Formal Aspects of Strategic Reasoning and Game Playing

Strategic Reasoning with Quantitative Goals

Munyque Mittelmann[1], Aniello Murano[1], Laurent Perrussel[2]

[1] University of Naples Federico II

[2] University Toulouse Capitole - IRIT

munyque.mittelmann@unina.it

Content

Content

Content

Strategic Reasoning with Quantitative Goals

- Boolean verification
  - Either the system satisfies a logic specification or it does not
  - $\mathrm{cleanRiver}$ is either true or false in a given state
- Quantitative verification
  - Assessing the *quality* of Multi-Agent Systems (MAS)
  - Levels of *quality* represented with weights
  - $\mathrm{cleanRiver}$ may be *partially* true in a state

Quantitative Logics for MAS

Logics with quantitative satisfaction

- Goals are expressed as a fuzzy temporal constraint:
  - Boolean satisfaction $\rightsquigarrow$ quantitative satisfaction;
  - Specification language $\rightsquigarrow$ LTL[$\mathcal{F}$][1], ATL*[$\mathcal{F}$]/ATL[$\mathcal{F}$][2], SL[$\mathcal{F}$][3]
  - System model $\rightsquigarrow$ Weighted Game Structure.

[1] Almagor, Boker, and Kupferman (2016). "Formally Reasoning about Quality". In: *Journal of the ACM*

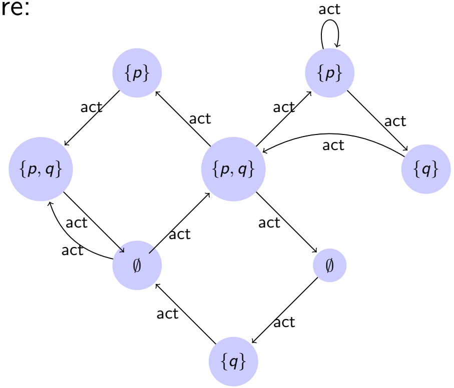
[2] Jamroga, Mittelmann, Murano, and Perelli (2024). "Playing Quantitative Games Against an Authority: On the Module Checking Problem". In: *AAMAS 2024*

[3] Bouyer, Kupferman, Markey, Maubert, Murano, and Perelli (2019). "Reasoning about Quality and Fuzziness of Strategic Behaviours". In: *IJCAI*

Concurrent Game Structures (CGS)

A CGS is a tuple $\mathcal{G} = (\mathrm{Ap}, \mathrm{Ag}, \mathrm{Ac}, V, d, o, \ell)$, where:

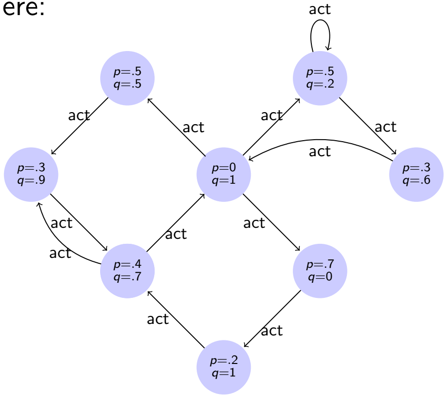
- Ap — *propositions* (relevant facts)
- Ag — *agents*
- Ac — agents' *actions*
- V — *states*
- $d : \mathrm{Ag} \times V \to 2^{\mathrm{Ac}}$ — *available actions*
- $o : V \times \mathrm{Ac}^{\mathrm{Ag}} \to V$ — *transition function*
- $\ell : V \to 2^{\mathrm{Ap}}$ — *labelling function*



Weighted CGS (wCGS)

A wCGS is a tuple $\mathcal{G} = (\mathrm{Ap}, \mathrm{Ag}, \mathrm{Ac}, V, d, o, \ell)$, where:

- Ap — *propositions* (relevant facts)
- Ag — *agents*
- Ac — agents' *actions*
- V — *states*
- $d : \mathrm{Ag} \times V \to 2^{\mathrm{Ac}}$ — *available actions*
- $o : V \times \mathrm{Ac}^{\mathrm{Ag}} \to V$ — *transition function*
- $\ell : V \times \mathrm{Ap} \to [0,1]$ — *weight function*



Weight function instead of labeling function to model degrees of truth. (fuzzy satisfaction)

Quantitative logics for MAS

The logics are parametrized over a set of functions $\mathcal{F}$ [4]:

$$f : [0,1]^n \to [0,1] \in \mathcal{F}$$

Example:

- $x \vee y := \max(x, y)$ (disjunction)
- $x \wedge y := \min(x, y)$ (conjunction)
- $\neg x := 1 - x$ (negation)

We assume that some standard functions belong to $\mathcal{F}$: $\leq$ (Boolean), $=$ (Boolean), bounded sum, etc.

[4] We assume the functions in $\mathcal{F}$ to be computable in polynomial time

Quantitative ATL* and ATL

ATL$^*[\mathcal{F}]$ Syntax

$$\varphi ::= p \mid f[\varphi, ..., \varphi] \mid \mathbf{X}\varphi \mid \varphi\mathbf{U}\varphi \mid \varphi\mathbf{R}\varphi \mid \langle\langle A\rangle\rangle\varphi$$

where p is a proposition, A is a coalition, and $f \in \mathcal{F}$

ATL$[\mathcal{F}]$ Syntax (no temporal nesting allowed)

$$\varphi ::= p \mid f[\varphi, ..., \varphi] \mid \langle\langle A\rangle\rangle\mathbf{X}\varphi \mid \langle\langle A\rangle\rangle\varphi\mathbf{U}\varphi \mid \langle\langle A\rangle\rangle\varphi\mathbf{R}\varphi$$

ATL$^*[\mathcal{F}]$ and ATL$[\mathcal{F}]$ Semantics

- "$f[\varphi, ..., \varphi]$" - compute the function over the satisfaction values of its inputs
- "$\langle\langle A\rangle\rangle\varphi$" - coalition A maximizes the satisfaction value of φ
- Abbreviations: $[\![A]\!]\varphi := \neg\langle\langle A\rangle\rangle\neg\varphi$ $\quad$ $\mathbf{F}\varphi := \top\mathbf{U}\varphi$ $\quad$ $\mathbf{G}\varphi := \bot\mathbf{R}\varphi$

Relation with Boolean ATL*

Can we capture ATL* *with* ATL$^*[\mathcal{F}]$*?*

Relation with Boolean ATL*

Can we capture ATL* *with* ATL$^*[\mathcal{F}]$*?*

Yes, when atomic propositions can only take values 0 and 1, and $\mathcal{F}$ contains only negation and disjunction.

Example: Drone battle

Two carrier drones *a* and *b* cooperate trying to bring an artifact to a rescue point and keep it away from the "villain" drone *v*:

- $\mathrm{rescued}$ denotes whether the artifact is at the rescue point
- *dis* computes the distance between two (normalized) positions
- pos_x denote the position of drone *x*
- Level of safety: minimum distance between any carrier and the villain

$$\varphi_{\mathrm{safe}} := \langle\langle a, b\rangle\rangle\ \mathit{min}[\mathit{dis}[\mathrm{pos}_a, \mathrm{pos}_v], \mathit{dis}[\mathrm{pos}_b, \mathrm{pos}_v]]\ \mathbf{U}\ \mathrm{rescued}$$

What does the formula φ_{safe} *captures?*

Example: Drone battle

Two carrier drones a and b cooperate trying to bring an artifact to a rescue point and keep it away from the "villain" drone v:

- $\mathrm{rescued}$ denotes whether the artifact is at the rescue point
- dis computes the distance between two (normalized) positions
- pos_x denote the position of drone x
- Level of safety: minimum distance between any carrier and the villain

$$\varphi_{\mathrm{safe}} := \langle\langle a, b\rangle\rangle\ min[dis[\mathrm{pos}_a, \mathrm{pos}_v], dis[\mathrm{pos}_b, \mathrm{pos}_v]]\ \mathbf{U}\ \mathrm{rescued}$$

What does the formula φ_{safe} captures?

Carriers a and b best-performing joint strategy to keep the villain as far as possible from the carriers, until the artifact is rescued.

Example: Drone battle

Two carrier drones *a* and *b* cooperate trying to bring an artifact to a rescue point and keep it away from the "villain" drone *v*:

- $\mathrm{rescued}$ denotes whether the artifact is at the rescue point
- *dis* computes the distance between two (normalized) positions
- pos_x denote the position of drone *x*
- Level of safety: minimum distance between any carrier and the villain

$$\varphi_{\text{safe}} := \langle\langle a, b\rangle\rangle\ min[dis[\mathrm{pos}_a, \mathrm{pos}_v], dis[\mathrm{pos}_b, \mathrm{pos}_v]]\ \mathbf{U}\ \mathrm{rescued}$$

What does the formula φ_{safe} captures?

Carriers *a* and *b* best-performing joint strategy to keep the villain as far as possible from the carriers, until the artifact is rescued.

What if the artifact is never rescued?

Example: Drone battle

Two carrier drones a and b cooperate trying to bring an artifact to a rescue point and keep it away from the "villain" drone v:

- $\mathrm{rescued}$ denotes whether the artifact is at the rescue point
- dis computes the distance between two (normalized) positions
- pos_x denote the position of drone x
- Level of safety: minimum distance between any carrier and the villain

$$\varphi_{\mathrm{safe}} := \langle\langle a, b \rangle\rangle \; min[dis[\mathrm{pos}_a, \mathrm{pos}_v], dis[\mathrm{pos}_b, \mathrm{pos}_v]] \; \mathbf{U} \; \mathrm{rescued}$$

What does the formula φ_{safe} captures?

Carriers a and b best-performing joint strategy to keep the villain as far as possible from the carriers, until the artifact is rescued.

What if the artifact is never rescued?

The satisfaction value of φ_{safe} would be 0.

Example: Drone battle (cont.)

Can we express that there is a strategy for the drone a such that for all strategies of the villain (v), the drone b has a response strategy?

Example: Drone battle (cont.)

Can we express that there is a strategy for the drone a such that for all strategies of the villain (v), the drone b has a response strategy? ✗

No, we cannot capture alternation of strategy quantification (each strategic quantifier resets previously assigned strategies).

Example: Drone battle (cont.)

Can we express that there is a strategy for the drone a such that for all strategies of the villain (v), the drone b has a response strategy?

No, we cannot capture alternation of strategy quantification (each strategic quantifier resets previously assigned strategies).

We need a more expressive logic...

Quantitative SL

SL[$\mathcal{F}$] Syntax

$$\varphi ::= p \mid \exists s.\varphi \mid (a, s)\varphi \mid f[\varphi, ..., \varphi] \mid \mathbf{X}\varphi \mid \varphi \mathbf{U} \varphi$$

where p is a proposition, s is a variable, a is an agent, and $f \in \mathcal{F}$

SL[$\mathcal{F}$] Semantics

- Defined over assignments of strategies to variables and agents
- "$\exists s.\varphi$" - the maximal satisfaction value of φ for the possible assignments of strategy to s
- "$(a, s)\varphi$" - the satisfaction value of φ when agent a is assigned to the str. assigned to s
- Abbreviations: $\forall s.\varphi := \neg\exists s.\neg\varphi$ $\quad \mathbf{F}\varphi := \top\mathbf{U}\varphi$ $\quad \mathbf{G}\varphi := \neg\mathbf{F}\neg\varphi$ $\quad \varphi\mathbf{R}\psi := \neg(\neg\varphi\mathbf{U}\neg\psi)$
- We call LTL[$\mathcal{F}$] the fragment without strategic operators and bindings

Example: Drone battle (cont.)

There is a strategy for drone *a* such that for all strategies of the villain *v*, *b* has a response strategy to keep the villain as far as possible, until the artifact is rescued:

$$\exists s.\forall t.\exists s'.(a, s)(v, t)(b, s')\ \mathit{min}[\mathit{dis}[\mathrm{pos}_a, \mathrm{pos}_v], \mathit{dis}[\mathrm{pos}_b, \mathrm{pos}_v]]\ \mathbf{U}\ \mathrm{rescued}$$

Example: Nash equilibrium

Assume each agent a has an LTL[$\mathcal{F}$] goal φ_a.
Let $\boldsymbol{s} = (s_a)_{a \in \mathrm{Ag}}$ denote a strategy profile.
Ag_{-a} denotes the set of agents without a.
$\boldsymbol{s_{-a}}$ denotes the strategies of Ag_{-a} in the profile $\boldsymbol{s}$.

Nash equilibrium (NE)

The strategy profile $\boldsymbol{s}$ is a *Nash equilibrium* if for each agent a, no alternative strategy t for a leads to a better utility than her strategy s_a (while all other agent' strategies play $\boldsymbol{s_{-a}}$).

Example: Nash equilibrium

Assume each agent a has an LTL[$\mathcal{F}$] goal φ_a.
Let $\boldsymbol{s} = (s_a)_{a \in \mathrm{Ag}}$ denote a strategy profile.
Ag_{-a} denotes the set of agents without a.
$\boldsymbol{s_{-a}}$ denotes the strategies of Ag_{-a} in the profile $\boldsymbol{s}$.

Nash equilibrium (NE)

The strategy profile $\boldsymbol{s}$ is a *Nash equilibrium* if for each agent a, no alternative strategy t for a leads to a better utility than her strategy s_a (while all other agent' strategies play $\boldsymbol{s_{-a}}$).

How can we express whether $\boldsymbol{s}$ *is a NE in* SL[$\mathcal{F}$]*?*

Example: Nash equilibrium

Assume each agent a has an LTL[$\mathcal{F}$] goal φ_a.
Let $\boldsymbol{s} = (s_a)_{a \in \mathrm{Ag}}$ denote a strategy profile.
Ag_{-a} denotes the set of agents without a.
$\boldsymbol{s_{-a}}$ denotes the strategies of Ag_{-a} in the profile $\boldsymbol{s}$.

Nash equilibrium (NE)

The strategy profile $\boldsymbol{s}$ is a *Nash equilibrium* if for each agent a, no alternative strategy t for a leads to a better utility than her strategy s_a (while all other agent' strategies play $\boldsymbol{s_{-a}}$).

How can we express whether $\boldsymbol{s}$ *is a NE in* SL[$\mathcal{F}$]*?*

$$\mathrm{NE}(\boldsymbol{s}) \stackrel{\text{def}}{=} \bigwedge_{a \in \mathrm{Ag}} \forall t.\ \left[(\mathrm{Ag}_{-a}, \boldsymbol{s}_{-a})(a, t)\varphi_a \leq (\mathrm{Ag}, \boldsymbol{s})\varphi_a\right]$$

Example: Nash equilibrium (cont)

We can also measure *how much* agent a can benefit from a selfish deviation using formula:

$$\exists t.\mathit{diff}\left[(\mathrm{Ag}_{-a}, \boldsymbol{s}_{-a})(a, t)\varphi_a, (\mathrm{Ag}, \boldsymbol{s})\varphi_a\right]$$

where $\mathit{diff}(x, y) = \mathit{max}\{0, x - y\}$.

Content

Model checking

Model checking problem

Given an SL[$\mathcal{F}$] (similarly ATL*[$\mathcal{F}$] or ATL[$\mathcal{F}$]) formula φ, a wCGS $\mathcal{G}$, a state v, and a predicate $P \subseteq (0, 1]$, decide whether the satisfaction value of φ in v is a subset or equal to P, denoted

$$[\![\varphi]\!]^{\mathcal{G}}(v) \subseteq P$$

The predicate can be the set of values above a threshold $\epsilon \in (0, 1]$:
Decide whether $[\![\varphi]\!]^{\mathcal{G}}(v) \geq \epsilon$.

Complexity of Model Checking

Using automata-theoretic approaches:

Theorem 1 (Bouyer et al., 2019)

Model-checking SL[$\mathcal{F}$] *in (k+1)* EXPTIME
(where k is the number of alternations of strategic operators)

Theorem 2 (Jamroga et al., 2024)

Model-checking ATL*[$\mathcal{F}$] 2EXPTIME-*complete*

Complexity of Model Checking

Algorithmic solution:

Theorem 3 (Jamroga et al., 2024)

Model-checking ATL[$\mathcal{F}$] — PTIME-*complete*

Theorem 4 (Maubert et al., 2021)

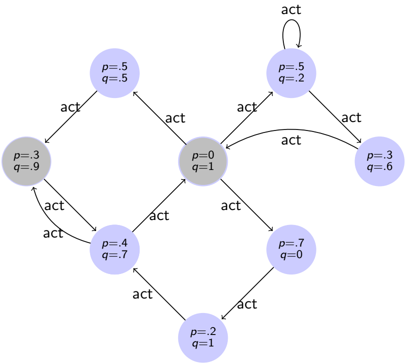
Model checking SL[$\mathcal{F}$] *with memoryless agents* — PSPACE-*complete*

Content

Weighted Module

Weighted Module is a special wCGS $\mathcal{G} = (\mathrm{Ap}, \mathrm{Ag}, \mathrm{Ac}, V, d, o, \ell)$:

- Ap — *propositions* (relevant facts)
- $\mathrm{Ag} \cup \{\mathrm{e}\}$ — *agents* and *environment*
- Ac — *actions*
- V — *states*
- $V_{\mathrm{e}} \subseteq V$ — environment's states
- $d : \mathrm{Ag} \times V \to 2^{\mathrm{Ac}}$ — *available actions*
- $o : V \times \mathrm{Ac}^{\mathrm{Ag}} \to V$ — *transition function*
- $\ell : V \times \mathrm{Ap} \to [0,1]$ — *weight function*



Environment states (gray) under the control of an "environmental" authority, who shapes the game by selecting possible successors at each iteration.

Module Checking

For a given weighted module $\mathcal{G}$:

- $\mathcal{T} \in \mathrm{exec}(\mathcal{G})$ is a possible wCGS resulting from the choices of e in $\mathcal{G}$.

Given an ATL$^*[\mathcal{F}]$ formula φ, a module $\mathcal{G}$, a position v:

- $[\![\varphi]\!]_r^{\mathcal{G}}(v) = \{[\![\varphi]\!]^{\mathcal{T}}(v) \mid \mathcal{T} \in \mathrm{exec}(\mathcal{G})\}$ all possible values in v according to $\mathcal{T}$

Definition 5 (Module Checking)

Deciding whether $[\![\varphi]\!]_r^{\mathcal{G}}(v) \subseteq P$, for a given predicate $P \subseteq [0, 1]$.

Complexity of Module Checking

Automata-theoretic approach

Theorem 6 (Jamroga et al., 2024)

- *Module-checking* ATL$^*[\mathcal{F}]$ *3*EXPTIME*-complete*
- *Module-checking* ATL$[\mathcal{F}]$ EXPTIME*-complete*

Relation with Boolean Module Checking and Model Checking

- $\mathrm{ATL}^*[\mathcal{F}]$ module checking is not subsumed by ATL^* module checking over weighted modules
- $\mathrm{ATL}^*[\mathcal{F}]$ *module* checking is not subsumed by $\mathrm{ATL}^*[\mathcal{F}]$ *model* checking.

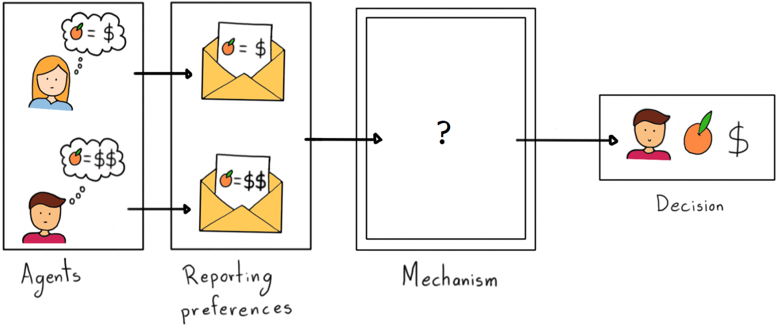
Contents

- Quantitative extensions of SL, ATL*, and ATL
- Model and module checking problems have the same computational complexity as the corresponding logics with Boolean semantics
- MAS with quantitative goals: application to mechanism design

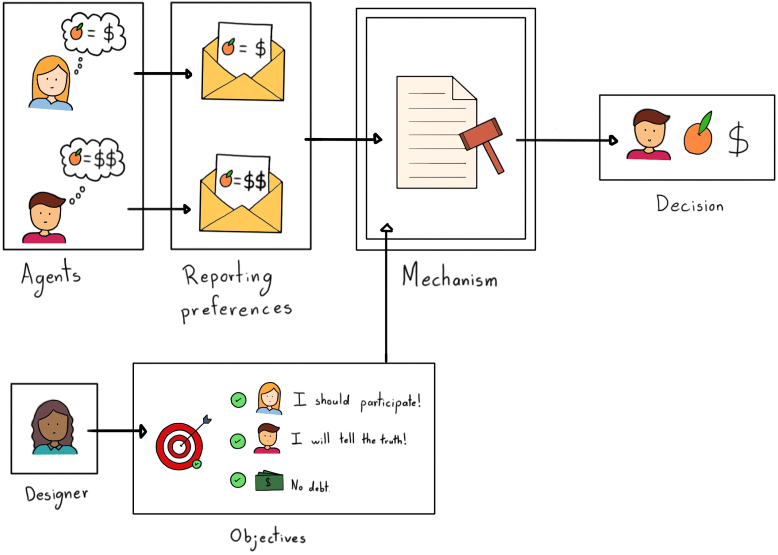
Content

Content

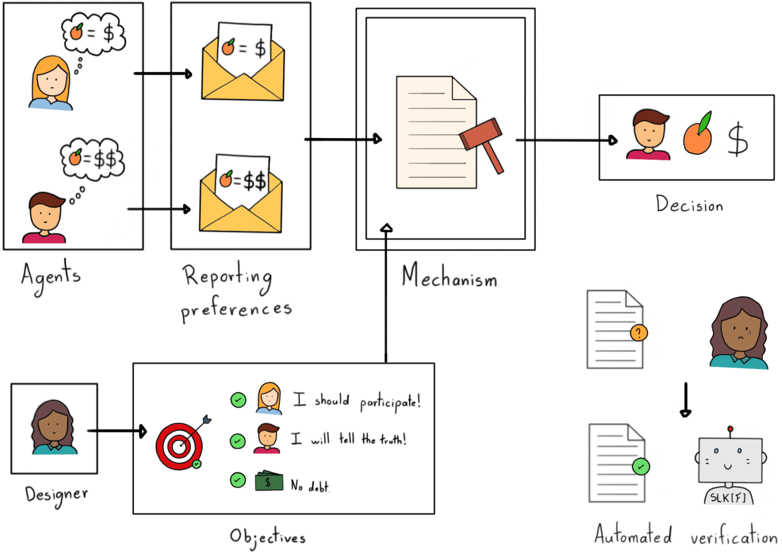
Mechanism Design



Mechanism Design



Mechanism Design



Motivation

- Preference aggregation problems
  - Auctions, elections, fair division protocols, etc
- Logic-based approach: verification[5] and synthesis of mechanisms[6]
  - We use the weights $[-1, 1]$ for convenience

[5] Maubert, Mittelmann, Murano, and Perrussel (2021). "Strategic Reasoning in Automated Mechanism Design". In: *KR 2021*.

[6] Mittelmann, Maubert, Murano, and Perrussel (2022). "Automated Synthesis of Mechanisms". In: *IJCAI 2022*.

Mechanisms

- Alternatives Alt
  - $\{(buyer_{Bob}, pays_k), (buyer_{Ann}, pays_k) : 0 \leq k \leq 10\}$ (selling an item)
  - $\{(Ann, Bob), (Ann, Carol), (Bob, Carol)\}$ (choosing two representatives)
  - $\{(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}), (\frac{1}{2}, \frac{1}{2}, 0), (1, 0, 0), ...\}$ (splitting a resource)

Mechanisms

- Alternatives Alt
  - $\{(buyer_{Bob}, pays_k), (buyer_{Ann}, pays_k) : 0 \leq k \leq 10\}$ (selling an item)
  - $\{(Ann, Bob), (Ann, Carol), (Bob, Carol)\}$ (choosing two representatives)
  - $\{(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}), (\frac{1}{2}, \frac{1}{2}, 0), (1, 0, 0), \ldots\}$ (splitting a resource)
- Many mechanisms describe monetary transfers, thus an alternative is in the form $(x, (\mathrm{p}_a)_{a \in \mathrm{Ag}})$ where $x \in X$ is a choice from a finite set of choices, and p_a is the payment for agent a.
  E.g., $x = buyer_{Bob}$, $\mathrm{p}_{Bob} = 10$, $\mathrm{p}_{Ann} = 0$

Mechanisms for social choice

- Agent's type (preference) $\theta_a \in \Theta_a$
- Valuation function $\mathrm{v}_{ag} : X \times \Theta_a \to \mathbb{R}$
- Utility function u_{ag} : Alt $\times\, \Theta_a \to \mathbb{R}$
  - E.g., Possible types in a single-item auction $\Theta_{Bob} = \{0, ..., 10\}$
  - $\theta_{Bob} = 2$ means Bob value to the item is 2 euros

Mechanisms for social choice

- Agent's type (preference) $\theta_a \in \Theta_a$
- Valuation function $\mathrm{v}_{ag} : X \times \Theta_a \rightarrow \mathbb{R}$
- Utility function u_{ag} : Alt $\times \Theta_a \rightarrow \mathbb{R}$
  - E.g., Possible types in a single-item auction $\Theta_{Bob} = \{0, ..., 10\}$
  - $\theta_{Bob} = 2$ means Bob value to the item is 2 euros
  - The valuation of Bob is

$$\mathrm{v}_{Bob}(buyer_{Bob}, \theta_{Bob}) = \theta_{Bob}$$

$$\mathrm{v}_{Bob}(buyer_{Ann}, \theta_{Bob}) = 0$$

  - The (quasi-linear) utility is

$$u_{Bob}((buyer_{Bob}, (\mathrm{p}_{Bob}, \mathrm{p}_{Ann})), \theta_{Bob}) = \mathrm{v}_{Bob}(buyer_{Bob}, \theta_{Bob}) - \mathrm{p}_{Bob}$$

$$u_{Bob}((buyer_{Bob}, (5, 0)), 2) = 2 - 5 = -3$$

Mechanisms for social choice

- Types $\boldsymbol{\Theta} = \prod_{a \in \mathsf{Ag}} \Theta_a$
- Strategies $S = \prod_{a \in \mathsf{Ag}} s_a$
- Mechanism $\mathcal{M} : S \to \mathsf{Alt}$
  - English auction: the agents increase the price until there are no other buyers interested
  - Dutch auction: the price decreases until one agent accepts to buy

Example: wCGS representing the Dutch auction

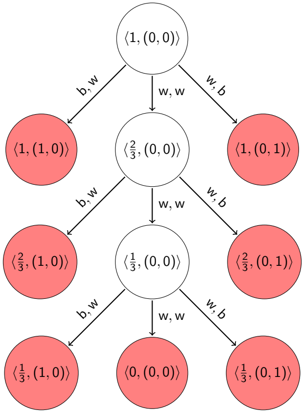


Figure 2: Part of the mechanism for the Dutch auction with two agents and decrement $\text{dec} = \frac{1}{3}$.

Mechanisms for social choice

Evaluation of a mechanism with rational agents: solution concepts

Mechanisms for social choice

Evaluation of a mechanism with rational agents: solution concepts

Example of properties:

- Budget-balance
- Strategyproof
- Individual rationality
- Efficiency
- ...

Solution concepts

- Nash equilibrium (NE): considers (unilateral) deviations of individual agents
- Dominant strategy equilibrium (DSE): the strategy associated with each agent weakly maximizes her utility, for all possible strategies of other agents
- m-resilient equilibrium (RE_m): considers deviations by coalitions of agents rather than individuals, it tolerates deviations of up to m agents

Mechanism Properties

Individual Rationality (IR):

$$\mathrm{IR} \stackrel{\text{def}}{=} \bigwedge_{a \in \mathrm{Ag}} 0 \leq \mathrm{util}_a$$

The Dutch auction is IR

Mechanism Properties

Strong Budget Balance (SBB):

$$\text{SBB} \stackrel{\text{def}}{=} 0 = \sum_{a \in \text{Ag}} \text{pay}_a$$

Weak Budget Balance (WBB):

$$\text{WBB} \stackrel{\text{def}}{=} 0 \leq \sum_{a \in \text{Ag}} \text{pay}_a$$

The Dutch auction is WBB and not SBB

Mechanism Properties

Strategyproofness (SP)
Let $\hat{\theta}_a$ be the truth-revealing strategy for a

$$\text{DSE}(\boldsymbol{s}) \text{ where } \mathcal{A}(s_a) = \hat{\theta}_a \text{ for each } a$$

The Dutch auction is not SP

Mechanism Properties

Strategyproofness (SP)
Let $\hat{\theta}_a$ be the truth-revealing strategy for a

$$\mathrm{DSE}(\boldsymbol{s}) \text{ where } \mathcal{A}(s_a) = \hat{\theta}_a \text{ for each } a$$

The Dutch auction is not SP

Efficiency, Pareto optimality, ...

Model-checking SL[$\mathcal{F}$]

Model checking mechanism properties with SL[$\mathcal{F}$] when agents are strategic:
For a given property φ and solution concept ζ, we check

$$\exists \boldsymbol{\sigma}.[\zeta(\boldsymbol{\sigma}) \wedge (\mathrm{Ag}, \boldsymbol{\sigma})\varphi]$$

More complex mechanisms

By changing the specification language, we can also verify mechanisms with imperfect information [7] and probabilistic features [8]

[7] Maubert, Mittelmann, Murano, and Perrussel (2021). "Strategic Reasoning in Automated Mechanism Design". In: *KR 2021*

[8] Mittelmann, Maubert, Murano, and Perrussel (2023). "Formal Verification of Bayesian Mechanisms". In: *AAAI*

Synthesis of Mechanisms

- Creating mechanisms from a logical specification in SL[$\mathcal{F}$]
- Satisfiability of SL (thus, SL[$\mathcal{F}$]) is undecidable in general
- Decidable cases

Synthesis of Mechanisms

Given a finite set $\mathcal{V} \subset [-1, 1]$ such that $\{-1, 1\} \subseteq \mathcal{V}$, the $\mathcal{V}$-*satisfiability problem* for SL[$\mathcal{F}$] is the restriction of the satisfiability problem to $\mathcal{V}$-weighted wCGS.

Theorem 7 (Mittelmann, Maubert, et al., 2022)

The satisfiability of SL[$\mathcal{F}$] *is decidable in the following cases:*

- $wCGS$ *with bounded actions*
- *Turn-based* $wCGS$

- Algorithms for the satisfiability → return a satisfying wCGS when one exists (see Pnueli and Rosner, 1989)

Optimal mechanism synthesis

Algorithm 2 Optimal mechanism synthesis

Data: A SL[$\mathcal{F}$] specification Φ and a set of possible values for atomic propositions $\mathcal{V}$
Result: A wCGS $\mathcal{G}$ such that $[\![\Phi]\!]^{\mathcal{G}}$ is maximal
Compute $\widetilde{\mathsf{Val}}_{\Phi,\mathcal{V}}$ Let $\nu_1, ..., \nu_n$ be a decreasing enumeration of $\widetilde{\mathsf{Val}}_{\Phi,\mathcal{V}}$ **for** *i=1...n* **do**
  Solve $\mathcal{V}$- satisfiability for Φ and $\varepsilon = \nu_i$ **if** *there exists* $\mathcal{G}$ *such that* $[\![\Phi]\!]^{\mathcal{G}} \geq \nu_i$ **then**
    return $\mathcal{G}$
  end
end

Advantage

- Optimal mechanism synthesis
- Synthesis from auction rules (e.g. ADL-like[9]) and strategic requirements (e.g. strategyproofness)

[9] Mittelmann, Bouveret, and Perrussel (2022). "Representing and reasoning about auctions". In: *Autonomous Agents and Multi-Agent Systems* 36.1, p. 20.

Example Auction rules

- $\mathbf{AG}((\neg \mathsf{sold} \wedge \mathsf{price} + inc < 1) \rightarrow (\mathsf{price} + inc = \mathbf{X}\mathsf{price} \wedge \neg \mathbf{X}\mathsf{terminal}))$
- $\mathbf{AG}((\mathsf{sold} \vee \mathsf{price} + inc \geq 1) \rightarrow (\mathsf{price} = \mathbf{X}\mathsf{price} \wedge \mathbf{X}\mathsf{terminal}))$
- $\mathbf{AG}(\mathsf{choice} = \mathsf{wins}_a \leftrightarrow \mathsf{bid}_a \wedge \bigwedge_{b \neq a} \neg \mathsf{bid}_a)$
- $\mathbf{AG}(\bigwedge_{a \in \mathsf{Ag}}(\mathsf{choice} = \mathsf{wins}_a \rightarrow \mathsf{pay}_a = \mathsf{price}))$

Contents

- Logic-Based Mechanism Design
  - Verifying properties under strategic behaviour $\rightarrow$ MC SL[$\mathcal{F}$]-formulas
  - Generating mechanisms $\rightarrow$ synthesis from SL[$\mathcal{F}$]-formulas

Contents

- Logic-Based Mechanism Design
  - Verifying properties under strategic behaviour $\rightarrow$ MC SL[$\mathcal{F}$]-formulas
  - Generating mechanisms $\rightarrow$ synthesis from SL[$\mathcal{F}$]-formulas
- Correctness of the encoding for classic mechanism design

Contents

- Logic-Based Mechanism Design
  - Verifying properties under strategic behaviour $\rightarrow$ MC SL[$\mathcal{F}$]-formulas
  - Generating mechanisms $\rightarrow$ synthesis from SL[$\mathcal{F}$]-formulas
- Correctness of the encoding for classic mechanism design
- Logics for MAS allows us to go further

Content

Partially redesigning a system

- We can design new mechanisms with *nice* properties when agents act rationally...

Partially redesigning a system

- We can design new mechanisms with *nice* properties when agents act rationally...
- What if we already have a mechanism (or a *system*) but it doesn't have those properties?
- What if we cannot redesign it from scratch?

Existing environmental legislation fails to reach sustainability targets.
How can we change the *system* to address this issue?

- How can we change the system to satisfy desirable properties?

Partially redesigning a system

- We can design new mechanisms with *nice* properties when agents act rationally...
- What if we already have a mechanism (or a *system*) but it doesn't have those properties?
- What if we cannot redesign it from scratch?

Existing environmental legislation fails to reach sustainability targets.
How can we change the *system* to address this issue?

- How can we change the system to satisfy desirable properties?
  - norms, incentives, ...

Partially redesigning a system

How can we convince agents to act on behalf of the environment?

- Laws prohibiting the use of disposable plastic bags
- Taxes based on companies' pollution rates
- Subsidizing public transportation fees

- Norm design[10]
- Incentive design[11]

[10] Alechina, De Giacomo, Logan, and Perelli (2022). "Automatic Synthesis of Dynamic Norms for Multi-Agent Systems". In: *KR*.

[11] Hyland, Mittelmann, Murano, Perelli, and Wooldridge (2024). "Incentive Design for Rational Agents". In: *KR (to appear)*.

Partially redesigning a system

How can we convince agents to act on behalf of the environment?

- Laws prohibiting the use of disposable plastic bags
- Taxes based on companies' pollution rates
- Subsidizing public transportation fees

- Norm design[10]
- Incentive design[11]

[10] Alechina, De Giacomo, Logan, and Perelli (2022). "Automatic Synthesis of Dynamic Norms for Multi-Agent Systems". In: *KR*.

[11] Hyland, Mittelmann, Murano, Perelli, and Wooldridge (2024). "Incentive Design for Rational Agents". In: *KR (to appear)*.

Incentive Design

- Agents try to maximize their utilities, expressed with LTL[$\mathcal{F}$]-goals
- We want to impose *incentive schemes*
- Rationality is defined w.r.t. solution concepts

Incentive Scheme

It is a function, that assigns new weights to some (or all) atomic propositions
It can be either:

- Static (memoryless)
- Dynamic (history-based)

We assume that incentive schemes have a fixed level of granularity

Example - River

- Two companies share the usage of a river
- At each moment, the companies can either *discharge waste water* in the river or *treat the waste water* (at a cost)
  - If both firms discharge, the water quality deteriorates
  - If only one discharges, the quality is not affected
  - If both firms clean, the river quality improves

Example - River

- Two companies share the usage of a river
- At each moment, the companies can either *discharge waste water* in the river or *treat the waste water* (at a cost)
  - If both firms discharge, the water quality deteriorates
  - If only one discharges, the quality is not affected
  - If both firms clean, the river quality improves
- A regulator can impose taxes on each company
  - Company *a* goal: $\mathbf{G}(\mathrm{utility}_a - \mathrm{tax}_a)$
  - Taxes are initially zero $\rightarrow$ it motivates the companies to discharge wastewater in the river
  - Regulator goal: $\mathbf{G}(\mathrm{quality} \wedge \mathrm{fair})$.

Example - River

- With static incentive schemes:
  - The regulator can set the taxes so that at least one of the firms is worse off by discharging
  - If only one firm is taxed, it may be seen as unfair
  - If both firms are taxed, there may be an unnecessary loss of profits to both firms
- With dynamic incentive schemes:
  - The regulator can alternate between taxing the firms a sufficient amount for discharging, which is more fair and efficient

Computational Problems

Incentive Verification

Check if an incentive scheme guarantees that the goal φ is satisfied at least c

Incentive Synthesis

Find an incentive scheme, if it exists, that guarantees that the goal φ is satisfied at least c

Variants of the problems

- $\zeta \in \{\mathsf{DSE}, \mathsf{NE}, \mathsf{RE}_m\}$ denotes the solution concept
- E (similarly, A) indicates that the goal is satisfied in *some* (resp. *all*) equilibrium (fixed ζ)
- S (similarly, D) indicates that the incentive scheme is *static* (resp. *dynamic*)

Static Case

- For verification, we apply the static incentive scheme to the wCGS and then check the corresponding SL[$\mathcal{F}$] formulas:

$$\exists \boldsymbol{\sigma}.[\zeta(\boldsymbol{\sigma}) \wedge (\mathrm{Ag}, \boldsymbol{\sigma})\varphi]$$

$$\forall \boldsymbol{\sigma}.[\zeta(\boldsymbol{\sigma}) \rightarrow (\mathrm{Ag}, \boldsymbol{\sigma})\varphi]$$

- For synthesis, we non-deterministically guess an incentive scheme, then proceed with verification

Complexity - Static Case

Theorem 8 (Hyland et al., 2024)

For $\zeta \in \{\mathsf{DSE}, \mathsf{NE}, \mathsf{RE}_m\}$, $m \in \{1, \ldots, |\mathsf{Ag}|\}$, *the following problems are* 2EXPTIME*-complete:*

- ζ-S-E-INCENTIVE-VERIFICATION
- ζ-S-A-INCENTIVE-VERIFICATION
- ζ-S-E-INCENTIVE-SYNTHESIS
- ζ-S-A-INCENTIVE-SYNTHESIS

Dynamic Case

- We transform the original wCGS into a modified one:
  - We embed the incentive designer into the wCGS as an agent
  - Her actions correspond to the application of incentives
  - The new wCGS interleaves actions of the incentive designer and the other agents
  - This requires to *inflate* the runs of the wCGS and translate formulas
- Then, verification is done similarly to the static case (with adapted SL[$\mathcal{F}$] formulas)
- For synthesis, we also check the existence of an incentive designer strategy (which leads to an additional alternation in the ζ-D-A case)

Complexity - Dynamic Case

Theorem 9 (Hyland et al., 2024)

For $\zeta \in \{\mathsf{DSE}, \mathsf{NE}, \mathsf{RE}_m\}$, $m \in \{1, ..., |\mathsf{Ag}|\}$, the following problems are 2EXPTIME-complete

- ζ-D-E-INCENTIVE-VERIFICATION
- ζ-D-A-INCENTIVE-VERIFICATION
- ζ-D-E-INCENTIVE-SYNTHESIS

Finally, ζ-D-A-INCENTIVE-SYNTHESIS is in 3EXPTIME and is 2EXPTIME-hard.

Contents

- Incentive Design allows the partial redesign of games through incentives
- For the cases considered, the complexity of the problems is not harder than the corresponding Boolean rational verification problems (Abate et al., 2021)

Content

Content

Future discounting in MAS

- Satisfying the goal sooner $>$ after a long wait
- Temporal discounting operators alongside Linear Temporal Logic ($\mathsf{LTL}^{\mathsf{disc}}[\mathcal{D}]$)[12]
- $\mathsf{SL}^{\mathsf{disc}}[\mathcal{D}]$: Strategy Logic + future discounting[13]

[12] Almagor, Boker, and Kupferman (2014). "Discounting in LTL". In: *TACAS*.
[13] Mittelmann, Murano, and Perrussel (2023). "Discounting in Strategy Logic". In: *IJCAI*.

Strategy Logic with Discounting

- Enable to express:
  1. Strategic abilities of agents with discounted goals
  2. Solution concepts in discounting games
- Parametrized by a set of discounting functions $\mathcal{D}$:
  - Agents may be affected differently by how long it takes to achieve their goal

Strategy Logic with Discounting

A discounting function is a function that tends to zero and is non-increasing (e.g., $d(i) = \frac{1}{i+1}$)
We assume the functions in $\mathcal{D}$ are computable in polynomial time

$\mathsf{SL}^{\mathsf{disc}}[\mathcal{D}]$ syntax

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists s.\, \varphi \mid (a, s)\varphi \mid \mathbf{X}\varphi \mid \varphi\mathbf{U}\varphi \mid \varphi\mathbf{U}_d\varphi$$

where $p \in \mathrm{Ap}$, $s \in \mathrm{Ap}$, $a \in \mathrm{Ag}$, and $d \in \mathcal{D}$.

$\mathsf{SL}^{\mathsf{disc}}[\mathcal{D}]$ semantics

Quantified semantics defined over Concurrent Game Structures
Discounted-until $\varphi_1\mathbf{U}_d\varphi_2$ is weighted by how far in the future φ_1 and φ_2 occur

Relation with $\mathsf{LTL}^{\mathsf{disc}}[\mathcal{D}]$, SL and $\mathsf{SL}[\mathcal{F}]$

- $\mathsf{LTL}^{\mathsf{disc}}[\mathcal{D}] \subset \mathsf{SL}^{\mathsf{disc}}[\mathcal{D}]$
- $\mathsf{SL} \subset \mathsf{SL}^{\mathsf{disc}}[\mathcal{D}]$
- $\mathsf{SL}[\mathcal{F}]$ is interpreted over a different class of models
  Functions are independent of *how far* in the play they are being evaluated

Example - Secretary Problem

- $\mathbf{F}_d$ k-hired
- $\exists s \forall \boldsymbol{t}(a, s)(\mathrm{Ag}_{-a}, \boldsymbol{t})(\bigvee_{j \in C} \neg \mathrm{present}_j)\mathbf{U}_d$ k-hired

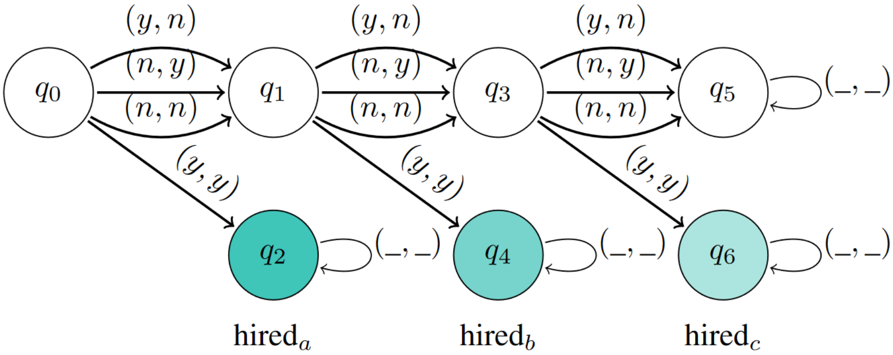


Figure 3: Instance of the secretary problem; the utility decreases the more time is taken to hire one.

Content

Model Checking $\mathsf{SL}^{\mathsf{disc}}[\mathcal{D}]$

Theorem 10 (Mittelmann, Murano, and Perrussel, 2023)

Model checking $\mathsf{SL}^{disc}[\mathcal{D}]$ *with memoryless agents* — PSPACE-*complete*

Theorem 11 (Mittelmann, Murano, and Perrussel, 2023)

Model checking $\mathsf{SL}^{disc}[\mathcal{D}]$ *with memoryfull agents* — $(k+1)$-EXPTIME
(when functions in $\mathcal{D}$ *are exponential-discounting, where* k *is the number of quantifiers alternations)*

Contents

- $\mathsf{SL}^{\mathsf{disc}}[\mathcal{D}]$: reasoning about temporal goals whose satisfaction value decays over time
- More expressive than SL
- Under certain restrictions, it has the same complexity as SL

Content

Directions for Future Work

- Synthesis from fragments of SL[$\mathcal{F}$]
- Partial synthesis
  - Incentives + Temporal Discounting
  - Fuzzy Norms
  - Finding minimal changes in the model
- SL[$\mathcal{F}$] + SL^{disc}[$\mathcal{D}$]?
- Extensions of model-checkers
  - MCMAS - `https://sail.doc.ic.ac.uk/software/mcmas/`
  - STV - `https://github.com/blackbat13/stv`
  - Vitamin - `https://arxiv.org/abs/2403.02170`

Thank you for following our course!

Formal Aspects of Strategic Reasoning and Game Playing

Strategic Reasoning with Quantitative Goals

Munyque Mittelmann[1], Aniello Murano[1], Laurent Perrussel[2]

[1] University of Naples Federico II

[2] University Toulouse Capitole - IRIT

munyque.mittelmann@unina.it

References I

- Abate, Gutierrez, Hammond, Harrenstein, Kwiatkowska, Najib, Perelli, Steeples, and Wooldridge (2021). “Rational verification: game-theoretic verification of multi-agent systems”. In: *Applied Intelligence* 51.9.
- Alechina, De Giacomo, Logan, and Perelli (2022). “Automatic Synthesis of Dynamic Norms for Multi-Agent Systems”. In: *KR*.
- Almagor, Boker, and Kupferman (2014). “Discounting in LTL”. In: *TACAS*.
- Almagor, Boker, and Kupferman (2016). “Formally Reasoning about Quality”. In: *Journal of the ACM*.
- Bouyer, Kupferman, Markey, Maubert, Murano, and Perelli (2019). “Reasoning about Quality and Fuzziness of Strategic Behaviours”. In: *IJCAI*.
- Hyland, Mittelmann, Murano, Perelli, and Wooldridge (2024). “Incentive Design for Rational Agents”. In: *KR (to appear)*.
- Jamroga, Mittelmann, Murano, and Perelli (2024). “Playing Quantitative Games Against an Authority: On the Module Checking Problem”. In: *AAMAS 2024*.
- Maubert, Mittelmann, Murano, and Perrussel (2021). “Strategic Reasoning in Automated Mechanism Design”. In: *KR 2021*.
- Mittelmann, Bouveret, and Perrussel (2022). “Representing and reasoning about auctions”. In: *Autonomous Agents and Multi-Agent Systems* 36.1, p. 20.

References II

- Mittelmann, Maubert, Murano, and Perrussel (2022). "Automated Synthesis of Mechanisms". In: *IJCAI 2022*.
- Mittelmann, Maubert, Murano, and Perrussel (2023). "Formal Verification of Bayesian Mechanisms". In: *AAAI*.
- Mittelmann, Murano, and Perrussel (2023). "Discounting in Strategy Logic". In: *IJCAI*.
- Pnueli and Rosner (1989). "On the Synthesis of a Reactive Module.". In: *Symposium on the Principles of Programming Languages (POPL 1989)*. New York: ACM, pp. 179–190.

This course is a part of the project *Strategic rEasoning for sociALly good mechanisms* (SEAL), which has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 101105549.

Funded by the European Union

This presentation uses several icons made by *Freepik* from Flaticon (www.flaticon.com).