Normative Mechanism Design

Mehdi Dastani

Utrecht University

This is a joint work with Nils Bulling

Background & Motivation

- The overall objectives of multi-agent systems can be ensured by coordinating the behaviors of individual agents through environment (E.g., Conference Management Systems, Train Station).
- Design multi-agent system environments in terms of norms and sanctions.
- Does a set of norms and sanctions implements specific social choice functions (designer's objectives) in specific equilibria? I.e., can specific behaviours be enforced by a normative environment program if agents follow their subjective preferences?

Environment: General Setting

- Agents perform synchronous actions
- Environment determines the outcome of actions
- Environment state evolves by agents' actions
- Actions are specified in terms of pre- and postconditions

Action Specification: *ActSpec*

{ reg(A) }	Submit (A,P)	{ paper(A,P) }
{ rev(R) , ass(R,P) }	Upload(R,P,V)	{ review(R,P,V) }

Some Notation

- Π is a set of atomic propositions
- $Q \subseteq \mathcal{P}(\Pi)$ is a set of states
- $q^I \in Q$
- $\mathbb{A}\mathrm{gt} = \{1, \ldots, k\}$ is a nonempty finite set of agents.
- $\vec{\alpha} = (\alpha_1, \ldots, \alpha_k)$ is a synchronous action (profile)
- *ActSpec* is a set of action specifications $(P, \vec{\alpha}, E)$

Environment Program Execution

- **action specifications + initial state** $= (ActSpec, q^I)$

Action Specification: *ActSpec*

{ reg(A) }	Submit (A,P)	{ paper(A,P) }
{ rev(R) , ass(R,P) }	Upload(R,P,V)	{ review(R,P,V) }

- Synchronous action $\vec{\alpha}$ updates state q

$$\frac{q \in Q \quad \& \quad q \models pre(\vec{\alpha}) \quad \& \quad q \oplus post(\vec{\alpha}) \in Q}{q \xrightarrow{\vec{\alpha}} q \oplus post(\vec{\alpha})}$$

$$q \oplus Y = \{\mathrm{p} \in \Pi \mid \mathrm{p} \in q \cup Y \text{ and } \neg\mathrm{p} \notin Y\}$$

Concurrent Game Structure

- The $(ActSpec, q^I)$-*generated* CGS is given by

$$\mathfrak{M} = \langle \mathbb{A}\mathrm{gt}, Q, Act, \hat{d}, d, o \rangle$$

  - $Act = \{\alpha_i \mid (P, (\alpha_1, \ldots, \alpha_k), E) \in ActSpec, 1 \leq i \leq k\}$
  - $\hat{d} : Q \rightarrow \mathcal{P}(Act^k)$ assigns applicable actions to states. An action is applicable if its precondition is satisfied and if the resulting state belongs to Q.
  - $d : \mathbb{A}\mathrm{gt} \times Q \rightarrow \mathcal{P}(Act)$ assigns at each state applicable individual actions to agents, i.e. $d(i, q) = \{\alpha_i \mid (\alpha_1, \ldots, \alpha_k) \in \hat{d}(q), 1 \leq i \leq k\}$
  - $o : Q \times Act^k \rightarrow Q$ is a partial (deterministic) transition function defined as $o(q, \vec{\alpha}) = q \oplus E$ for $(P, \vec{\alpha}, E)$

Normative Environment

- Environment specification in terms of norms and sanctions
- Norms represented by counts-as R^{cr} and sanction R^{sr} rules

Norm Set: $M = (R^{cr}, R^{sr})$

$$\begin{array}{ll} paper(A,P) \wedge pages(P) > 8 & \rightarrow_{cr} viol(A,P) \\ viol(A,P) & \rightarrow_{sr} pay(A,P,m) \end{array}$$

- Counts-as rules are interpreted as obligations

$$\phi \rightarrow_{cr} viol \quad \equiv \quad O\neg\phi$$

Normative Environment Program Execution

- **(action specifications + norm set + initial state)**

$$(ActSpec, \underbrace{(R^{cr}, R^{sr})}_{M}, q^I)$$

- State q restricted by norm set M: $q \restriction M = Cl^{sr}(Cl^{cr}(q))$
- $Q \restriction M = \{q \restriction M \mid q \in Q\}$

$$\frac{q \in Q \restriction M \;\&\; q \models pre(\vec{\alpha}) \;\&\; q' = (q \oplus post(\vec{\alpha})) \restriction M \in Q \restriction M}{q \xrightarrow{\vec{\alpha}} q'}$$

Concurrent Game Structure with Norms

- The $(ActSpec, M, q^I)$-*generated* CGS is given by

$$\mathfrak{M} \upharpoonright M = \langle \mathbb{A}\mathrm{gt}, Q \upharpoonright M, Act, \hat{d}, d, o \rangle$$

  - $Act = \{\alpha_i \mid (P, (\alpha_1, \ldots, \alpha_k), E) \in ActSpec, 1 \leq i \leq k\}$
  - $\hat{d} : Q \upharpoonright M \rightarrow \mathcal{P}(Act^k)$ assigns applicable actions to states. An action is applicable if its precondition is satisfied and if the resulting state belongs to $Q \upharpoonright M$.
  - $d : \mathbb{A}\mathrm{gt} \times Q \upharpoonright M \rightarrow \mathcal{P}(Act)$ assigns at each state applicable individual actions to agents, i.e. $d(i, q) = \{\alpha_i \mid (\alpha_1, \ldots, \alpha_k) \in \hat{d}(q), 1 \leq i \leq k\}$
  - $o : Q \upharpoonright M \times Act^k \rightarrow Q \upharpoonright M$ is a partial (deterministic) transition function defined as $o(q, \vec{\alpha}) = (q \oplus E) \upharpoonright M$ for $(P, \vec{\alpha}, E)$

Concurrent Game Structure with Norms

- The $(ActSpec, M, q^I)$-*generated* CGS is given by

$$\mathfrak{M} \restriction M = \langle \mathbb{A}\mathrm{gt}, Q \restriction M, Act, \hat{d}, d, o \rangle$$

  - $\gamma_i \in \mathcal{L}_{LTL}(\Pi_{\mathfrak{M}})$ is a preference of agent i.
  - $\vec{\gamma} = (\gamma_1, \ldots, \gamma_k)$ is a preference profile.
  - A social choice function f (designer objective) based on $(\mathfrak{M}, \mathcal{P}refs)$ is a mapping $f : \mathcal{P}refs \to \mathcal{L}_{LTL}(\Pi_{\mathfrak{M}})$.
  - Implementation question:

Does a norm set M $\mathcal{S}$-implements f over $\mathfrak{M}, q^I$, and $\mathcal{P}refs$?

How to relate solution concepts to $\mathfrak{M}$ and $\mathcal{P}refs$?

CGS $\rightsquigarrow$ strategic game

$\Gamma(\mathfrak{M}, \vec{\gamma}, q^I)$: **strategic game** associated with $\mathfrak{M}, \vec{\gamma}$

$\mathrm{CGS} \rightsquigarrow$ strategic game

$\Gamma(\mathfrak{M}, \vec{\gamma}, q^I)$: **strategic game** associated with $\mathfrak{M}, \vec{\gamma}$

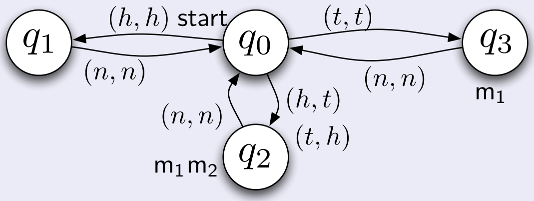
- $\mathfrak{M}$: generated CGS (by some action specifications)
- $\vec{\gamma} = (\gamma_1, \ldots, \gamma_k) \in \mathcal{P}refs$: preferences (**LTL**-formulae)
- q^I: initial state.

CGS $\rightsquigarrow$ strategic game

$\Gamma(\mathfrak{M}, \vec{\gamma}, q^I)$: **strategic game** associated with $\mathfrak{M}, \vec{\gamma}$

- $\mathfrak{M}$: generated CGS (by some action specifications)
- $\vec{\gamma} = (\gamma_1, \ldots, \gamma_k) \in \mathit{Prefs}$: preferences (**LTL**-formulae)
- q^I: initial state.

CGS



$\Gamma(\mathfrak{M}, \gamma^1, q_0)$

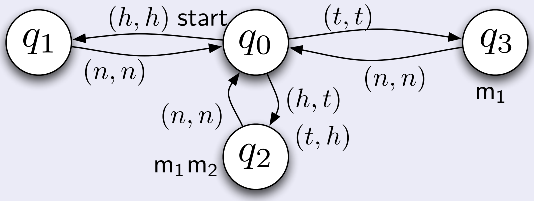
1/2	s_h	s_t
s_h		
s_t		

CGS $\rightsquigarrow$ strategic game

$\Gamma(\mathfrak{M}, \vec{\gamma}, q^I)$: **strategic game** associated with $\mathfrak{M}, \vec{\gamma}$

- $\mathfrak{M}$: generated CGS (by some action specifications)
- $\vec{\gamma} = (\gamma_1, \ldots, \gamma_k) \in \mathit{Prefs}$: preferences (**LTL**-formulae)
- q^I: initial state.

CGS



Preferences:

- player 1: $\Diamond(\mathrm{m}_1 \wedge \mathrm{m}_2)$
- player 2: $\Diamond \mathrm{m}_2$

$\Gamma(\mathfrak{M}, \gamma^1, q_0)$

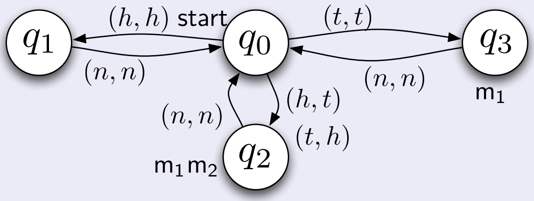
1/2	s_h	s_t
s_h		
s_t		

CGS $\rightsquigarrow$ strategic game

$\Gamma(\mathfrak{M}, \vec{\gamma}, q^I)$: **strategic game** associated with $\mathfrak{M}, \vec{\gamma}$

- $\mathfrak{M}$: generated CGS (by some action specifications)
- $\vec{\gamma} = (\gamma_1, \ldots, \gamma_k) \in \mathit{Prefs}$: preferences (**LTL**-formulae)
- q^I: initial state.

CGS



Preferences:

- player 1: $\Diamond(\mathrm{m}_1 \wedge \mathrm{m}_2)$
- player 2: $\Diamond\mathrm{m}_2$

$\Gamma(\mathfrak{M}, \gamma^1, q_0)$

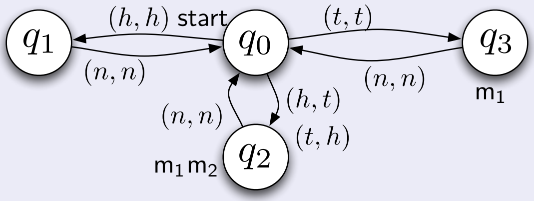
1/2	s_h	s_t
s_h	0, 0	
s_t		

CGS $\rightsquigarrow$ strategic game

$\Gamma(\mathfrak{M}, \vec{\gamma}, q^I)$: **strategic game** associated with $\mathfrak{M}, \vec{\gamma}$

- $\mathfrak{M}$: generated CGS (by some action specifications)
- $\vec{\gamma} = (\gamma_1, \ldots, \gamma_k) \in \mathit{Prefs}$: preferences (**LTL**-formulae)
- q^I: initial state.

CGS



Preferences:

- player 1: $\Diamond(m_1 \wedge m_2)$
- player 2: $\Diamond m_2$

$\Gamma(\mathfrak{M}, \gamma^1, q_0)$

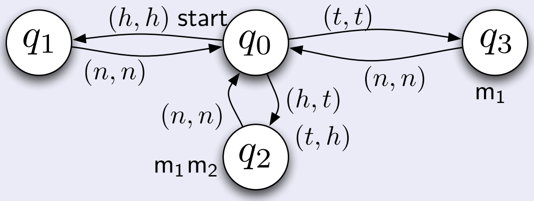
1/2	s_h	s_t
s_h	$0,0$	$\mathbf{1,1}$
s_t		

CGS $\rightsquigarrow$ strategic game

$\Gamma(\mathfrak{M}, \vec{\gamma}, q^I)$: **strategic game** associated with $\mathfrak{M}, \vec{\gamma}$

- $\mathfrak{M}$: generated CGS (by some action specifications)
- $\vec{\gamma} = (\gamma_1, \ldots, \gamma_k) \in \mathit{Prefs}$: preferences (**LTL**-formulae)
- q^I: initial state.

CGS



Preferences:

- player 1: $\Diamond(\mathrm{m}_1 \wedge \mathrm{m}_2)$
- player 2: $\Diamond \mathrm{m}_2$

$\Gamma(\mathfrak{M}, \gamma^1, q_0)$

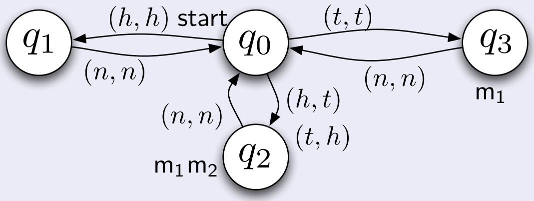
1/2	s_h	s_t
s_h	$0, 0$	$\mathbf{1, 1}$
s_t	$\mathbf{1, 1}$	

CGS $\rightsquigarrow$ strategic game

$\Gamma(\mathfrak{M}, \vec{\gamma}, q^I)$: **strategic game** associated with $\mathfrak{M}, \vec{\gamma}$

- $\mathfrak{M}$: generated CGS (by some action specifications)
- $\vec{\gamma} = (\gamma_1, \ldots, \gamma_k) \in \mathit{Prefs}$: preferences (**LTL**-formulae)
- q^I: initial state.

CGS



Preferences:

- player 1: $\Diamond(m_1 \wedge m_2)$
- player 2: $\Diamond m_2$

$\Gamma(\mathfrak{M}, \gamma^1, q_0)$

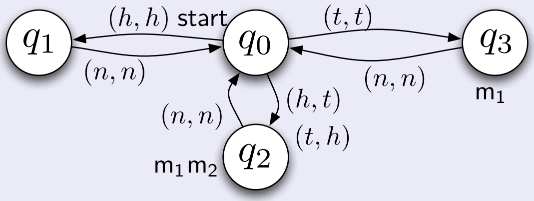
$1/2$	s_h	s_t
s_h	$0, 0$	$\mathbf{1}, \mathbf{1}$
s_t	$\mathbf{1}, \mathbf{1}$	$0, 0$

CGS ⇝ strategic game

$\Gamma(\mathfrak{M}, \vec{\gamma}, q^I)$: **strategic game** associated with $\mathfrak{M}, \vec{\gamma}$

- $\mathfrak{M}$: generated CGS (by some action specifications)
- $\vec{\gamma} = (\gamma_1, \ldots, \gamma_k) \in \mathit{Prefs}$: preferences (**LTL**-formulae)
- q^I: initial state.

CGS



Preferences:

- player 1: $\Diamond(m_1 \wedge m_2)$
- player 2: $\Diamond m_2$

$\Gamma(\mathfrak{M}, \gamma^1, q_0)$

1/2	s_h	s_t
s_h	0, 0	**1, 1**
s_t	**1, 1**	0, 0

If a **transition** specified by the strategy **does not exist** the **payoff is set** -1!

Implementability

M $\mathcal{S}$-**implements** f **over** $\mathfrak{M}, q^I, \mathcal{P}refs$ iff

$$\forall \vec{\gamma} \in \mathcal{P}refs \quad \forall s \in \mathcal{S}(\mathfrak{M} \restriction M, \vec{\gamma}, q^I) : paths_{\mathfrak{M} \restriction M}(q^I, s) \models^{\mathbf{LTL}} f(\vec{\gamma}).$$

- M: norm set (i.e. set of rules **modifying states**)
- $\mathcal{S}$: solution concept
- $f : \mathcal{P}refs \to \mathcal{L}_{\mathbf{LTL}}(\Pi_{\mathfrak{M}})$: social choice function

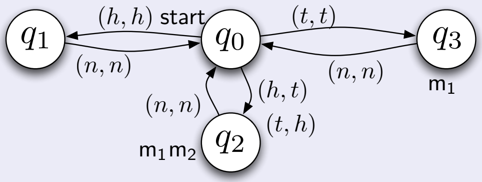
Implementability

M $\mathcal{S}$**-implements** f **over** $\mathfrak{M}, q^I, \mathcal{P}refs$ iff

$$\forall \vec{\gamma} \in \mathcal{P}refs \quad \forall s \in \mathcal{S}(\mathfrak{M} \upharpoonright M, \vec{\gamma}, q^I) : paths_{\mathfrak{M}\upharpoonright M}(q^I, s) \models^{\mathbf{LTL}} f(\vec{\gamma}).$$

- M: norm set (i.e. set of rules **modifying states**)
- $\mathcal{S}$: solution concept
- $f : \mathcal{P}refs \to \mathcal{L}_{\mathbf{LTL}}(\Pi_{\mathfrak{M}})$: social choice function

Example



$\Gamma(\mathfrak{M}, \gamma^1, q_0)$

1/2	s_h	s_t
s_h	0, 0	**1, 1**
s_t	**1, 1**	0, 0

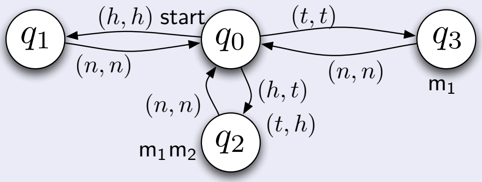
Implementability

M $\mathcal{S}$**-implements** f **over** $\mathfrak{M}, q^I, \mathcal{P}refs$ iff

$$\forall \vec{\gamma} \in \mathcal{P}refs \quad \forall s \in \mathcal{S}(\mathfrak{M} \upharpoonright M, \vec{\gamma}, q^I) : paths_{\mathfrak{M} \upharpoonright M}(q^I, s) \models^{\mathbf{LTL}} f(\vec{\gamma}).$$

- M: norm set (i.e. set of rules **modifying states**)
- $\mathcal{S}$: solution concept
- $f : \mathcal{P}refs \to \mathcal{L}_{\mathbf{LTL}}(\Pi_{\mathfrak{M}})$: social choice function

Example



$\Gamma(\mathfrak{M}, \gamma^1, q_0)$

1/2	s_h	s_t
s_h	0, 0	**1, 1**
s_t	**1, 1**	0, 0

Does $M_\emptyset$ $\mathcal{NE}$-implement $f(\vec{\gamma}) = \Box\neg(m_1 \wedge m_2)$?

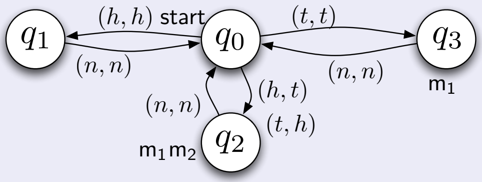
Implementability

M $\mathcal{S}$**-implements** f **over** $\mathfrak{M}, q^I, \mathcal{P}refs$ iff

$$\forall \vec{\gamma} \in \mathcal{P}refs \quad \forall s \in \mathcal{S}(\mathfrak{M} \upharpoonright M, \vec{\gamma}, q^I) : paths_{\mathfrak{M} \upharpoonright M}(q^I, s) \models^{\mathbf{LTL}} f(\vec{\gamma}).$$

- M: norm set (i.e. set of rules **modifying states**)
- $\mathcal{S}$: solution concept
- $f : \mathcal{P}refs \rightarrow \mathcal{L}_{\mathbf{LTL}}(\Pi_{\mathfrak{M}})$: social choice function

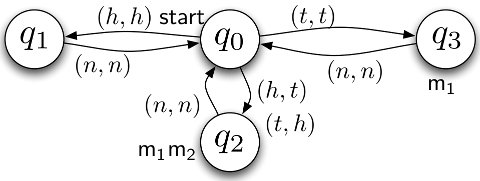
Example



$\Gamma(\mathfrak{M}, \gamma^1, q_0)$

1/2	s_h	s_t
s_h	0, 0	**1, 1**
s_t	**1, 1**	0, 0

Does $M_\emptyset$ $\mathcal{NE}$-implement $f(\vec{\gamma}) = \Box\neg(m_1 \wedge m_2)$? **NO!** $(q_0 q_2)^\omega \not\models f(\vec{\gamma})$

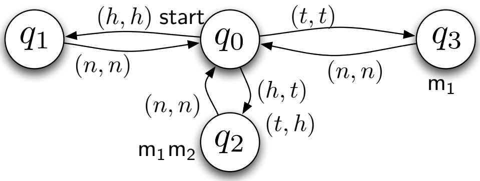


- $\gamma_1 = \Diamond(m_1 \wedge m_2)$
  $\gamma_2 = \Diamond m_2$
- $f(\vec{\gamma}) = \Box\neg(m_1 \wedge m_2)$

$\Gamma(\mathfrak{M}, \vec{\gamma}, q_0)$:

1/2	s_h	s_t
s_h	0, 0	**1, 1**
s_t	**1, 1**	0, 0

Is there a **norm set** M which $\mathcal{NE}$-implements $f(\vec{\gamma}) = \Box\neg(m_1 \wedge m_2)$?

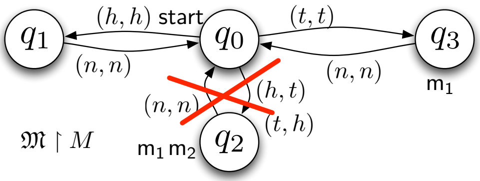


- $\gamma_1 = \Diamond(m_1 \wedge m_2)$
  $\gamma_2 = \Diamond m_2$
- $f(\vec{\gamma}) = \Box\neg(m_1 \wedge m_2)$

$\Gamma(\mathfrak{M}, \vec{\gamma}, q_0)$:

1/2	s_h	s_t
s_h	0, 0	**1, 1**
s_t	**1, 1**	0, 0

Is there a **norm set** M which $\mathcal{NE}$-implements $f(\vec{\gamma}) = \Box\neg(m_1 \wedge m_2)$?



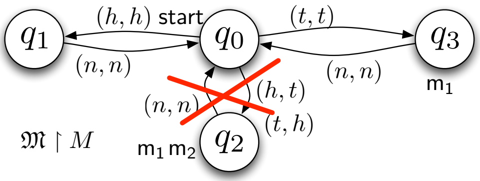
$\mathfrak{M} \upharpoonright M$

- $\gamma_1 = \Diamond(m_1 \wedge m_2)$
  $\gamma_2 = \Diamond m_2$
- $f(\vec{\gamma}) = \Box\neg(m_1 \wedge m_2)$

$\Gamma(\mathfrak{M}, \vec{\gamma}, q_0)$:

1/2	s_h	s_t
s_h	0, 0	**1, 1**
s_t	**1, 1**	0, 0

Norm set: $M = (\ \{\neg\text{start} \wedge m_1 \wedge m_2 \rightarrow_{cr} v\}, \{v \rightarrow_{sr} \neg m_1\})$



- $\gamma_1 = \Diamond(m_1 \wedge m_2)$
  $\gamma_2 = \Diamond m_2$
- $f(\vec{\gamma}) = \Box\neg(m_1 \wedge m_2)$

$\Gamma(\mathfrak{M}, \vec{\gamma}, q_0)$:

1/2	s_h	s_t
s_h	$0, 0$	$\mathbf{1, 1}$
s_t	$\mathbf{1, 1}$	$0, 0$

Norm set: $M = (\ \{\neg start \wedge m_1 \wedge m_2 \rightarrow_{cr} v\}, \{v \rightarrow_{sr} \neg m_1\})$

$\Gamma(\mathfrak{M} \upharpoonright M, \vec{\gamma}, q_0)$:

1/2	s_h	s_t
s_h	$\mathbf{0, 0}$	$-1, -1$
s_t	$-1, -1$	$\mathbf{0, 0}$

Verification Problems

$(\mathcal{S}, M)$-implementation problem $\mathrm{IP}^{\mathcal{S}}_{\mathrm{M}}$

Given a **norm set** M do the **outcome paths** of $\mathfrak{M} \upharpoonright M$ satisfy f if agents follow $\mathcal{S}$-equilibria strategy profiles?

$\mathcal{S}$-implementation problem $\mathrm{IP}^{\mathcal{S}}$

Is there a **norm set** M such that ...?

Verification Results

The following results are about **Nash equilibria**!

Theorem (M-implementation problem)

The problem $\mathsf{IP}_{\mathsf{M}}^{\mathcal{NE}}$ *is* $\mathbf{\Pi}_2^{\mathbf{P}}$*-complete.*

Theorem (Implementation problem)

The problem $\mathsf{IP}^{\mathcal{NE}}$ *is* $\mathbf{\Sigma}_3^{\mathbf{P}}$*-complete.*

Remark: $\mathbf{\Pi}_2^{\mathbf{P}} = \mathbf{coNP}^{\mathbf{NP}}$ $\quad \mathbf{\Sigma}_2^{\mathbf{P}} = \mathbf{NP}^{\mathbf{NP}}$ $\quad \mathbf{\Sigma}_3^{\mathbf{P}} = \mathbf{NP}^{\mathbf{\Sigma}_2^{\mathbf{P}}}$

Conclusions

- Modelling **normative systems**
- **Mechanisms design** to implement **designers' objectives**
- **Verification problems** $\rightsquigarrow$ **Synthesis** of a mechanism
- Interesting: **Game theory** in proofs
- **Future work**:
  - expressivity (norms set, preferences, . . .)
  - other solution concepts
  - (non-)implementability