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

Valentin Goranko
Technical University of Denmark
and
Nils Bulling
Clausthal University of Technology, Germany

Highlights'2013
Paris, September 21, 2013

Introduction: strategic abilities of agents in multi-player games

Two traditions:

Introduction: strategic abilities of agents in multi-player games

Two traditions:

Quantitative: study of the abilities of rational players achieve quantitative objectives: optimizing payoffs or, more generally, preferences on outcomes.

Introduction: strategic abilities of agents in multi-player games

Two traditions:

Quantitative: study of the abilities of rational players achieve quantitative objectives: optimizing payoffs or, more generally, preferences on outcomes.

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

Introduction:
strategic abilities of agents in multi-player games

Two traditions:

Quantitative: study of the abilities of rational players achieve quantitative objectives: optimizing payoffs or, more generally, preferences on outcomes.

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

Qualitative: 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.

Introduction: strategic abilities of agents in multi-player games

Two traditions:

Quantitative: study of the abilities of rational players achieve quantitative objectives: optimizing payoffs or, more generally, preferences on outcomes.

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

Qualitative: 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.

Introduction: strategic abilities of agents in multi-player games

Two traditions:

Quantitative: study of the abilities of rational players achieve quantitative objectives: optimizing payoffs or, more generally, preferences on outcomes.

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

Qualitative: 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.

We develop a logical framework combining both traditions.

Introduction: strategic abilities of agents in multi-player games

Two traditions:

Quantitative: study of the abilities of rational players achieve quantitative objectives: optimizing payoffs or, more generally, preferences on outcomes.

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

Qualitative: 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.

We develop a logical framework combining both traditions.

Builds on several existing types of models and logics.

Towards quantitative reasoning: Concurrent game models with payoffs and guards

Towards quantitative reasoning: Concurrent game models with payoffs and guards

Concurrent game model with payoffs and guards (GGMPG): extend concurrent game models by associating with every state a strategic game with payoffs. Thus:

Towards quantitative reasoning: Concurrent game models with payoffs and guards

Concurrent game model with payoffs and guards (GGMPG): extend concurrent game models by associating with every state a strategic game with payoffs. Thus:

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

Towards quantitative reasoning: Concurrent game models with payoffs and guards

Concurrent game model with payoffs and guards (GGMPG): extend concurrent game models by associating with every state a strategic game with payoffs. Thus:

– 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;

Towards quantitative reasoning: Concurrent game models with payoffs and guards

Concurrent game model with payoffs and guards (GGMPG): extend concurrent game models by associating with every state a strategic game with payoffs. Thus:

– 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;

Towards quantitative reasoning: Concurrent game models with payoffs and guards

Concurrent game model with payoffs and guards (GGMPG): extend concurrent game models by associating with every state a strategic game with payoffs. Thus:

– 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;

Towards quantitative reasoning: Concurrent game models with payoffs and guards

Concurrent game model with payoffs and guards (GGMPG): extend concurrent game models by associating with every state a strategic game with payoffs. Thus:

– 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, by means of guards – arithmetical constraints over the current utilities.

Towards quantitative reasoning: Concurrent game models with payoffs and guards

Concurrent game model with payoffs and guards (GGMPG): extend concurrent game models by associating with every state a strategic game with payoffs. Thus:

– 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;

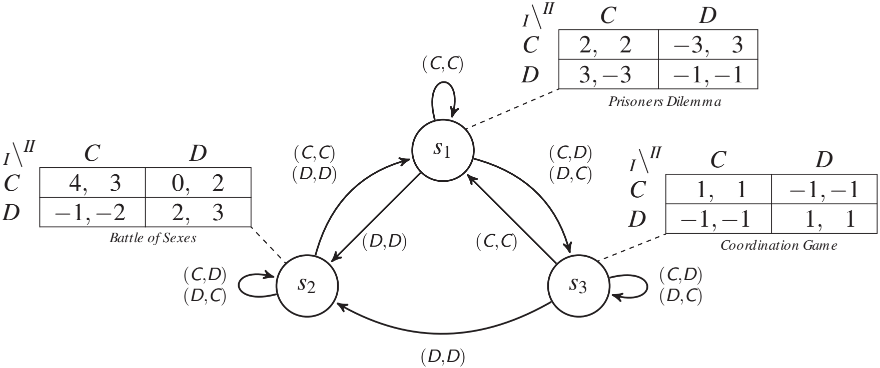
– same happens at the successor state, etc., thus eventually generating an infinite play;

So, players accumulate utilities in the course of the play;

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

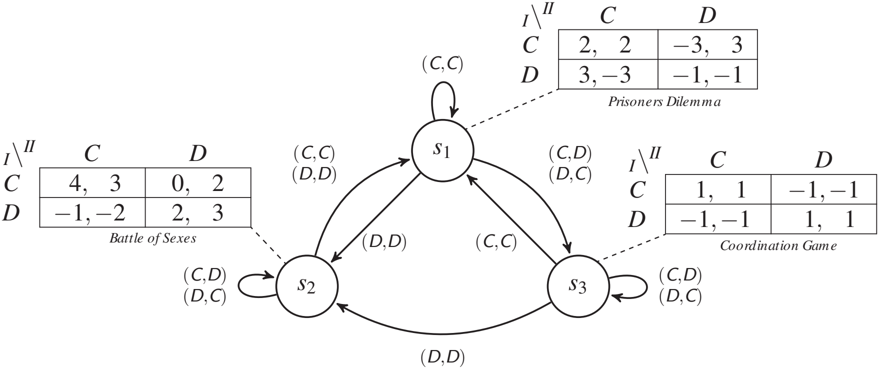
CGMPGs: games with qualitative and quantitative objectives.

Guarded concurrent game model with payoffs: example



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

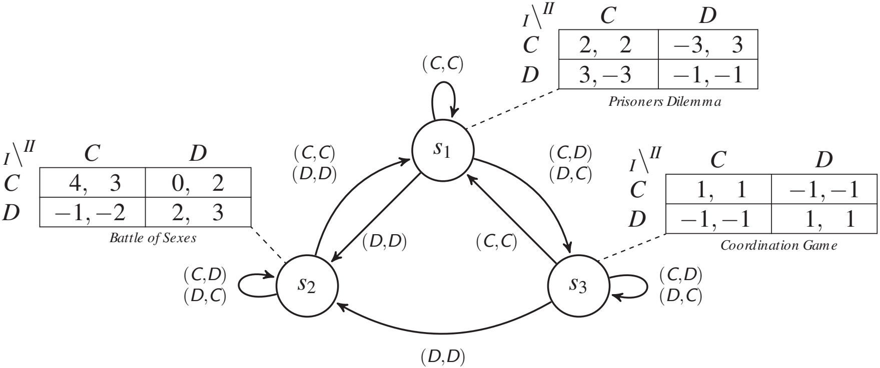
Guarded concurrent game model with payoffs: example



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,

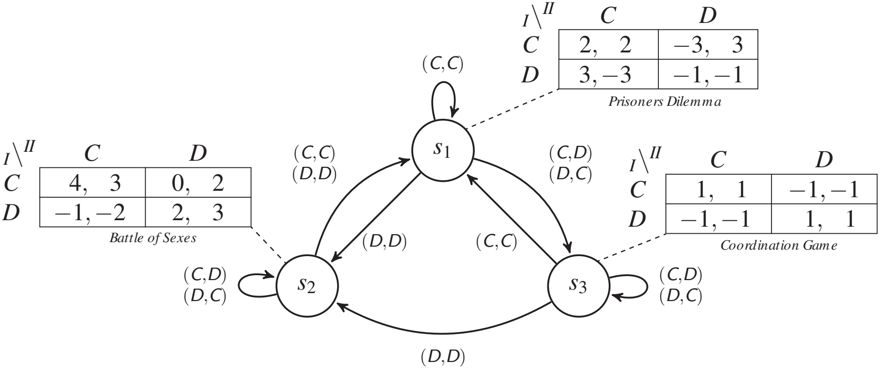
Guarded concurrent game model with payoffs: example



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,

Guarded concurrent game model with payoffs: 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.

Configurations, plays and histories in a GCGMP

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 utilities of the agents at that state.

Configurations, plays and histories in a GCGMP

Configuration in $\mathfrak{M} = (\mathcal{S}, \mathsf{payoff}, \{g_{\mathbf{a}}\}_{\mathbf{a}\in\mathbb{A}}, \{d_{\mathbf{a}}\}_{\mathbf{a}\in\mathbb{A}})$:
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: $\mathsf{Con}(\mathfrak{M}) = S \times \mathrm{D}^{|\mathbb{A}|}$.

Configurations, plays and histories in a GCGMP

Configuration in $\mathfrak{M} = (\mathcal{S}, \mathsf{payoff}, \{g_{\mathbf{a}}\}_{\mathbf{a}\in\mathbb{A}}, \{d_{\mathbf{a}}\}_{\mathbf{a}\in\mathbb{A}})$:
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: $\mathsf{Con}(\mathfrak{M}) = S \times \mathrm{D}^{|\mathbb{A}|}$.

Partial configuration transition function:

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

Configurations, plays and histories in a GCGMP

Configuration in $\mathfrak{M} = (\mathcal{S}, \mathsf{payoff}, \{g_{\mathbf{a}}\}_{\mathbf{a}\in\mathbb{A}}, \{d_{\mathbf{a}}\}_{\mathbf{a}\in\mathbb{A}})$:
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: $\mathsf{Con}(\mathfrak{M}) = S \times \mathrm{D}^{|\mathbb{A}|}$.

Partial configuration transition function:

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

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

Configurations, plays and histories in a GCGMP

Configuration in $\mathfrak{M} = (\mathcal{S}, \mathsf{payoff}, \{g_{\mathbf{a}}\}_{\mathbf{a}\in\mathbb{A}}, \{d_{\mathbf{a}}\}_{\mathbf{a}\in\mathbb{A}})$:
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: $\mathsf{Con}(\mathfrak{M}) = S \times \mathrm{D}^{|\mathbb{A}|}$.

Partial configuration transition function:

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

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

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

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 utilities of the agents at that state.

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

Partial configuration transition function:

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

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

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

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

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 utilities of the agents at that state.

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

Partial configuration transition function:

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

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

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

(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) $u'_{\mathbf{a}} = u_{\mathbf{a}} + \mathsf{payoff}_{\mathbf{a}}(s, \overrightarrow{\alpha})$ for each $\mathbf{a} \in \mathbb{A}$

Configurations, plays and histories in a GCGMP

Configuration in $\mathfrak{M} = (\mathcal{S}, \mathsf{payoff}, \{g_{\mathbf{a}}\}_{\mathbf{a}\in\mathbb{A}}, \{d_{\mathbf{a}}\}_{\mathbf{a}\in\mathbb{A}})$:
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: $\mathsf{Con}(\mathfrak{M}) = S \times \mathrm{D}^{|\mathbb{A}|}$.

Partial configuration transition function:

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

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

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

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

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 $\overrightarrow{u} = (u_1, \dots, u_k)$ of currently accumulated utilities of the agents at that state.

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

Partial configuration transition function:

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

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

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

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

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

Configurations, plays and histories in a GCGMP

Configuration in $\mathfrak{M} = (\mathcal{S}, \mathsf{payoff}, \{g_{\mathbf{a}}\}_{\mathbf{a}\in\mathbb{A}}, \{d_{\mathbf{a}}\}_{\mathbf{a}\in\mathbb{A}})$:
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: $\mathsf{Con}(\mathfrak{M}) = S \times \mathrm{D}^{|\mathbb{A}|}$.

Partial configuration transition function:

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

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

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

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

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

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

Strategies

Strategies

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

Strategies

A strategy of a player **a** is a function $s_{\mathbf{a}}$: Hist $\longrightarrow$ 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: strategies are based on histories of configurations and actions.

Strategies

A strategy of a player **a** is a function $s_{\mathbf{a}} : \mathrm{Hist} \to \mathrm{Act}$ 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: strategies are based on histories of configurations and actions.

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

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: strategies are based on histories of configurations and actions.

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.

Strategies

A strategy of a player $\mathbf{a}$ is a function $s_{\mathbf{a}} : \mathsf{Hist} \rightarrow \mathsf{Act}$ 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: strategies are based on histories of configurations and actions.

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

Strategies

A strategy of a player $\mathbf{a}$ is a function $s_{\mathbf{a}} : \mathsf{Hist} \to \mathsf{Act}$ 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: strategies are based on histories of configurations and actions.

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 $\mathsf{outcome\_play}_{\mathfrak{M}}(c, (s_A, s_{\mathbb{A}\setminus A}))$ 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 and outputs defined by arithmetical constraints on the current configurations.

QATL*: Quantitative extension of ATL*

QATL*: Quantitative extension of ATL*

Language AC of arithmetic formulae over accumulated utilities: Boolean combinations of equalities and inequalities between terms built by applying addition over a set of variables $V_{\mathbb{A}} = \{v_{\mathbf{a}} \mid \mathbf{a} \in \mathbb{A}\}$ for the accumulated utilities and a fixed set X of constants.

QATL*: Quantitative extension of ATL*

Language AC of arithmetic formulae over accumulated utilities: Boolean combinations of equalities and inequalities between terms built by applying addition over a set of variables $V_{\mathbb{A}} = \{v_{\mathbf{a}} \mid \mathbf{a} \in \mathbb{A}\}$ for the accumulated utilities and a fixed set X of constants.

Language of QATL*. Extends ATL* with formulae from AC:

QATL*: Quantitative extension of ATL*

Language AC of arithmetic formulae over accumulated utilities: Boolean combinations of equalities and inequalities between terms built by applying addition over a set of variables $V_{\mathbb{A}} = \{v_{\mathbf{a}} \mid \mathbf{a} \in \mathbb{A}\}$ for the accumulated utilities and a fixed set X of constants.

Language of QATL*. Extends ATL* with formulae from AC:

State formulae $\varphi ::= p \mid \mathsf{ac} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle A \rangle\rangle\gamma$

QATL*: Quantitative extension of ATL*

Language AC of arithmetic formulae over accumulated utilities: Boolean combinations of equalities and inequalities between terms built by applying addition over a set of variables $V_{\mathbb{A}} = \{v_{\mathbf{a}} \mid \mathbf{a} \in \mathbb{A}\}$ for the accumulated utilities and a fixed set X of constants.

Language of QATL*. Extends ATL* with formulae from AC:

State formulae $\varphi ::= p \mid \mathsf{ac} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\!\langle A \rangle\!\rangle\gamma$

Path formulae: $\gamma ::= \varphi \mid \neg\gamma \mid \gamma \wedge \gamma \mid \mathcal{X}\gamma \mid \mathcal{G}\gamma \mid \gamma\mathcal{U}\gamma$

QATL*: Quantitative extension of ATL*

Language AC of arithmetic formulae over accumulated utilities: Boolean combinations of equalities and inequalities between terms built by applying addition over a set of variables $V_{\mathbb{A}} = \{v_{\mathbf{a}} \mid \mathbf{a} \in \mathbb{A}\}$ for the accumulated utilities and a fixed set X of constants.

Language of QATL*. Extends ATL* with formulae from AC:

State formulae $\varphi ::= p \mid \mathrm{ac} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\!\langle A \rangle\!\rangle\gamma$

Path formulae: $\gamma ::= \varphi \mid \neg\gamma \mid \gamma \wedge \gamma \mid \mathcal{X}\gamma \mid \mathcal{G}\gamma \mid \gamma\mathcal{U}\gamma$

where $A \subseteq \mathbb{A}$, ac $\in$ AC and $p \in$ Prop.

QATL*: Quantitative extension of ATL*

Language AC of arithmetic formulae over accumulated utilities: Boolean combinations of equalities and inequalities between terms built by applying addition over a set of variables $V_{\mathbb{A}} = \{v_{\mathbf{a}} \mid \mathbf{a} \in \mathbb{A}\}$ for the accumulated utilities and a fixed set X of constants.

Language of QATL*. Extends ATL* with formulae from AC:

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 $A \subseteq \mathbb{A}$, ac $\in$ AC and $p \in$ Prop.

An extension: with arithmetic formulae over entire plays. Requires adding discounting factors on payoffs. Will not be discussed here.

Semantics of QATL*

Semantics of QATL*

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

Semantics of QATL*

Given: $\mathfrak{M}$ be a GCGMP, 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 $p \in \mathsf{L}(c^s)$;

Semantics of QATL*

Given: $\mathfrak{M}$ be a GCGMP, 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 $c^u \models \mathsf{ac}$,

Semantics of QATL*

Given: $\mathfrak{M}$ be a GCGMP, 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 $s_{\mathbb{A}\setminus A}$: $\mathfrak{M}, \mathsf{outcome\_play}^{\mathfrak{M}}(c, (s_A, s_{\mathbb{A}\setminus A})) \models \gamma$.

Semantics of QATL*

Given: $\mathfrak{M}$ be a GCGMP, 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 $\mathfrak{M}, \pi[0] \models \varphi$,

Semantics of QATL*

Given: $\mathfrak{M}$ be a GCGMP, 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 $s_{\mathbb{A} \setminus A}$: $\mathfrak{M}, \mathrm{outcome\_play}^{\mathfrak{M}}(c, (s_A, s_{\mathbb{A} \setminus A})) \models \gamma$.

$\mathfrak{M}, \pi \models \varphi$ iff $\mathfrak{M}, \pi[0] \models \varphi$,
$\mathfrak{M}, \pi \models \mathcal{X} \gamma$ iff $\mathfrak{M}, \pi[1] \models \gamma$,

Semantics of QATL*

Given: $\mathfrak{M}$ be a GCGMP, 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 $i \in \mathbb{N}$,

Semantics of QATL*

Given: $\mathfrak{M}$ be a GCGMP, 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 $p \in \mathsf{L}(c^s)$;
$\mathfrak{M}, c \models \mathrm{ac}$ 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 $0 \leq i < j$.

Semantics of QATL*

Given: $\mathfrak{M}$ be a GCGMP, 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 $s_{\mathbb{A}\setminus A}$: $\mathfrak{M}, \mathrm{outcome\_play}^{\mathfrak{M}}(c, (s_A, s_{\mathbb{A}\setminus A})) \models \gamma$.

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

Expressing properties in QATL*

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

Expressing properties in QATL*

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

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

Expressing properties in QATL*

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

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

▷ QATL* can also express quantitative properties,

Expressing properties 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*" ,

Expressing properties 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*" ,

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

Expressing properties 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"* ,

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

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

Expressing properties 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.:

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

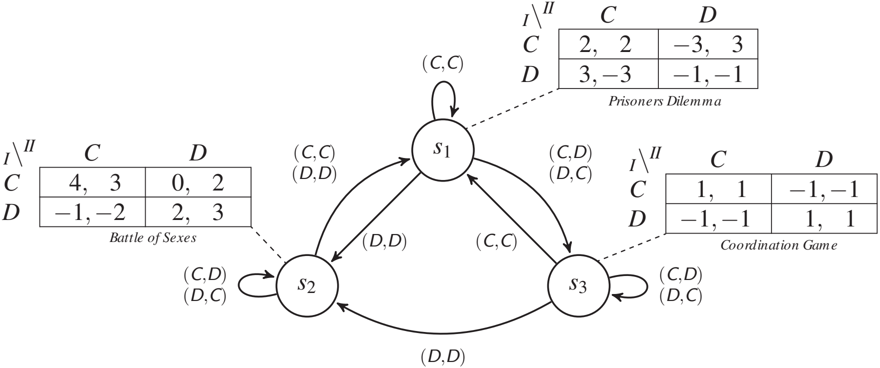
"*Player* **a** *has a strategy to maintain his accumulated utility positive*" ,

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

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

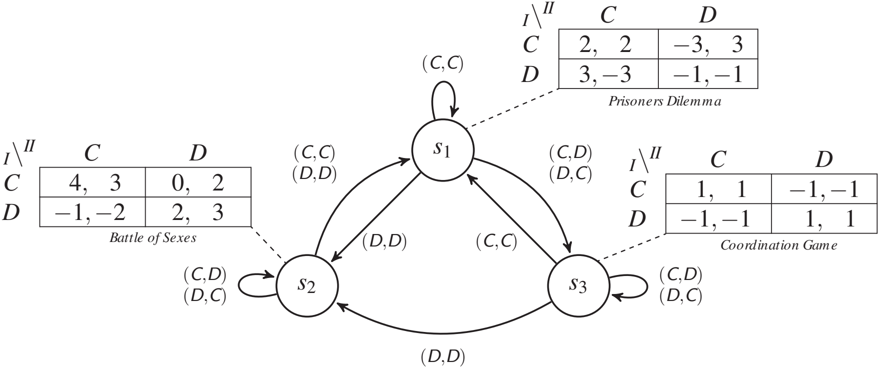
"*Player* **a** *has a strategy to reach accumulated utility of one million and meanwhile stay in "happy" states.*"

Expressing properties in QATL*: more examples



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

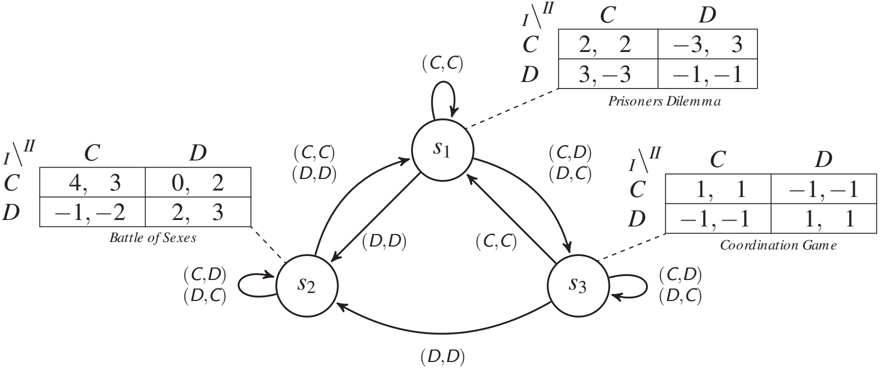
Expressing properties in QATL*: more examples



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

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

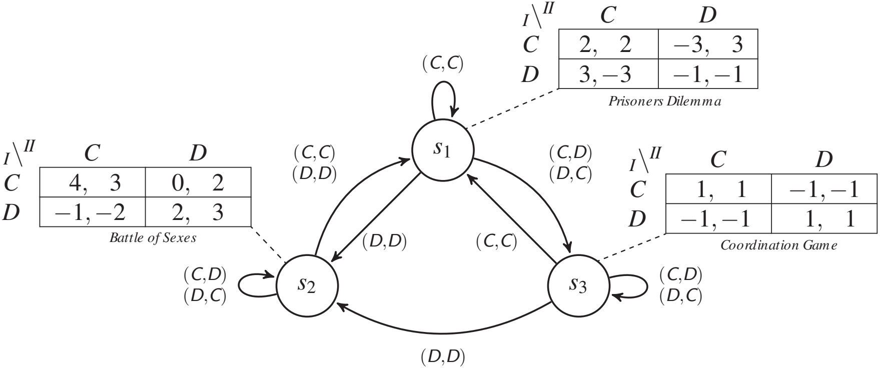
Expressing properties in QATL*: more examples



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

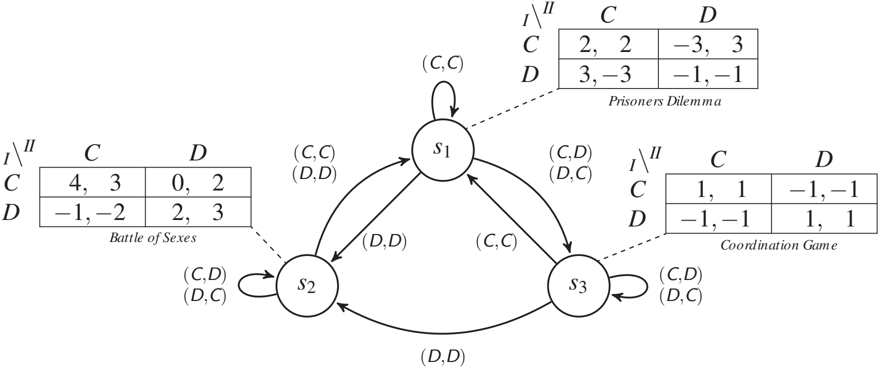
Expressing properties in QATL*: more examples



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

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

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

Some undecidability results about QATL*

Some undecidability results about QATL*

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

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 (2-counter automaton) A a finite 2-player GCGMP $\mathfrak{M}^A$ 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.

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 (2-counter automaton) A a finite 2-player GCGMP $\mathfrak{M}^A$ 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.*

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

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 (2-counter automaton) A a finite 2-player GCGMP $\mathfrak{M}^A$ 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.

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

1. Two players, no arithmetic constraints in the formula.

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 (2-counter automaton) A a finite 2-player GCGMP $\mathfrak{M}^A$ 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.*

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

1. Two players, no arithmetic constraints in the formula.
2. Two players, state-based guards.

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 (2-counter automaton) A a finite 2-player GCGMP $\mathfrak{M}^A$ 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.*

Thm Model checking in the logic QATL* is undecidable, even for the fragment with no nested cooperation modalities, where $\mathcal{S}^p = \mathsf{S}^{mem}$ and $\mathcal{S}^o = \mathsf{S}^{pos}$, 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*

Some decidability results and conjectures about QATL*

Thm: MC in the logic QATL* is decidable in the following cases:

Some decidability results and conjectures about QATL*

Thm: MC in the logic QATL* is decidable in the following cases:

1. Many players, all executing bounded memory effective strategies.

Some decidability results and conjectures about QATL*

Thm: MC in the logic QATL* is decidable in the following cases:

1. Many players, all executing bounded memory effective strategies.
2. Two-player turn-based GCGMPs, for the fragment with formulae involving only player 1's accumulated utility.

Some decidability results and conjectures about QATL*

Thm: MC in the logic QATL* is decidable in the following cases:

1. Many players, all executing bounded memory effective strategies.
2. Two-player turn-based GCGMPs, for the fragment with formulae involving only player 1's accumulated utility.

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

Some decidability results and conjectures about QATL*

Thm: MC in the logic QATL* is decidable in the following cases:

1. Many players, all executing bounded memory effective strategies.
2. Two-player turn-based GCGMPs, for the fragment with formulae involving only player 1's accumulated utility.

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

1. Two players and non-negative payoffs.

Some decidability results and conjectures about QATL*

Thm: MC in the logic QATL* is decidable in the following cases:

1. Many players, all executing bounded memory effective strategies.
2. Two-player turn-based GCGMPs, for the fragment with formulae involving only player 1's accumulated utility.

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 the quantitative atomic formulae to only allow comparisons between players' payoffs and constants, i.e. of the type $v_i \circ c$ but not $v_i \circ v_j$, where $\circ \in \{>, =, <\}$.

Concluding remarks

We have proposed a logical framework combining qualitative with quantitative reasoning in multi-payer games.

Concluding remarks

We have proposed a logical framework combining qualitative with quantitative reasoning in multi-payer games.

▷ Three perspectives of research agenda:

Concluding remarks

We have proposed a logical framework combining qualitative with quantitative reasoning in multi-payer games.

▷ Three perspectives of research agenda:

- Logic: Expressiveness, formal reasoning, deduction.

Concluding remarks

We have proposed a logical framework combining qualitative with quantitative reasoning in multi-payer games.

▷ Three perspectives of research agenda:

- Logic: Expressiveness, formal reasoning, deduction.
- Computation: decidability, algorithms and complexity for model checking and synthesis.

Concluding remarks

We have proposed a logical framework combining qualitative with quantitative reasoning in multi-payer games.

▷ Three perspectives of research agenda:

- Logic: Expressiveness, formal reasoning, deduction.
- Computation: decidability, algorithms and complexity for model checking and synthesis.
- Game theory: solution concepts, equilibria, extending results from repeated games (e.g., folk theorems), etc.

Concluding remarks

We have proposed a logical framework combining qualitative with quantitative reasoning in multi-payer games.

▷ Three perspectives of research agenda:

- Logic: Expressiveness, formal reasoning, deduction.
- Computation: decidability, algorithms and complexity for model checking and synthesis.
- Game theory: solution concepts, equilibria, extending results from repeated games (e.g., folk theorems), etc.

▷ Many still unexplored directions, including:

Concluding remarks

We have proposed a logical framework combining qualitative with quantitative reasoning in multi-payer games.

▷ Three perspectives of research agenda:

- Logic: Expressiveness, formal reasoning, deduction.
- Computation: decidability, algorithms and complexity for model checking and synthesis.
- Game theory: solution concepts, equilibria, extending results from repeated games (e.g., folk theorems), etc.

▷ Many still unexplored directions, including:

- games with imperfect information,

Concluding remarks

We have proposed a logical framework combining qualitative with quantitative reasoning in multi-payer games.

▷ Three perspectives of research agenda:

- Logic: Expressiveness, formal reasoning, deduction.
- Computation: decidability, algorithms and complexity for model checking and synthesis.
- Game theory: solution concepts, equilibria, extending results from repeated games (e.g., folk theorems), etc.

▷ Many still unexplored directions, including:

- games with imperfect information,
- satisfiability testing and model synthesis,

Concluding remarks

We have proposed a logical framework combining qualitative with quantitative reasoning in multi-payer games.

▷ Three perspectives of research agenda:

- Logic: Expressiveness, formal reasoning, deduction.
- Computation: decidability, algorithms and complexity for model checking and synthesis.
- Game theory: solution concepts, equilibria, extending results from repeated games (e.g., folk theorems), etc.

▷ Many still unexplored directions, including:

- games with imperfect information,
- satisfiability testing and model synthesis,
- stochastic games with probabilistic strategies, etc.

Concluding remarks

We have proposed a logical framework combining qualitative with quantitative reasoning in multi-payer games.

▷ Three perspectives of research agenda:

- Logic: Expressiveness, formal reasoning, deduction.
- Computation: decidability, algorithms and complexity for model checking and synthesis.
- Game theory: solution concepts, equilibria, extending results from repeated games (e.g., folk theorems), etc.

▷ Many still unexplored directions, including:

- games with imperfect information,
- satisfiability testing and model synthesis,
- stochastic games with probabilistic strategies, etc.

The End