −3	−1, −1

Prisoners Dilemma

I\II	C	D
C	4, 3	0, 2
D	−1, −2	2, 3

Battle of Sexes

I\II	C	D
C	1, 1	−1, −1
D	−1, −1	1, 1

Coordination Game

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

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

Configurations, plays and histories in a GCGMP

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

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

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

(i) $\text{out}(s, \overrightarrow{\alpha}) = s'$

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

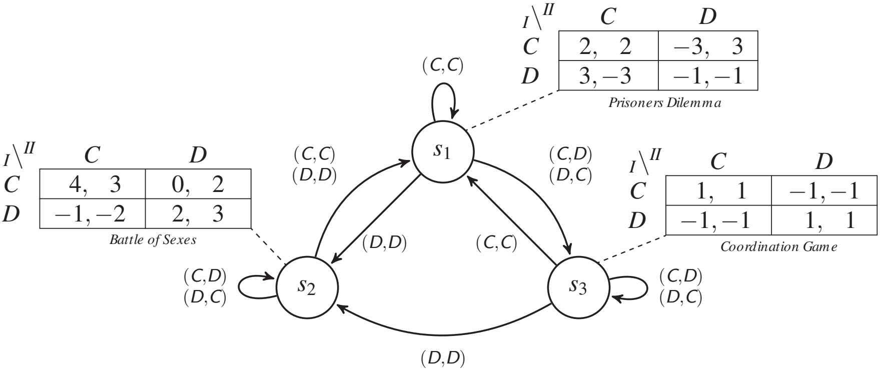
(iii) $u'_{\mathbf{a}} = u_{\mathbf{a}} + d^l_{\mathbf{a}} \cdot \text{payoff}_{\mathbf{a}}(s, \overrightarrow{\alpha})$ for each $\mathbf{a} \in \mathbb{A}$

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 $\widehat{\text{out}}$.

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 $\text{Plays}_{\mathfrak{M}}$.

Configurations and plays: some examples



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

$-(s_1, 0, 0)(C, C)(s_1, 2, 2)(D, D)(s_2, 1, 1)(D, C)(s_2, 0, -1)(C, D)(s_2, 0, 1), (s_2, 0, 3)\ldots$

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

NB: If player II has enough memory or can observe the accumulated payoffs 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.

Strategies

A strategy of a player **a** is a function $s_{\mathbf{a}} : \mathrm{Hist} \rightarrow \mathrm{Act}$ 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 $\mathrm{outcome\_play}_{\mathfrak{M}}(c, (s_A, s_{\mathbb{A} \setminus A}), l)$ emerges from the execution of any strategy profile $(s_A, s_{\mathbb{A} \setminus A})$ from configuration c at the stage l of the game.

The logic of qualitative strategic abilities ATL*

Introduced by Alur, Henzinger, and Kupferman during (1997-2002) under the name Alternating-time Temporal Logic. It 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: $\gamma ::= \varphi \mid \mathsf{apc} \mid \neg\gamma \mid \gamma \wedge \gamma \mid \mathcal{X}\gamma \mid \mathcal{G}\gamma \mid \gamma\mathcal{U}\gamma$
where $A \subseteq \mathbb{A}$, ac $\in$ AC, apc $\in$ APC, and $p \in$ Prop.

Semantics of QATL*:

Let $\mathfrak{M}$ be a GCGMP, c a configuration, $\varphi, \varphi_1, \varphi_2$ state formulae, $\gamma, \gamma_1, \gamma_2$ path formulae, $l \in \mathbb{N}$, $\mathcal{S}^p$ and $\mathcal{S}^o$ two classes of strategies.

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

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

$\mathfrak{M}, \pi, l \models \varphi$ iff $\mathfrak{M}, \pi[0], l \models \varphi$; $\mathfrak{M}, \pi, l \models \mathsf{apc}$ iff $\pi^u, l \models \mathsf{apc}$.

$\mathfrak{M}, \pi, l \models \mathcal{X}\gamma$ iff $\mathfrak{M}, \pi[1], l+1 \models \gamma$,

$\mathfrak{M}, \pi, l \models \mathcal{G}\gamma$ iff $\mathfrak{M}, \pi[i], l+i \models \gamma$ for all $i \in \mathbb{N}$,

$\mathfrak{M}, \pi, l \models \gamma_1 \mathcal{U} \gamma_2$ iff there is $j \in \mathbb{N}_0$ such that $\mathfrak{M}, \pi[j], l+j \models \gamma_2$ and $\mathfrak{M}, \pi[i], l+i \models \gamma_1$ 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 payoff 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*".

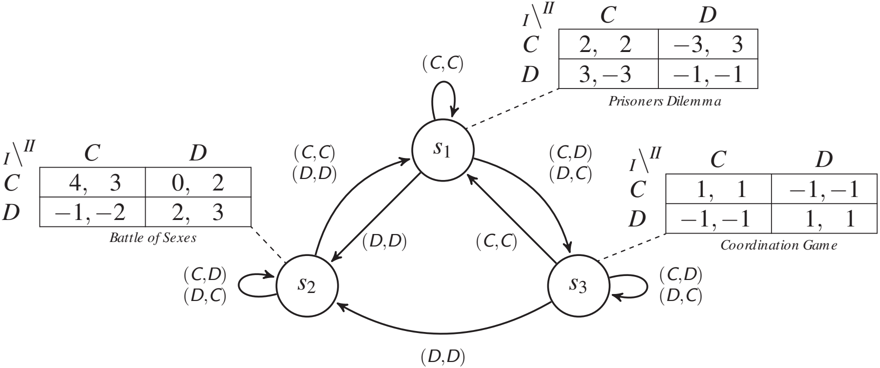
▷ Moreover, QATL* can naturally express combined qualitative and quantitative properties, e.g.

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

or

$$\langle\langle\{\mathbf{a},\mathbf{b}\}\rangle\rangle((v_{\mathbf{a}} + v_{\mathbf{b}} > v_{\mathbf{c}})\ \mathcal{U}\ \mathcal{G}(\mathbf{a}\ \mathtt{is\ happy})))$$

Expressing properties in QATL*: more examples



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. $\neg \langle\langle \{I, II\} \rangle\rangle \mathcal{F}(p_3 \wedge \mathcal{G}(p_3 \wedge v_I + v_{II} > 0))$.

Reduction of qualitative to quantitative reasoning

Idea: given a finite GCGMP $\mathcal{M}$ with a state space St:

1. Label the states by integers, i.e., assume $\mathsf{St} = \{1, \ldots, n\}$.
2. Introduce an extra player **s** with payoff function in $\mathcal{M}$ so that the current payoff of **s** always equals the current state.

   Details: assign only one, unguarded action to **s** at every state. make its payoffs to be: #(successor state) – #(current state).
3. With every $p \in \mathsf{Prop}$ associate the quantitative objective:

$$\delta_{\mathbf{s}}(p) = \bigvee_{i \in \mathsf{L}(p)} (v_{\mathbf{s}} = i)$$

   NB: $\delta_{\mathbf{s}}(p)$ is true at a configuration $(s, \overrightarrow{u})$ iff $p \in \mathsf{L}(s)$.
4. Translate any QATL*-formula ψ into a purely quantitative one $\psi^{\#}$ by replacing every occurrence of each $p \in \mathsf{Prop}$ by $\delta_{\mathbf{s}}(p)$.

NB: the reduction above only works if negative payoffs are allowed.

Research agenda

Three perspectives:

- 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.

Some undecidability results about QATL*

The framework is very general and easily leads to undecidable MC.

Lemma (Reduction from the Halting problem for Minsky machines)
For any Minsky machine A one can construct a finite 2-player GCGMP $\mathfrak{M}^A$ using a proposition `halt` such that A halts on empty input iff there is a play π in $\mathfrak{M}^A$ which reaches a `halt`-state.

Theorem Model checking in the logic QATL* is undecidable, even for the fragment with no nested cooperation modalities, where $\mathcal{S}^p = \mathsf{S}^{pr}$ and $\mathcal{S}^o = \mathsf{S}^m$, in each of the following cases:

1. Two players, no arithmetic constraints in the formula.
2. Two players, state-based guards.
3. Three players, no guards, non-negative payoffs only.

Some decidability results and conjectures about QATL*

Theorem: Model checking in the logic QATL* is decidable in each of the following cases:

1. Many players, memoryless strategies, flat fragment: by reduction to VASS reachability and coverability problems.
2. Two-player turn-based GCGMPs, for the fragment with formulae involving only player 1's accumulated payoff: by reduction to energy parity games.

Conjectures: Model checking in the logic QATL* is decidable in each of the following cases:

1. Two players and non-negative payoffs?
2. Many players, no guards, restriction to only allow in formulae comparisons between players' payoffs and constants, i.e. of the type $v_i > / = / < c$ but not $v_i > / = / < v_j$?
3. Many players, with guards but only memoryless strategies.

Concluding remarks

- ▷ This is the beginning of a long-term project.
- ▷ Intends to strengthen ties between Logic, Game theory and CS.
- ▷ Wide spectrum of related work.
- ▷ Many still unexplored directions:
  - solution concepts and equilibria
  - games with imperfect information
  - stochastic games with probabilistic strategies
  - satisfiability testing and model synthesis