An introduction to cognitive planning

Jorge Luis Fernandez Davila, Dominique Longin, Emiliano Lorini, Frédéric Maris

An introduction to cognitive planning

J. Fernandez, D. Longin, E. Lorini, F. Maris
IRIT, Université de Toulouse, CNRS, Toulouse INP, UT3
{prenom.nom}@irit.fr

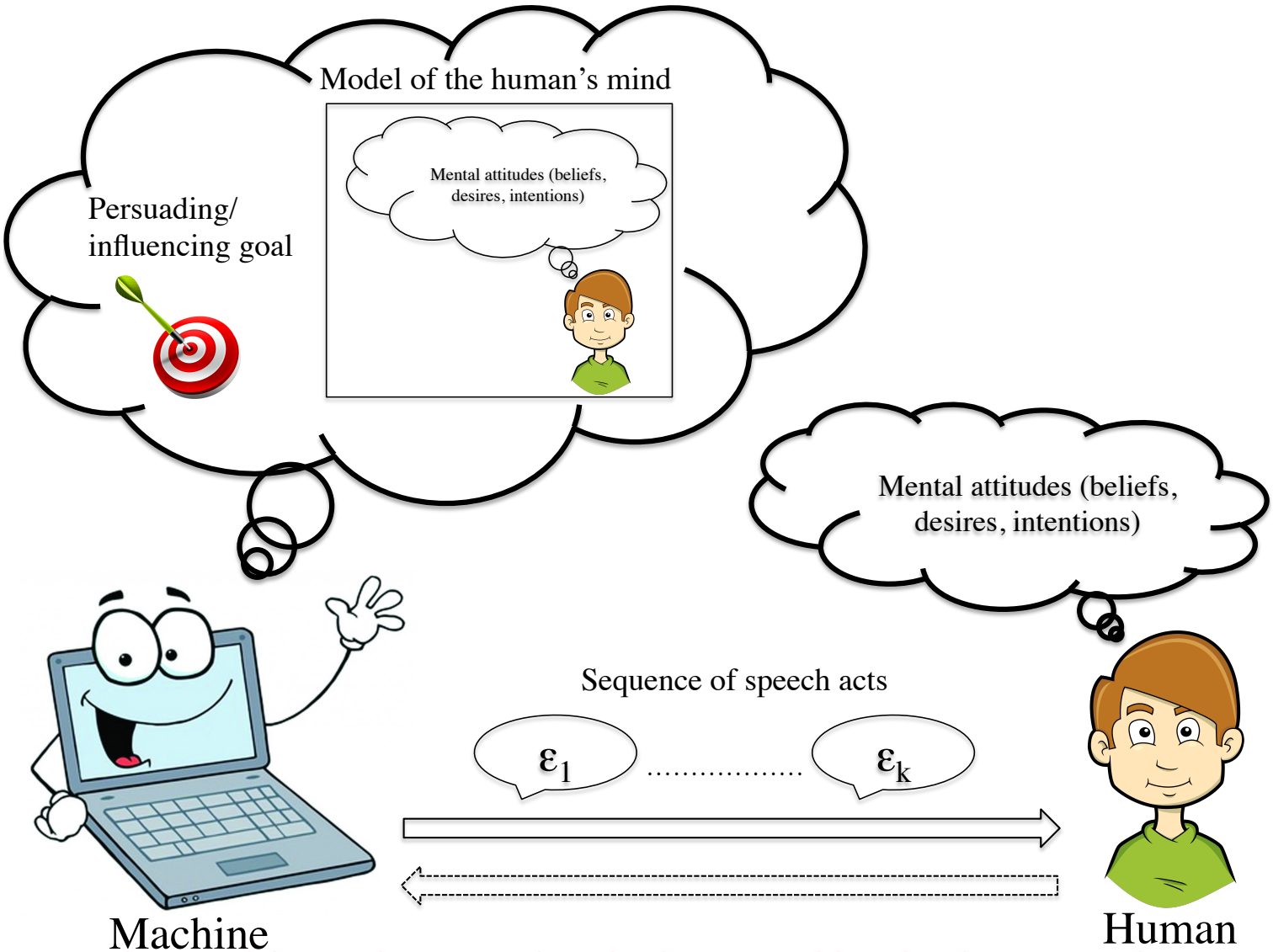
From classical and epistemic planning to cognitive planning

- Classical planning
  - Actions: ontic (physical) actions
  - Goal: modifying the environment
- Epistemic planning
  - Actions: epistemic/informative actions
  - Goal: affecting/changing the target's beliefs

From classical and epistemic planning to cognitive planning

- **Cognitive planning**
  - Actions: speech acts
  - Goal: affecting/changing the target's cognitive state
    - beliefs
    - intentions and behaviour
  - It relies on a theory of the influence of beliefs and desires (reasons) on intentions/behaviors
- ANR project CoPains ("Cognitive Planning in Persuasive Multimodal Communication") 2019-2023:
  $\Rightarrow$ Webpage: `https://www.irit.fr/CoPains/`

Cognitive planning



Cognitive planning for persuasion and influence

Influence: inducing an agent to perform a certain action

1. by impeding or enabling some of her actions/choices (regimentation)
2. by changing her payoffs (incentive/deterrent)
3. by changing her beliefs (persuasion)

Cognitive planning for persuasion and influence

Example (Influence through regimentation)

The gouvernement influences people to vaccinate against covid-19 by precluding access to public spaces (restaurants, museums, etc) in the absence of a vaccination certificate.

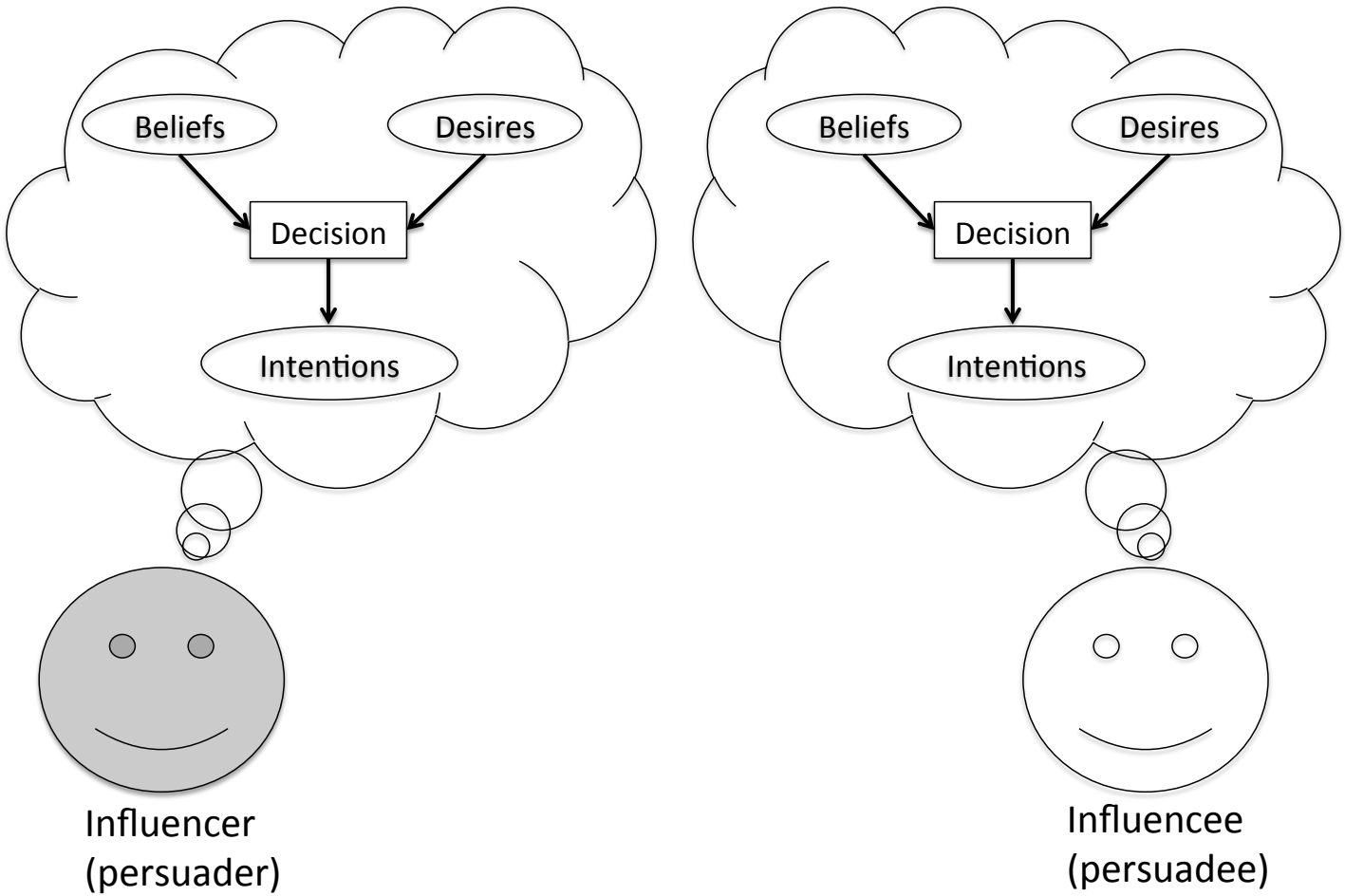
Example (Influence through incentive creation)

The gouvernement influences people to buy electric cars by offering electric car tax cuts.

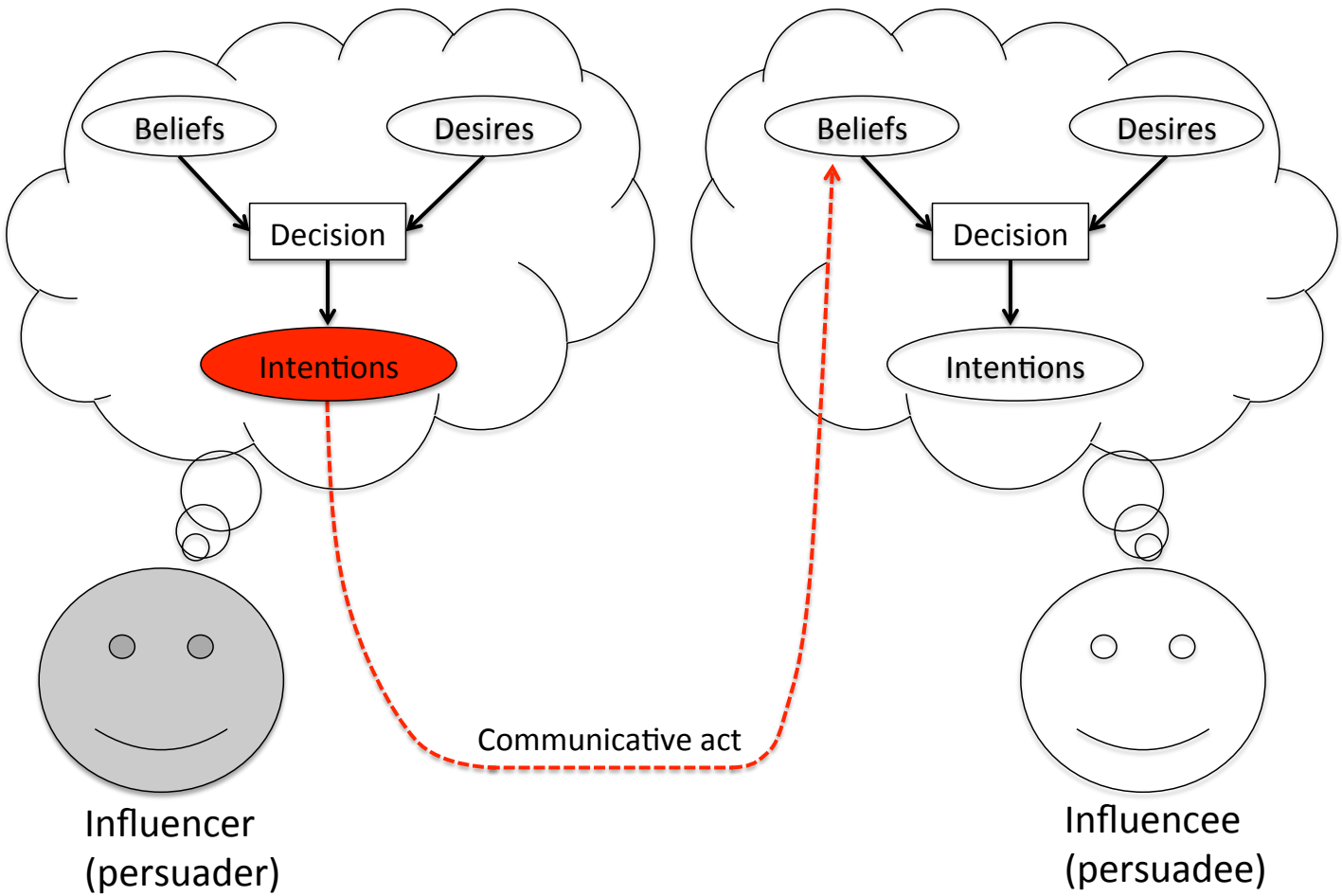
Example (Influence through persuasion)

Bob influences Ann to not vaccinate against covid-19 by convincing her that covid-19 mRNA vaccines modify human DNA.

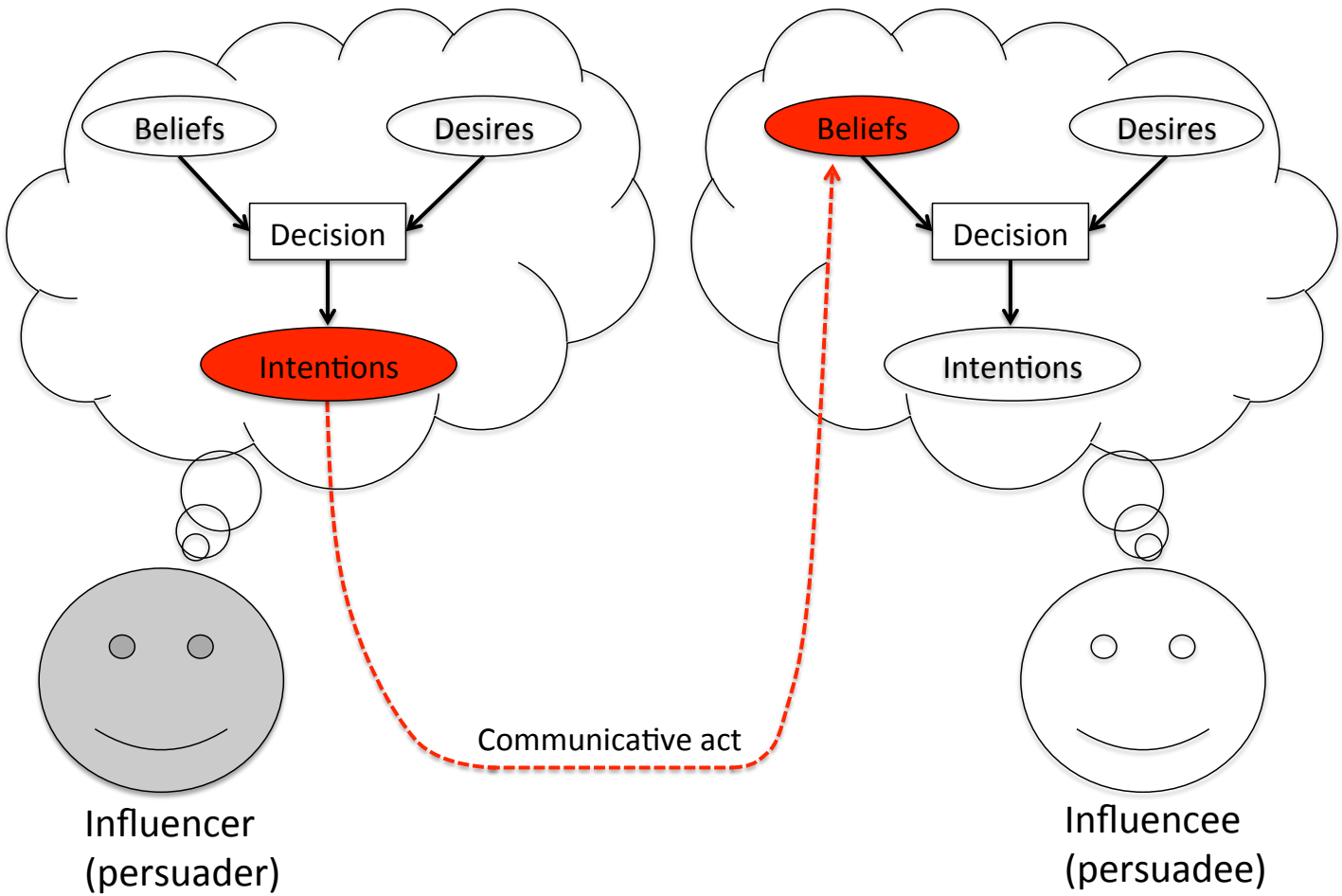
Cognitive planning for persuasion and influence



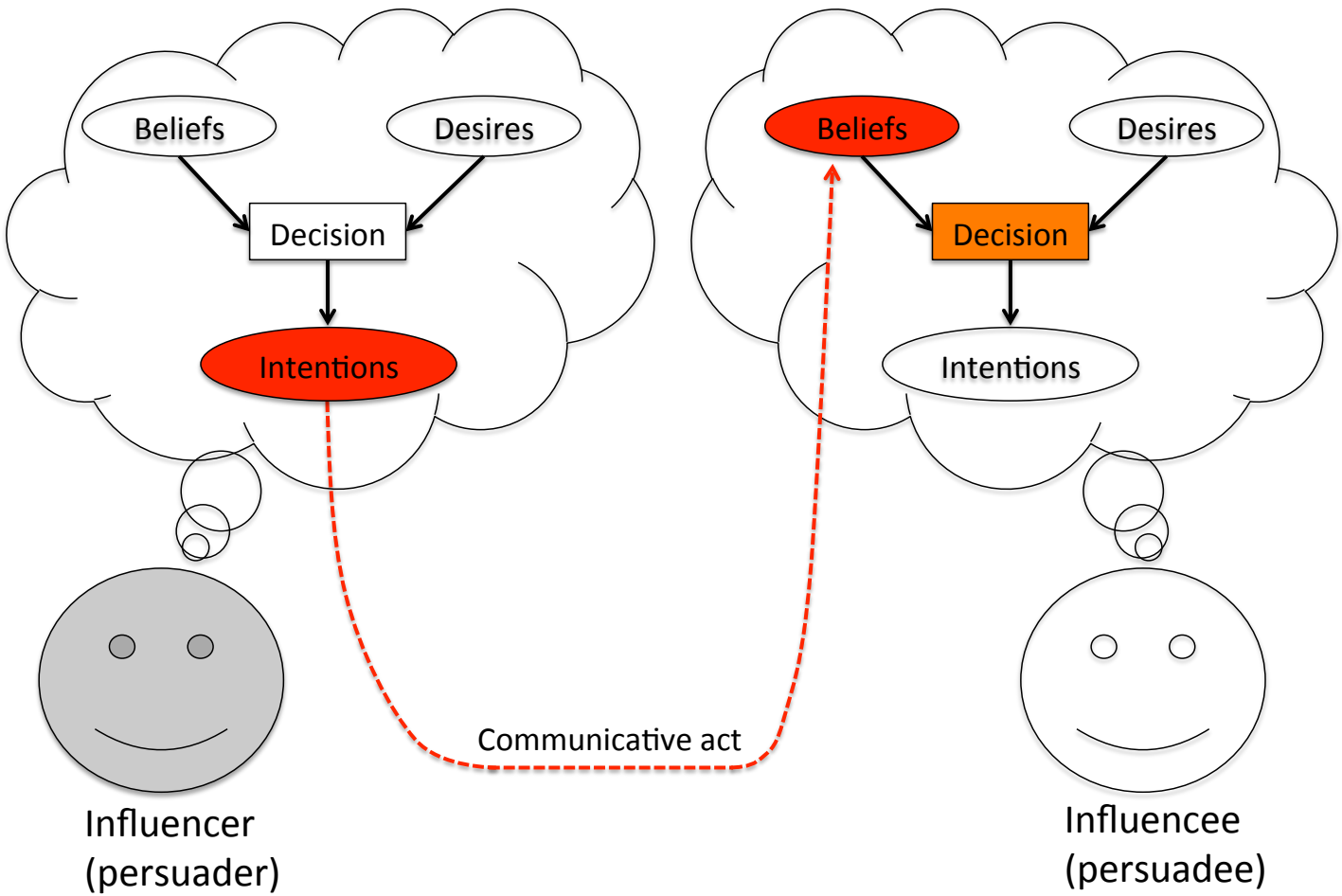
Cognitive planning for persuasion and influence



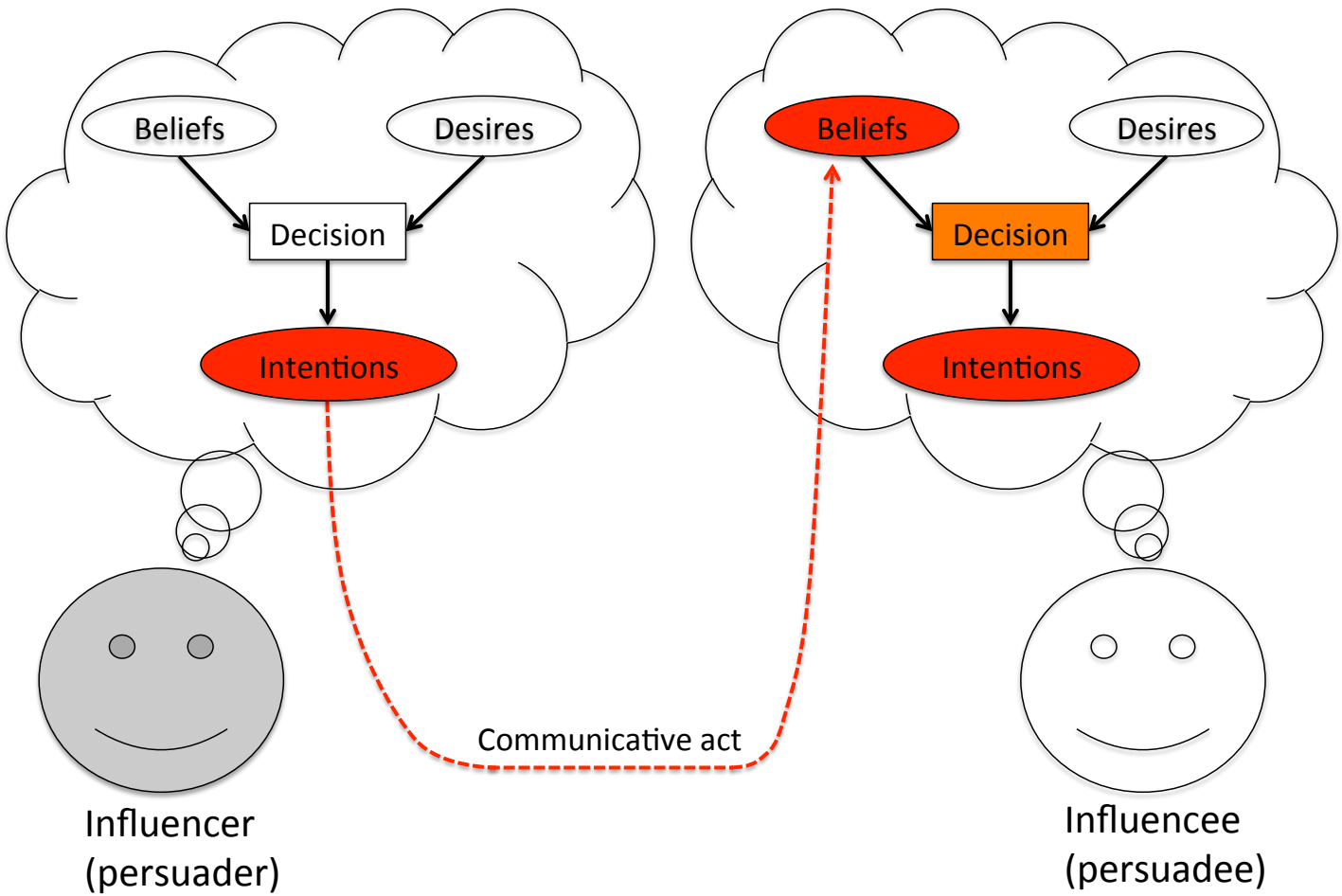
Cognitive planning for persuasion and influence



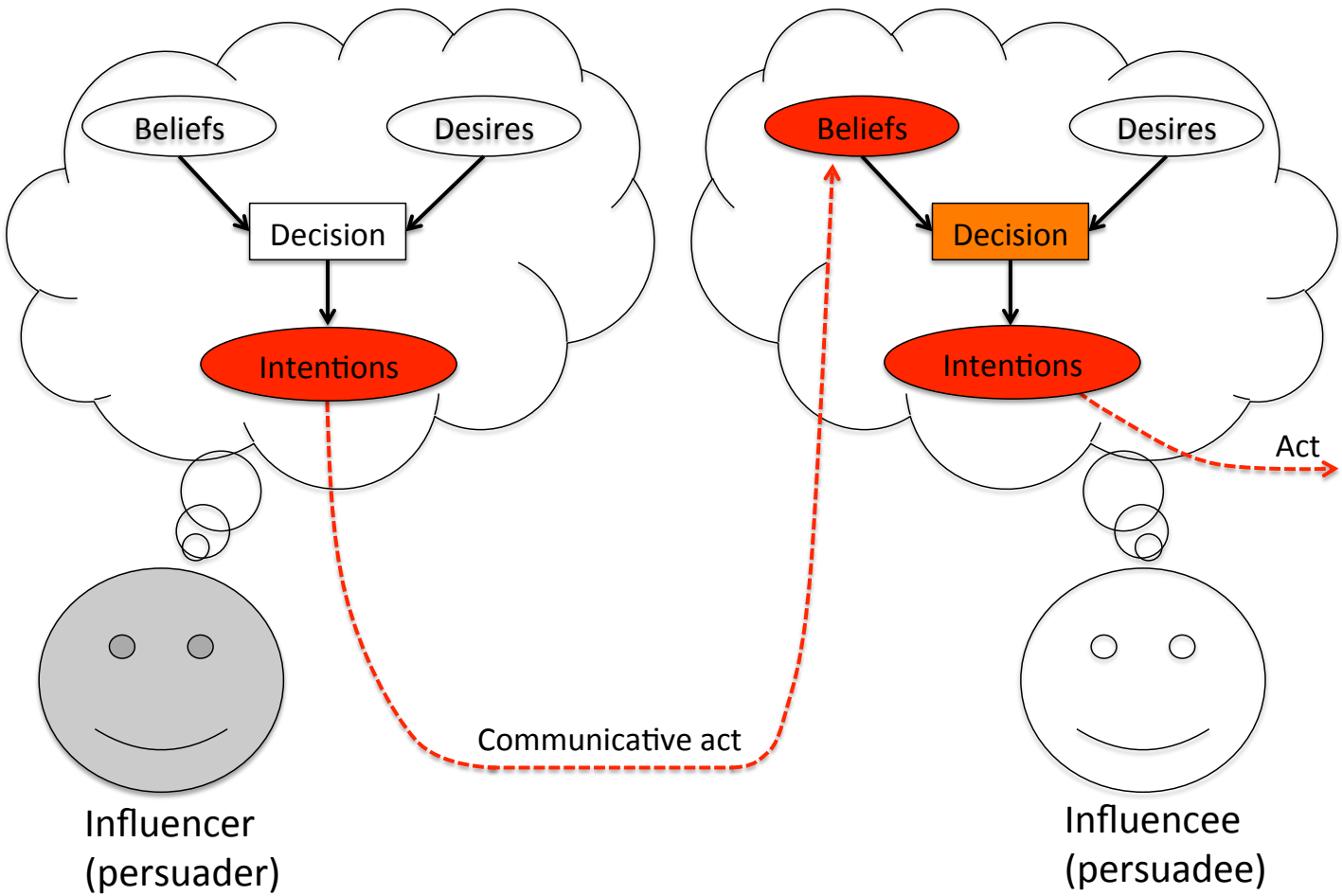
Cognitive planning for persuasion and influence



Cognitive planning for persuasion and influence



Cognitive planning for persuasion and influence



Plan of the tutorial

- Part 1 Logical framework for cognitive planning
  - Epistemic logic with a semantics exploiting belief bases
  - NP-fragment and reduction to SAT
  - Formalization of cognitive planning problem and complexity results
- Part 2 QBF encoding of cognitive planning and ToulST platform
- Part 3 Implementation and applications
  - Recommender system for sport activities
  - Conversational agent for motivational interviewing
- Part 4 Formal extension and implementation for collaborative gaming
  - Extended logical framework and encoding of the Yōkai game
  - General architecture and implementation

Language

Language:

$$\begin{array}{lrcl} \mathcal{L}_0: & \alpha & ::= & p \mid \neg\alpha \mid \alpha_1 \wedge \alpha_2 \mid \alpha_1 \vee \alpha_2 \mid \triangle_i \alpha, \\ \mathcal{L}: & \varphi & ::= & \alpha \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \Box_i \varphi \mid \Diamond_i \varphi, \end{array}$$

with $p \in Atm$ and $i \in Agt$

$\triangle_i \alpha$: agent i explicitly believes that α

$\Box_i \varphi$: agent i implicitly believes that φ

- E. Lorini (2020). Rethinking Epistemic Logic with Belief Bases. *Artificial Intelligence*, 282.
- E. Lorini (2018). In Praise of Belief Bases: Doing Epistemic Logic without Possible Worlds. In *Proc. of AAAI-18*, AAAI Press, pp. 1915-1922.

Semantics

Definition (State)

A state is a tuple $B = (B_1, \ldots, B_n, V)$ where:

- for every $i \in Agt$, $B_i \subseteq \mathcal{L}_0$ is agent i's belief base,
- $V \subseteq Atm$ is the actual environment.

The set of all states is denoted by $\mathbf{S}$.

Semantics

Definition (Satisfaction relation)

Let $B = (B_1, \ldots, B_n, V) \in \mathbf{S}$. Then:

$$
\begin{aligned}
B \models p &\iff p \in V \\
B \models \neg\alpha &\iff B \not\models \alpha \\
B \models \alpha_1 \wedge \alpha_2 &\iff B \models \alpha_1 \text{ and } B \models \alpha_2 \\
B \models \triangle_i \alpha &\iff \alpha \in B_i
\end{aligned}
$$

Semantics

Definition (Multi-agent belief model)

A multi-agent belief model (MAB) is a pair (B, Cxt), where $B \in \mathbf{S}$ and $Cxt \subseteq \mathbf{S}$. The class of MABs is denoted by $\mathbf{M}$.

Definition (Epistemic alternatives)

Let $B = (B_1, \ldots, B_n, V), B' = (B'_1, \ldots, B'_n, V') \in \mathbf{S}$. Then,

$$B \mathcal{R}_i B' \text{ if and only if } \forall \alpha \in B_i : B' \models \alpha.$$

Semantics

Definition (Satisfaction relation)

Let $(B, Cxt) \in \mathbf{M}$. Then:

$$(B, Cxt) \models \alpha \iff B \models \alpha$$

$$(B, Cxt) \models \Box_i \varphi \iff \forall B' \in Cxt : \text{if } B \mathcal{R}_i B' \text{ then } (B', Cxt) \models \varphi$$

Theorem

Checking satisfiability of $\mathcal{L}(Atm, Agt)$ formulas in the class $\mathbf{M}$ is a PSPACE-hard problem.

NP-fragment

Single-reasoner fragment with $Agt = \{\mathfrak{m}, \mathfrak{h}\}$:

$$\mathcal{L}_{\text{Frag}}: \quad \varphi \quad ::= \quad \alpha \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \Box_{\mathfrak{m}}\alpha \mid \Diamond_{\mathfrak{m}}\alpha$$

where α ranges over $\mathcal{L}_0$

Recall: $\mathcal{L}_0: \quad \alpha \quad ::= \quad p \mid \neg\alpha \mid \alpha_1 \wedge \alpha_2 \mid \alpha_1 \vee \alpha_2 \mid \triangle_{\mathfrak{m}}\alpha \mid \triangle_{\mathfrak{h}}\alpha$

Agent $\mathfrak{m}$: the ‘machine’

Agent $\mathfrak{h}$: the ‘human’

Polynomial reduction to SAT

$$\mathcal{L}_{\text{Frag}} \overset{nnf}{\dashrightarrow} \mathcal{L}^{\text{NNF}}_{\text{Frag}} \overset{tr_1}{\dashrightarrow} \mathcal{L}_{\text{Mod}} \overset{tr_2}{\dashrightarrow} \mathcal{L}_{\text{Prop}}$$

Figure: Summary of reduction process

Theorem

Checking satisfiability of formulas in $\mathcal{L}_{\text{Frag}}$ in the class $\mathbf{M}$ is an NP-complete problem.

Dynamic extension

Dynamic language:

$$\mathcal{L}^{+}_{\text{Frag}}: \quad \varphi \quad ::= \quad \alpha \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \Box_{\mathfrak{m}}\alpha \mid \Diamond_{\mathfrak{m}}\alpha \mid [+_i\alpha]\varphi$$

$[+_i\alpha]\varphi$: φ holds after agent i has privately expanded her belief base with α

Definition (Satisfaction relation, cont.)

Let $B = (B_1, \ldots, B_n, V) \in \mathbf{S}$ and let $(B, Cxt) \in \mathbf{M}$. Then:

$$(B, Cxt) \models [+_i\alpha]\varphi \iff (B^{+_i\alpha}, Cxt) \models \varphi$$

with $V^{+_i\alpha} = V$, $B_i^{+_i\alpha} = B_i \cup \{\alpha\}$ and $B_j^{+_i\alpha} = B_j$ for all $j \neq i$.

Dynamic extension

The following equivalences are valid in the class $\mathbf{M}$:

$$[+_i\alpha]\alpha' \leftrightarrow \begin{cases} \top, & \text{if } \alpha' = \triangle_i\alpha, \\ \alpha', & \text{otherwise;} \end{cases}$$

$$[+_i\alpha]\neg\varphi \leftrightarrow \neg[+_i\alpha]\varphi;$$

$$[+_i\alpha](\varphi_1 \wedge \varphi_2) \leftrightarrow [+_i\alpha]\varphi_1 \wedge [+_i\alpha]\varphi_2;$$

$$[+_i\alpha](\varphi_1 \vee \varphi_2) \leftrightarrow [+_i\alpha]\varphi_1 \vee [+_i\alpha]\varphi_2;$$

$$[+_i\alpha]\Box_{\mathfrak{m}}\alpha' \leftrightarrow \begin{cases} \Box_{\mathfrak{m}}(\alpha \to \alpha'), & \text{if } i = \mathfrak{m}, \\ \Box_{\mathfrak{m}}\alpha', & \text{otherwise;} \end{cases}$$

$$[+_i\alpha]\Diamond_{\mathfrak{m}}\alpha' \leftrightarrow \begin{cases} \Diamond_{\mathfrak{m}}(\alpha \wedge \alpha'), & \text{if } i = \mathfrak{m}, \\ \Diamond_{\mathfrak{m}}\alpha', & \text{otherwise.} \end{cases}$$

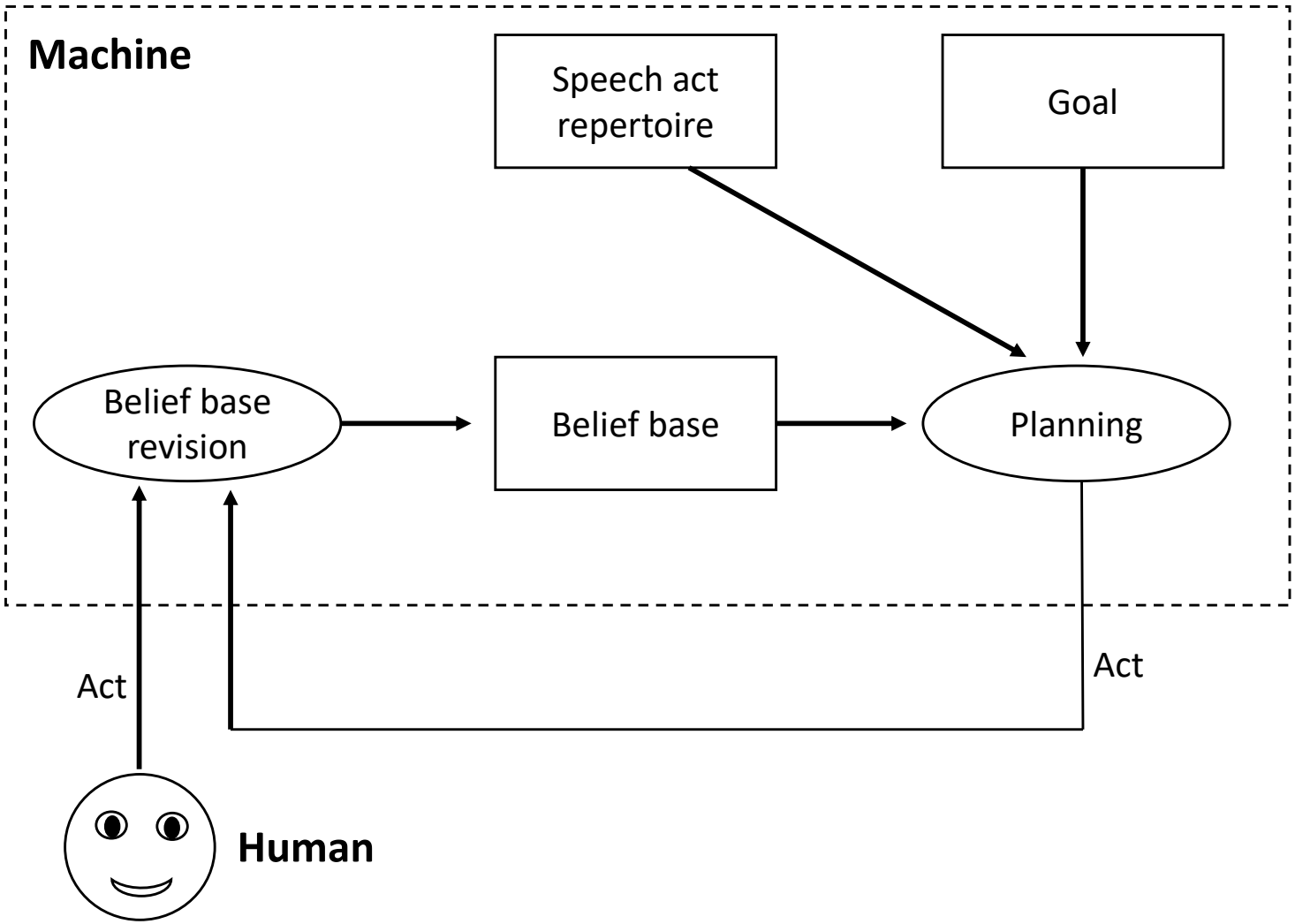
Dynamic extension

Polynomial reduction of satisfiability for $\mathcal{L}^{+}_{\text{Frag}}$ to satisfiability for $\mathcal{L}_{\text{Frag}}$ via the previous reduction axioms

Theorem

Checking satisfiability of formulas in $\mathcal{L}^{+}_{\text{Frag}}$ in the class $\mathbf{M}$ is an NP-complete problem.

General architecture



General architecture

- Interrogative phase
  - Agent $\mathfrak{m}$ gathers information about agent $\mathfrak{h}$'s cognitive state
  - Sequence of questions by $\mathfrak{m}$ to $\mathfrak{h}$
  - After $\mathfrak{h}$'s answer to $\mathfrak{m}$'s question, $\mathfrak{m}$'s expands/revises its beliefs
- Informative phase
  - Core of the persuasion/influence process
  - Sequence of assertions by $\mathfrak{m}$ aimed at modifying $\mathfrak{h}$'s cognitive state

Cognitive planning problems

⇒ **Agent $\mathfrak{m}$'s set of informative actions**:

$$Act_{\mathfrak{m}} = \{+_{\mathfrak{m}}\alpha : \alpha \in \mathcal{L}_0\}$$

with elements $\epsilon, \epsilon', \ldots$

⇒ **Agent $\mathfrak{m}$'s set of yes-no questions**:

$$Que_{\mathfrak{m}} = \{?_{\mathfrak{m},\mathfrak{h}}\alpha : \alpha \in \mathcal{L}_0\}$$

with elements $\lambda, \lambda', \ldots$

Cognitive planning problems

⇒ **Answer function**:

$$\mathcal{A} : Que_{\mathfrak{m}} \longrightarrow 2^{Act_{\mathfrak{m}}}$$

with $\mathcal{A}(?_{\mathfrak{m},\mathfrak{h}}\alpha) = \{ +_{\mathfrak{m}} \triangle_{\mathfrak{h}}\alpha, +_{\mathfrak{m}}\neg\triangle_{\mathfrak{h}}\alpha\}$
and default answer $da(?_{\mathfrak{m},\mathfrak{h}}\alpha) = +_{\mathfrak{m}}\triangle_{\mathfrak{h}}\alpha$

⇒ **Necessary consequence of a question**:

$$[\lambda]\varphi \stackrel{\text{def}}{=} \bigwedge_{\rho \in \mathcal{A}(\lambda)} [\rho]\varphi$$

Cognitive planning problems

⇒ **Set of events**: $Evt_{\mathfrak{m}} = Act_{\mathfrak{m}} \cup Que_{\mathfrak{m}}$ with elements $\gamma, \gamma', \ldots$

⇒ **Executability precondition function**:

$$\mathcal{P} : Evt_{\mathfrak{m}} \longrightarrow \mathcal{L}_{\mathsf{Frag}}$$

⇒ **Necessary consequence of an event**:

$$\langle\!\langle \gamma \rangle\!\rangle \varphi \stackrel{\mathsf{def}}{=} \mathcal{P}(\gamma) \wedge [\gamma]\varphi$$

$\langle\!\langle \gamma \rangle\!\rangle \varphi$: event γ can take place and φ holds after its occurence

Cognitive planning problems

Definition (Informative planning problem)

An interrogative planning problem is a tuple $\langle \Sigma, Op_{\text{inf}}, \alpha_G \rangle$ where:

- $\Sigma \subset \mathcal{L}_0$ is a finite set of agent $\mathfrak{m}$'s available information,
- $Op_{\text{inf}} \subset Act_{\mathfrak{m}}$ is a finite set of agent $\mathfrak{m}$'s informative actions,
- $\alpha_G \in \mathcal{L}_0$ is agent $\mathfrak{m}$'s goal.

A solution plan to the informative planning problem is a sequence of informative actions $\epsilon_1, \ldots, \epsilon_k$ from Op_{inf} for some k such that

$$\Sigma \models_{\mathbf{M}} \langle\langle \epsilon_1 \rangle\rangle \ldots \langle\langle \epsilon_k \rangle\rangle \Box_{\mathfrak{m}} \alpha_G$$

Cognitive planning problems

Definition (Interrogative planning problem)

An interrogative planning problem is a tuple $\langle \Sigma, Op_{\text{inf}}, Op_{\text{quest}}, \alpha_G \rangle$ where:

- $\Sigma \subset \mathcal{L}_0$ is a finite set of agent $\mathfrak{m}$'s available information,
- $Op_{\text{inf}} \subset Act_{\mathfrak{m}}$ is a finite set of agent $\mathfrak{m}$'s informative actions,
- $Op_{\text{quest}} \subset Que_{\mathfrak{m}}$ is a finite set of agent $\mathfrak{m}$'s questions,
- $\alpha_G \in \mathcal{L}_0$ is agent $\mathfrak{m}$'s goal.

Cognitive planning problems

A strong solution plan to the interrogative planning problem is a sequence of questions $\lambda_1, \ldots, \lambda_m$ from Op_{quest} such that

$$\Sigma \models_{\mathbf{M}} \langle\langle \lambda_1 \rangle\rangle \ldots \langle\langle \lambda_m \rangle\rangle \top$$

and $\forall \rho_1 \in \mathcal{A}(\lambda_1), \ldots, \forall \rho_m \in \mathcal{A}(\lambda_m)$, $\exists \tau_1, \ldots, \tau_k \in Op_{\text{inf}}$ such that

$$\Sigma \models_{\mathbf{M}} [\rho_1] \ldots [\rho_m] \langle\langle \tau_1 \rangle\rangle \ldots \langle\langle \tau_k \rangle\rangle \Box_{\mathfrak{m}} \alpha_G$$

Cognitive planning problems

A weak solution plan to the interrogative planning problem is a sequence of questions $\lambda_1, \ldots, \lambda_m$ from Op_{quest} such that

$$\Sigma \models_{\mathbf{M}} \langle\langle \lambda_1 \rangle\rangle \ldots \langle\langle \lambda_m \rangle\rangle \top$$

and $\exists \tau_1, \ldots, \tau_k \in Op_{\text{inf}}$ such that

$$\Sigma \models_{\mathbf{M}} [da(\lambda_1)] \ldots [da(\lambda_m)] \langle\langle \tau_1 \rangle\rangle \ldots \langle\langle \tau_k \rangle\rangle \Box_{\mathfrak{m}} \alpha_G$$

Cognitive planning problems

Problems:

- ES-INF: checking existence of a solution for an informative planning problem
- EWS-INT: checking existence of a weak solution for an interrogative planning problem
- ESS-INT: checking existence of a strong solution for an interrogative planning problem

Theorem

The ES-INF problem and EWS-INT problem are $\mathrm{NP}^{\mathrm{NP}} = \Sigma_2^{\mathrm{P}}$*-complete.*

Belief revision

- Propositional language $\mathcal{L}_{\text{PROP}}$ built from the following set of atomic formulas:

$$Atm^{+} = Atm \cup \{p_{\triangle_i \alpha} : \triangle_i \alpha \in \mathcal{L}_0\}$$

- Translation $tr_{\text{PROP}} : \mathcal{L}_0 \longrightarrow \mathcal{L}_{\text{PROP}}$:

$$\begin{aligned} tr_{\text{PROP}}(p) &= p \\ tr_{\text{PROP}}(\neg\alpha) &= \neg tr_{\text{PROP}}(\alpha) \\ tr_{\text{PROP}}(\alpha_1 \wedge \alpha_2) &= tr_{\text{PROP}}(\alpha_1) \wedge tr_{\text{PROP}}(\alpha_2) \\ tr_{\text{PROP}}(\triangle_i \alpha) &= p_{\triangle_i \alpha} \end{aligned}$$

- For $X \subseteq \mathcal{L}_0$, $tr_{\text{PROP}}(X) = \{tr_{\text{PROP}}(\alpha) : \alpha \in X\}$
- X is propositionally consistent if and only if $\bot \notin Cn\big(tr_{\text{PROP}}(X)\big)$

Belief revision

- Agent $\mathfrak{m}$'s core (or, immutable) information set $\Sigma_{core} \subseteq \mathcal{L}_0$
- Agent $\mathfrak{m}$'s volatile (or, mutable) information set $\Sigma_{mut} \subseteq \mathcal{L}_0$
- Agent $\mathfrak{m}$'s input set $\Sigma_{input} \subseteq \mathcal{L}_0$

Belief revision

Revision of $(\Sigma_{core}, \Sigma_{mut})$ by Σ_{input}:

1. if $\Sigma_{core} \cup \Sigma_{input}$ is not propositionally consistent then $Rev(\Sigma_{core}, \Sigma_{mut}, \Sigma_{input}) = (\Sigma_{core}, \Sigma_{mut})$
2. otherwise, $Rev(\Sigma_{core}, \Sigma_{mut}, \Sigma_{input}) = (\Sigma'_{core}, \Sigma'_{mut})$ with

$$\Sigma'_{core} = \Sigma_{core}$$

$$\Sigma'_{mut} = \bigcap_{X \in MCS(\Sigma_{core}\Sigma_{mut}, \Sigma_{input})} X$$

where $X \in MCS(\Sigma_{core}, \Sigma_{mut}, \Sigma_{input})$ if and only if:

- $X \subseteq \Sigma_{mut} \cup \Sigma_{input}$
- $\Sigma_{input} \subseteq X$
- $X \cup \Sigma_{core}$ is propositionally consistent
- there is no $X' \subseteq \Sigma_{mut} \cup \Sigma_{input}$ such that $X \subset X'$ and $X' \cup \Sigma_{core}$ is propositionally consistent

Plan of the tutorial

1. Logical framework for cognitive planning (Emiliano)
2. **QBF encoding of cognitive planning and TouIST platform (Frédéric) NOW!**
   - $\exists\forall$ optimal encoding of simple cognitive planning
   - TouIST: Toulouse Integrated Satisfiability Tool
3. Implementation and applications (Jorge)
4. Formal extension and implementation for collaborative gaming (Dominique & Jorge)

Part 2 - QBF encoding of cognitive planning and TouIST platform

∃∀ optimal encoding

ES-INF planning problem (recall)

A solution plan to an ES-INF planning problem $\langle \Sigma, Op, \alpha_G \rangle$ is a sequence of operators $\epsilon_1, \ldots, \epsilon_k$ from Op such that $\Sigma \models_{\mathbf{M}} \langle\langle \epsilon_1 \rangle\rangle \ldots \langle\langle \epsilon_k \rangle\rangle \Box_{\mathfrak{m}} \alpha_G$

Theorem

An ES-INF planning problem $\langle \Sigma, Op, \alpha_G \rangle$ has a solution plan if and only if it has a poly-size solution plan $\epsilon_1, \ldots, \epsilon_k$ with $k \leq |Op|$ and $\epsilon_i \neq \epsilon_j$ for all $i < j$.

Theorem

Checking plan existence for an ES-INF planning problem is Σ_2^{P}-complete.

∃∀ encoding of ES-INF planning problem $\langle \Sigma, Op, \alpha_G \rangle$

Theorem

Let $\varphi \in \mathcal{L}^{+}_{\text{Frag}}$ and let $\Sigma \subset \mathcal{L}_0$ be finite.
Then, $\Sigma \models_{\text{M}} \varphi$ if and only if $\models_{\text{M}} \bigwedge_{\alpha \in \Sigma} \Box_{\text{m}}\, \alpha \rightarrow \varphi$.

A plan candidate $\epsilon_1, \ldots, \epsilon_k$ is a solution plan for $\langle \Sigma, Op, \alpha_G \rangle$ if and only if the following formula is valid:

$$\left(\bigwedge_{\alpha \in \Sigma} \Box_{\text{m}}\, \alpha \right) \rightarrow \langle\!\langle \epsilon_1 \rangle\!\rangle \ldots \langle\!\langle \epsilon_{k-1} \rangle\!\rangle \langle\!\langle \epsilon_k \rangle\!\rangle \Box_{\text{m}} \alpha_G$$

which can be also written as:

$$\left(\bigwedge_{\alpha \in \Sigma} \Box_{\text{m}}\, \alpha \right) \rightarrow \left(\bigwedge_{i \in \{1,\ldots,k\}} [+_{\text{m}} \alpha_{\epsilon_1}] \ldots [+_{\text{m}} \alpha_{\epsilon_{i-1}}] \mathcal{P}(\epsilon_i) \right) \wedge [+_{\text{m}} \alpha_{\epsilon_1}] \ldots [+_{\text{m}} \alpha_{\epsilon_{k-1}}] [+_{\text{m}} \alpha_{\epsilon_k}] \Box_{\text{m}} \alpha_G$$

∃∀ encoding of ES-INF planning problem $\langle \Sigma, Op, \alpha_G \rangle$

- Global selector variables $V_s = \{s_{\epsilon \preceq \epsilon'} : \epsilon, \epsilon' \in Op\} \subseteq Atm$
- φ_{V_s} denotes the conjunction of the following axioms:

$$\bigwedge_{\epsilon \in Op} \left(\neg s_{\epsilon \preceq \epsilon} \rightarrow \bigwedge_{\substack{\epsilon' \in Op \\ \epsilon \neq \epsilon'}} \neg s_{\epsilon \preceq \epsilon'} \wedge \neg s_{\epsilon' \preceq \epsilon} \right) \tag{S1}$$

$$\bigwedge_{\epsilon \in Op} \bigwedge_{\substack{\epsilon' \in Op \\ \epsilon \neq \epsilon'}} \left(\neg s_{\epsilon \preceq \epsilon'} \vee \neg s_{\epsilon' \preceq \epsilon} \right) \tag{S2}$$

$$\bigwedge_{\epsilon \in Op} \bigwedge_{\substack{\epsilon' \in Op \\ \epsilon \neq \epsilon'}} \bigwedge_{\substack{\epsilon'' \in Op \\ \epsilon \neq \epsilon'' \\ \epsilon' \neq \epsilon''}} \left(s_{\epsilon \preceq \epsilon'} \wedge s_{\epsilon' \preceq \epsilon''} \rightarrow s_{\epsilon \preceq \epsilon''} \right) \tag{S3}$$

∃∀ encoding of ES-INF planning problem $\langle \Sigma, Op, \alpha_G \rangle$

- φ_{V_s} denotes the conjunction of the following axioms:

$$\bigwedge_{\epsilon \in Op} \bigwedge_{\substack{\epsilon' \in Op \\ \epsilon \neq \epsilon'}} \left(s_{\epsilon \preceq \epsilon} \wedge s_{\epsilon' \preceq \epsilon'} \rightarrow s_{\epsilon \preceq \epsilon'} \vee s_{\epsilon' \preceq \epsilon} \right) \tag{S4}$$

$$\bigwedge_{\epsilon \in Op} \bigwedge_{\epsilon' \in Op} \left(s_{\epsilon \preceq \epsilon'} \leftrightarrow \Box_{\mathfrak{m}} \, s_{\epsilon \preceq \epsilon'} \right) \tag{S5}$$

Dynamic extension (recall)

The following equivalences are valid in the class $\mathbf{M}$:

$$[+_i\alpha]\alpha' \leftrightarrow \begin{cases} \top, & \text{if } \alpha' = \triangle_i\alpha, \\ \alpha', & \text{otherwise;} \end{cases}$$

$$[+_i\alpha]\neg\varphi \leftrightarrow \neg[+_i\alpha]\varphi;$$

$$[+_i\alpha](\varphi_1 \wedge \varphi_2) \leftrightarrow [+_i\alpha]\varphi_1 \wedge [+_i\alpha]\varphi_2;$$

$$[+_i\alpha](\varphi_1 \vee \varphi_2) \leftrightarrow [+_i\alpha]\varphi_1 \vee [+_i\alpha]\varphi_2;$$

$$[+_i\alpha]\,\Box_{\mathfrak{m}}\,\alpha' \leftrightarrow \begin{cases} \Box_{\mathfrak{m}}(\alpha \to \alpha'), & \text{if } i = \mathfrak{m}, \\ \Box_{\mathfrak{m}}\,\alpha', & \text{otherwise;} \end{cases}$$

$$[+_i\alpha]\,\Diamond_{\mathfrak{m}}\,\alpha' \leftrightarrow \begin{cases} \Diamond_{\mathfrak{m}}(\alpha \wedge \alpha'), & \text{if } i = \mathfrak{m}, \\ \Diamond_{\mathfrak{m}}\,\alpha', & \text{otherwise.} \end{cases}$$

∃∀ encoding of ES-INF planning problem

- For $\epsilon \in Op$, we define the function $\Pi_{\preceq\epsilon} : \mathcal{L}_{\text{Frag}} \longrightarrow \mathcal{L}_{\text{Frag}}$.

$$\Pi_{\preceq\epsilon}(p) = p,$$

$$\Pi_{\preceq\epsilon}(\alpha) = \begin{cases} s_{\epsilon' \preceq \epsilon} \vee \triangle_{\mathrm{m}} \alpha_{\epsilon'}, & \text{if } \exists \epsilon' \neq \epsilon : \alpha = \triangle_{\mathrm{m}} \alpha_{\epsilon'}, \\ \alpha, & \text{otherwise;} \end{cases}$$

$$\Pi_{\preceq\epsilon}(\neg\varphi) = \neg\Pi_{\preceq\epsilon}(\varphi),$$

$$\Pi_{\preceq\epsilon}(\varphi_1 \wedge \varphi_2) = \Pi_{\preceq\epsilon}(\varphi_1) \wedge \Pi_{\preceq\epsilon}(\varphi_2),$$

$$\Pi_{\preceq\epsilon}(\varphi_1 \vee \varphi_2) = \Pi_{\preceq\epsilon}(\varphi_1) \vee \Pi_{\preceq\epsilon}(\varphi_2),$$

$$\Pi_{\preceq\epsilon}(\Box_{\mathrm{m}}\, \alpha) = \Box_{\mathrm{m}} \left(\left(\bigvee_{\substack{\epsilon' \in Op \\ \epsilon \neq \epsilon'}} s_{\epsilon' \preceq \epsilon} \wedge \neg\alpha_{\epsilon'} \right) \vee \alpha \right)$$

$$\Pi_{\preceq\epsilon}(\Diamond_{\mathrm{m}}\, \alpha) = \Diamond_{\mathrm{m}} \left(\bigwedge_{\substack{\epsilon' \in Op \\ \epsilon \neq \epsilon'}} \left(\neg s_{\epsilon' \preceq \epsilon} \vee \alpha_{\epsilon'} \right) \wedge \alpha \right)$$

∃∀ encoding of ES-INF planning problem

Let $F_{\langle \Sigma, Op, \alpha_G \rangle}$ be the reduction in $\mathcal{L}_{\text{Prop}}$ of the following formula of $\mathcal{L}_{\text{Frag}}$:

$$\varphi_{V_s} \wedge \left(\left(\bigwedge_{\alpha \in \Sigma} \Box_{\mathfrak{m}} \alpha \right) \rightarrow \bigwedge_{\epsilon \in Op} \left(s_{\epsilon \preceq \epsilon} \rightarrow \Pi_{\preceq \epsilon}(\mathcal{P}(\epsilon)) \right) \wedge \varphi_P \right)$$

Theorem

The ES-INF planning problem $\langle \Sigma, Op, \alpha_G \rangle$ has a solution plan iff the following quantified boolean formula is true:

$$Q = \exists_{s_{\epsilon \preceq \epsilon'} \in V_s} s_{\epsilon \preceq \epsilon'} \quad \forall_{p \in Vars(F_{\langle \Sigma, Op, \alpha_G \rangle}) \setminus V_s} p \quad F_{\langle \Sigma, Op, \alpha_G \rangle}$$

Part 2 - QBF encoding of cognitive planning and TouIST platform

ToulST Platform

Motivation for a new tool

Remark 1: existing languages are not intuitive

- ⊕ SAT solvers are very efficient
- ⊖ but with a “low level” input language

Formula	Solver input (DIMACS)

$a \rightarrow$ $b \vee \neg (a \rightarrow c)$ $\vee (c \rightarrow \neg a)$	`p cnf 5 4`
	`-1 5 3 0`
	`-1 -5 4 1 0`
	`-1 -5 4 -2 0`
	`-1 -3 -2 -1 0`

Motivation for a new tool

Remark 2: existing languages are not compact

- ⊕ SMT-LIB is much more expressive than DIMACS
- ⊖ but with a lack of compacity

Formula	SMT-LIB example
$\bigwedge_{i \in [1..1000]} x_i$	`(assert (and x1 (and x2 (and x3 (and x4 (and x5 (and x6 ... x1000)))))))`

Motivation for a new tool

Technical goal:

- an intuitive language for modeling
- a user-friendly graphical interface

Academic goal:

- teaching of logic
- research (comparison of solvers, encodings)

Principle of use for reasoning on a little example

Example of the politician

If growth increases, then unemployment decreases ;
Now, growth decreases.
Then unemployment increases !

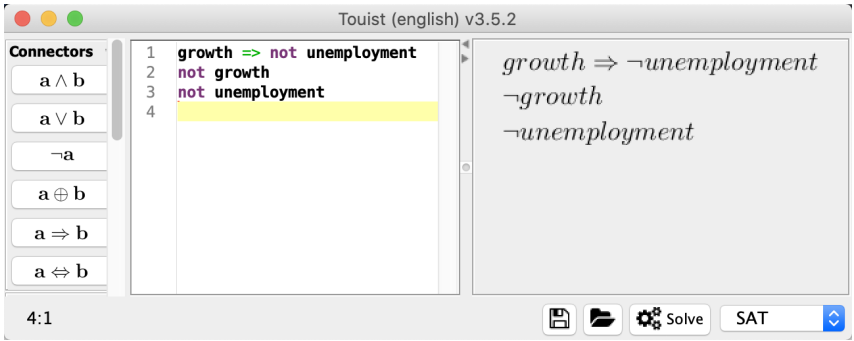
Problem

Is the conclusion justified ?

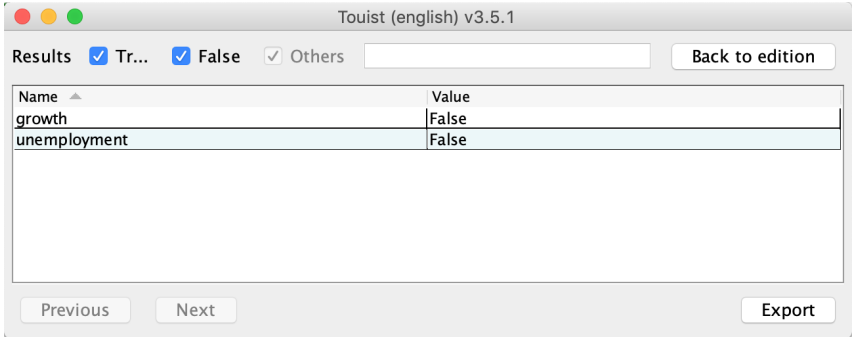
$$\{\text{growth} \rightarrow \neg\text{unemployment}, \neg\text{growth}\} \models \text{unemployment}$$

Principle of use for reasoning on a little example

$$\{\text{growth} \rightarrow \neg\text{unemployment}, \neg\text{growth}\} \models \text{unemployment}$$



Principle of use for reasoning on a little example



A model is found, then the reasoning **is not valid !**

$\{\text{growth} \rightarrow \neg\text{unemployment}, \neg\text{growth}\} \not\models \text{unemployment}$

Implementation aspects

TouIST (Toulouse Integrated Satisfiability Tool)

Graphical interface for writing logical formulas
in a **compact language** and using different **external solvers**.

- under MIT license, sources : `github.com/touist/touist`
- graphical interface (**Java**)
- translation to solver languages by a compiler (**OCaml**)
- compatible with **SAT, SMT and QBF**

Implementation aspects

Compatible solvers :

Solver	Formula	TouIST input language
SAT	$a \wedge b \rightarrow c$	`a and b => c`
SMT (QF-LIA)	$(x + y - z \geq 4) \vee b$	`(x + y - z >= 4) or a`
SMT (QF-LRA)	$(x + y - z > 4.0) \rightarrow c$	`(x + y - z > 4.0) => c`
SMT (QF-IDL)	$(x - y > 2) \wedge c$	`(x - y > 2) and c`
SMT (QF-RDL)	$(x - y > 1.0) \wedge c$	`(x - y > 1.0) and c`
QBF	$\forall b. a \wedge b$	`forall b: a and b`

QF-LIA = Quantifier-Free Linear Integer Arithmetic

QF-LRA = Quantifier-Free Linear Rational Arithmetic

QF-IDL = Quantifier-Free Integer Difference Logic

QF-RDL = Quantifier-Free Rational Difference Logic

Implementation aspects

ToulST input language :

- variables and sets

```
$N = 5
$Rows = [1..$N]
$Countries = [France, Italy, TheNetherlands]
```

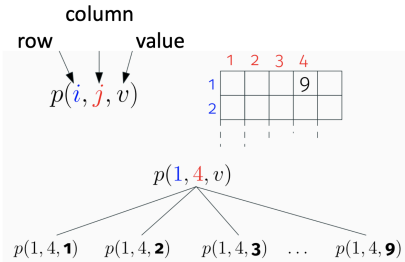
- generalized connectors

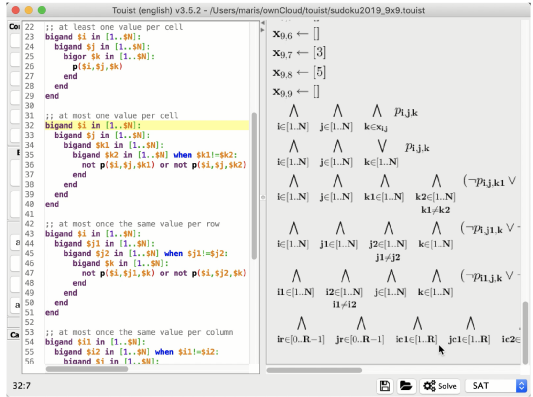
```
in(France) and in(Italy) and in(TheNetherlands)

bigand $c in $Countries:
  in($c)
end
```

Solving Puzzles with ToulST (SAT)

	2					4	8	
1			8		2	3		7
				9	3	2	1	
	4	8						1
			6	1	8			
7						9	3	
	1	5	9	4				
4		2	3		6			9
	3	7					5	



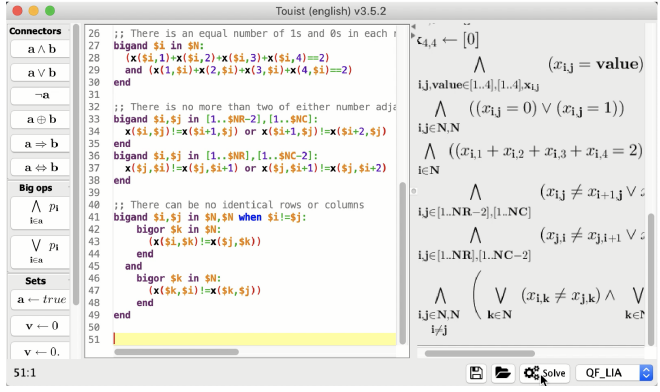


Solving Puzzles with TouIST (SMT)

	1		0
		0	
	0		
1	1		0

0	1	1	0
1	0	0	1
0	0	1	1
1	1	0	0

A compact SMT encoding
with atoms of QF_LIA
(linear arithmetic on integers)



Solving an ES-INF planning problem with TouIST

- **SAT solver:** to verify the validity of a plan candidate

$$\left(\bigwedge_{\alpha \in \Sigma} \Box_{\mathfrak{m}} \alpha \right) \rightarrow \langle\!\langle \epsilon_1 \rangle\!\rangle \dots \langle\!\langle \epsilon_{k-1} \rangle\!\rangle \langle\!\langle \epsilon_k \rangle\!\rangle \Box_{\mathfrak{m}} \alpha_G$$

$$\mathcal{L}_{\mathsf{Frag}} \overset{nnf}{\dashrightarrow} \mathcal{L}^{\mathsf{NNF}}_{\mathsf{Frag}} \overset{tr_1}{\dashrightarrow} \mathcal{L}_{\mathsf{Mod}} \overset{tr_2}{\dashrightarrow} \mathcal{L}_{\mathsf{Prop}}$$

- **QBF solver:** to extract a solution plan from $\exists\forall$ encoding

$$Q = \underset{s_{\epsilon \preceq \epsilon'} \in V_s}{\exists} s_{\epsilon \preceq \epsilon'} \quad \underset{p \in Vars(F_{\langle \Sigma, Op, \alpha_G \rangle}) \setminus V_s}{\forall} p \quad F_{\langle \Sigma, Op, \alpha_G \rangle}$$

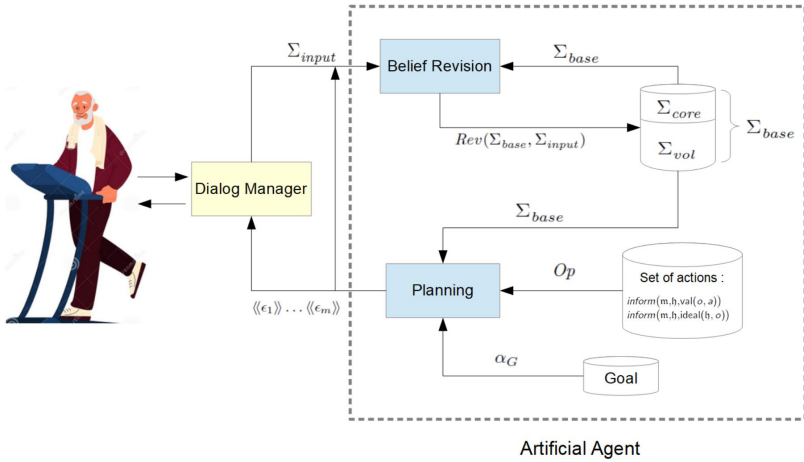
Plan of the tutorial

1. Logical framework for cognitive planning (Emiliano)
2. QBF encoding of cognitive planning and ToulST platform (Frédéric)
3. **Implementation and applications (Jorge) NOW!**
   - Recommender system for sport activities
   - Conversational agent for motivational interviewing
4. Formal extension and implementation for collaborative gaming (Dominique & Jorge)

Part 3 - Implementation and applications

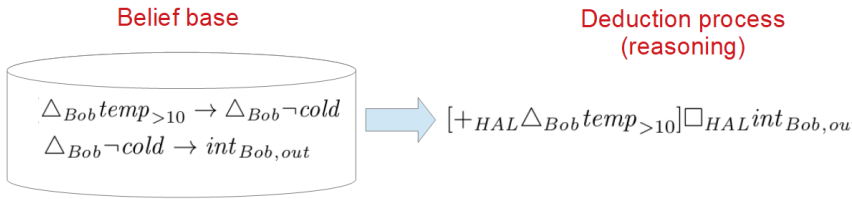
Introduction

System Architecture

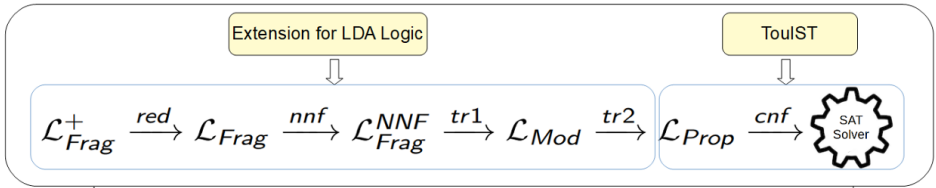


Deduction process

We consider a conversational agent that has to interact with a human user in order to support her activity and to take care of her well-being. Specifically, HAL is an artificial companion which takes care of an elderly person called Bob and keeps him company. Bob has to do regular physical activity to be in good health.



Checking satisfiability of formulas in $\mathcal{L}_{\text{Frag}}$



Part 3 - Implementation and applications

Recommender system for sport activities

Example 1: Artificial assistant

- Agent $\mathfrak{h}$: human user who has to choose a sport to practice
- Agent $\mathfrak{m}$: artificial assistant
- Agent $\mathfrak{m}$'s goal: agent $\mathfrak{h}$ forms the intention to practice a sport
- Solution plan: sequence of speech acts by $\mathfrak{m}$

Example 1: Artificial assistant(cont.)

Opt	*Var*					
	env	**loc**	**soc**	**cost**	**dan**	**intens**
sw	*water*	*mixed*	*single*	*med*	*low*	*high*
ru	*land*	*outdoor*	*single*	*low*	*med*	*high*
hr	*land*	*outdoor*	*single*	*high*	*high*	*low*
te	*land*	*mixed*	*mixed*	*high*	*med*	*med*
so	*land*	*mixed*	*team*	*med*	*med*	*med*
yo	*land*	*mixed*	*single*	*med*	*low*	*low*
di	*water*	*mixed*	*single*	*high*	*high*	*low*
sq	*land*	*indoor*	*mixed*	*high*	*med*	*med*

Table: For every option $o \in Opt$ and variable $x \in Var$, we denote by $v_{o,x}$ the corresponding entry in the table.

Example 1: Artificial assistant(cont.)

$$\alpha_1 \stackrel{\text{def}}{=} \bigwedge_{\substack{o\in Opt\\ x\in Var\\ v,v'\in Val_x: v\neq v'}} \big(\mathsf{val}(o, x\mapsto v) \rightarrow \neg\mathsf{val}(o, x\mapsto v')\big),$$

$$\alpha_2 \stackrel{\text{def}}{=} \bigwedge_{\substack{o\in Opt\\ x\in Var\\ v,v'\in Val_x: v\neq v'}} \big(\triangle_{\mathfrak{h}}\mathsf{val}(o, x\mapsto v) \rightarrow \triangle_{\mathfrak{h}}\neg\mathsf{val}(o, x\mapsto v')\big),$$

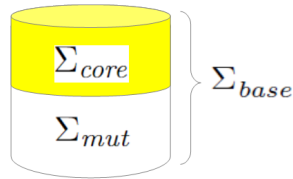
$$\alpha_3 \stackrel{\text{def}}{=} \bigwedge_{\Gamma,\Gamma'\in 2^{Des*}: \Gamma\neq\Gamma'} \big(\mathsf{des}(\mathfrak{h},\Gamma) \rightarrow \neg\mathsf{des}(\mathfrak{h},\Gamma')\big),$$

$$\alpha_4 \stackrel{\text{def}}{=} \bigvee_{\Gamma\in 2^{Des*}} \mathsf{des}(\mathfrak{h},\Gamma),$$

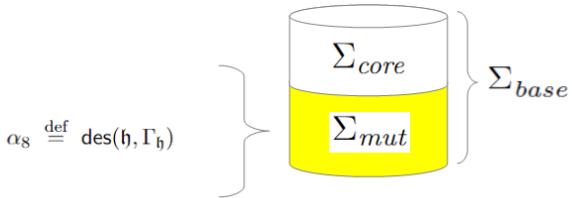
$$\alpha_5 \stackrel{\text{def}}{=} \bigwedge_{o\in Opt} \big(\mathsf{ideal}(\mathfrak{h}, o) \leftrightarrow \bigvee_{\Gamma\in 2^{Des*}} \big(\mathsf{des}(\mathfrak{h},\Gamma) \wedge \bigwedge_{\gamma\in\Gamma} f_{comp}(o,\gamma)\big)\big),$$

$$\alpha_6 \stackrel{\text{def}}{=} \bigwedge_{o\in Opt} \Big(\mathsf{justif}(\mathfrak{h}, o) \leftrightarrow \bigvee_{\Gamma\in 2^{Des*}} \big(\mathsf{des}(\mathfrak{h},\Gamma) \wedge \bigwedge_{\gamma\in\Gamma} f^{\mathfrak{h}}_{comp}(o,\gamma)\big)\Big).$$

$$\alpha_7^{o,x} \stackrel{\text{def}}{=} \mathsf{val}(o, x\mapsto v_{o,x}).$$



Example 1: Artificial assistant(cont.)



$$\Gamma_\mathfrak{h} = \{\mathbf{env} \mapsto land, \mathbf{intens} \mapsto med, \sim\mathbf{loc} \mapsto indoor, [\mathbf{cost} \mapsto high] \rightsquigarrow \mathbf{soc} \mapsto mixed\}.$$

Potential intention: $\text{potIntend}(\mathfrak{h}, o) \stackrel{\text{def}}{=} \triangle_\mathfrak{h}\text{ideal}(\mathfrak{h}, o) \wedge \text{justif}(\mathfrak{h}, o)$

Example 1: Artificial assistant(cont.)

Operators:

$$Op = \{ inform(\mathfrak{m},\mathfrak{h},\mathsf{val}(o,a)) : o \in Opt \text{ and } a \in Assign \} \cup \{ inform(\mathfrak{m},\mathfrak{h},\mathsf{ideal}(\mathfrak{h},o)) : o \in Opt \}$$

Executability preconditions:

$$\mathcal{P}\Big(inform(\mathfrak{m},\mathfrak{h},\mathsf{val}(o,a))\Big) = \Box_{\mathfrak{m}} \Big(\mathsf{val}(o,a) \wedge \bigwedge_{v \in Val_{\mathbf{dan}}} (\mathsf{val}(o,\mathbf{dan} \mapsto v) \rightarrow \triangle_{\mathfrak{h}} \mathsf{val}(o,\mathbf{dan} \mapsto v))\Big)$$

$$\text{if } a \notin Assign_{\mathbf{dan}},$$

$$\mathcal{P}\Big(inform(\mathfrak{m},\mathfrak{h},\mathsf{val}(o,a))\Big) = \Box_{\mathfrak{m}}\, \mathsf{val}(o,a)$$

$$\text{if } a \in Assign_{\mathbf{dan}},$$

$$\mathcal{P}\Big(inform(\mathfrak{m},\mathfrak{h},\mathsf{ideal}(\mathfrak{h},o))\Big) = \Box_{\mathfrak{m}} \big(\mathsf{ideal}(\mathfrak{h},o) \wedge \mathsf{justif}(\mathfrak{h},o)\big).$$

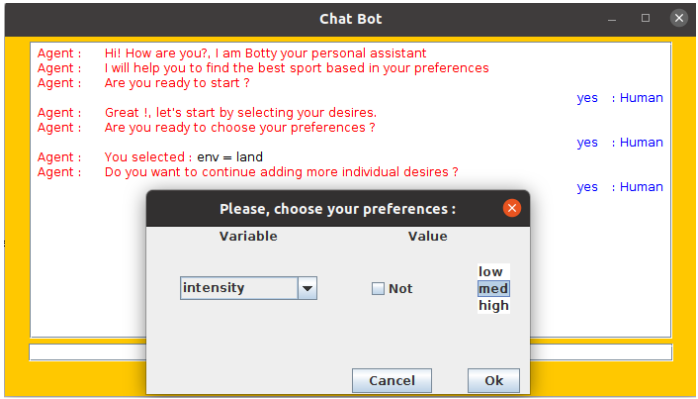
Example 1: Artificial assistant(cont.)

Planning goal: $\alpha_G \stackrel{\text{def}}{=} \bigvee_{o \in Opt} \mathsf{potIntend}(\mathfrak{h}, o)$

$\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4, \epsilon_5, \epsilon_6$ is a solution for the planning problem $\langle \Sigma, Op, \alpha_G \rangle$ where: $\Sigma = \{\alpha_1, \ldots, \alpha_8\} \cup \{\alpha_7^{o,x} : o \in Opt \text{ and } x \in Var\}$ and

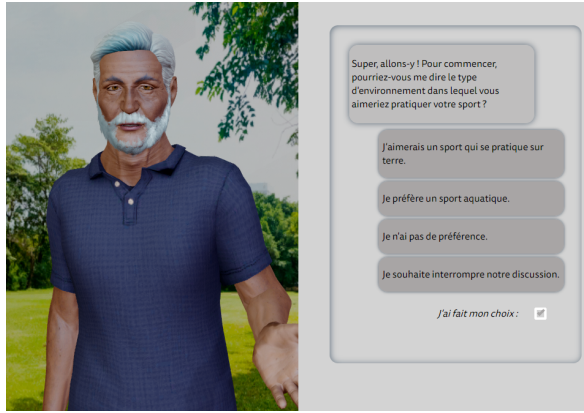
$$\epsilon_1 \stackrel{\text{def}}{=} inform\big(\mathfrak{m},\mathfrak{h},\mathsf{val}(\mathrm{te}, \mathbf{dan} \mapsto med)\big),$$
$$\epsilon_2 \stackrel{\text{def}}{=} inform\big(\mathfrak{m},\mathfrak{h},\mathsf{val}(\mathrm{te}, \mathbf{env} \mapsto land)\big),$$
$$\epsilon_3 \stackrel{\text{def}}{=} inform\big(\mathfrak{m},\mathfrak{h},\mathsf{val}(\mathrm{te}, \mathbf{intens} \mapsto med)\big),$$
$$\epsilon_4 \stackrel{\text{def}}{=} inform\big(\mathfrak{m},\mathfrak{h},\mathsf{val}(\mathrm{te}, \mathbf{loc} \mapsto mixed)\big),$$
$$\epsilon_5 \stackrel{\text{def}}{=} inform\big(\mathfrak{m},\mathfrak{h},\mathsf{val}(\mathrm{te}, \mathbf{soc} \mapsto mixed)\big),$$
$$\epsilon_6 \stackrel{\text{def}}{=} inform\big(\mathfrak{m},\mathfrak{h},\mathsf{ideal}(\mathfrak{h}, \mathrm{te})\big).$$

Demo



- Source code:
  `https://www.irit.fr/CoPains/software/`

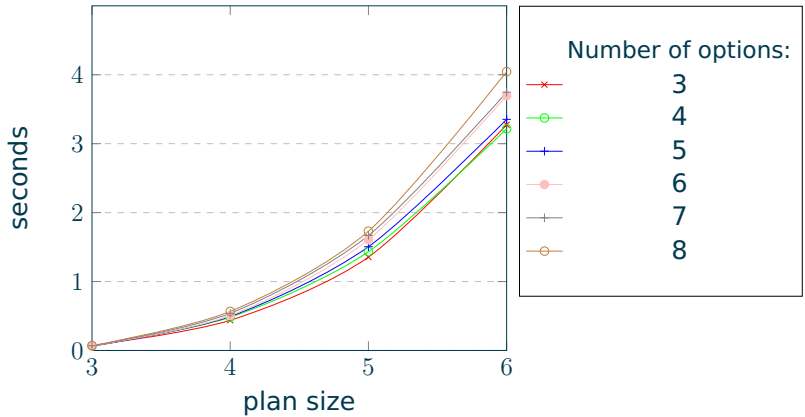
Web interface



- Web avatar developed in collaboration with the company DAVI: `https://cognitive-planning.schm.fr/`

Experiments

Processing time used by the brute force approach:



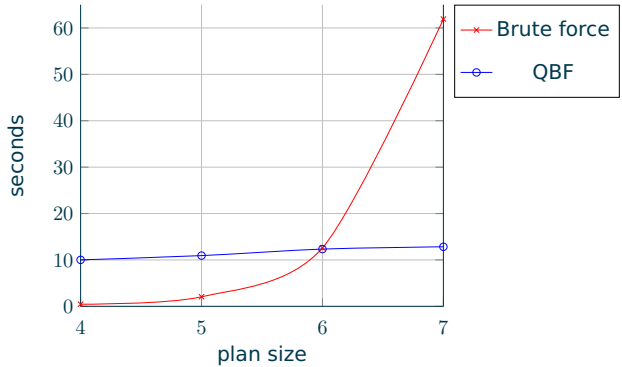
Experiments(cont.)

Processing time (in seconds) used by the brute force approach:

Plan size	Number of options (sports)					
	3	4	5	6	7	8
3	0.059	0.067	0.063	0.066	0.068	0.070
4	0.438	0.482	0.494	0.506	0.539	0.567
5	1.355	1.433	1.505	1.608	1.668	1.731
6	3.274	3.217	3.353	3.696	3.747	4.045

Experiments(cont.)

Comparison in performance between the brute force and the QBF approaches:



Experiments(cont.)

Comparison in performance between the brute force and the QBF approaches:

Size of the plan	Planning module processing time (seconds)	
	Brute force	QBF
4	0.464	10.015
5	2.059	10.936
6	12.610	12.343
7	61.920	12.837

Conclusions

- Our implementation demonstrates that our NP-complete fragment and the cognitive planning problem formulated in this logic are suitable for real-world applications in the domain of HMI.
- We intend to include a setting parameter in the artificial agent in order to let the system select the most convenient approach (SAT or QBF) depending on the scenario.
- We plan to extend the implemented system by speech acts of type question to capture both sides of interaction, from agent m to agent h (handled by the actual implementation) and from agent h to agent m.

Part 3 - Implementation and applications

Conversational agent for motivational interviewing

Example 2: Virtual coaching

Agent $\mathfrak{m}$ is a virtual coaching agent which has to motivate agent $\mathfrak{h}$ to practice a physical activity.

$\Rightarrow$ Agent $\mathfrak{h}$ has to become aware of the inconsistency between her current behavior and her desires

Desires : $DesAtm$	Conditions : $CondAtm$	Actions : $ActAtm$
dr: "$\mathfrak{h}$ has dietary restrictions" pw: "$\mathfrak{h}$ puts on weight" lw: "$\mathfrak{h}$ loses weight" at: "$\mathfrak{h}$ is attractive" gh: "$\mathfrak{h}$ is in good health" st: "$\mathfrak{h}$ is stressed"	ow: "$\mathfrak{h}$ has an office work" sl: "$\mathfrak{h}$ has a sedentary life style' co: "$\mathfrak{h}$ is a commuter/time in traffic"	does($\mathfrak{h}$,ps^*)

*ps :"$\mathfrak{h}$ practices sport"

Example 2: Virtual coaching(cont.)

Agent $\mathfrak{m}$'s model of agent $\mathfrak{h}$'s cognitive state:

$$\alpha_1 \stackrel{\text{def}}{=} \bigwedge_{l \in Lit} \triangle_{\mathfrak{h}} \mathsf{nec}(\{l\}, l),$$

$$\alpha_2 \stackrel{\text{def}}{=} \bigwedge_{l \in Lit} \big(\triangle_{\mathfrak{h}} \mathsf{nec}(\emptyset, l) \leftrightarrow \triangle_{\mathfrak{h}} l\big),$$

$$\alpha_3 \stackrel{\text{def}}{=} \bigwedge_{l \in Lit, X, X' \in LitSet : X' \subseteq X} \Big(\triangle_{\mathfrak{h}} \mathsf{nec}(X, l) \rightarrow \big(\bigwedge_{l' \in X'} \triangle_{\mathfrak{h}} l' \rightarrow \triangle_{\mathfrak{h}} \mathsf{nec}(X \setminus X', l)\big)\Big),$$

$$\alpha_4 \stackrel{\text{def}}{=} \bigwedge_{l \in Lit, X, X' \in LitSet : X \subseteq X'} \big(\triangle_{\mathfrak{h}} \mathsf{nec}(X, l) \rightarrow \triangle_{\mathfrak{h}} \mathsf{nec}(X', l)\big),$$

$$\alpha_5 \stackrel{\text{def}}{=} \bigwedge_{l \in Lit} \Big(\big(\mathsf{des}(\mathfrak{h}, l) \leftrightarrow \triangle_{\mathfrak{h}} \mathsf{des}(\mathfrak{h}, l)\big) \wedge \big(\neg \mathsf{des}(\mathfrak{h}, l) \leftrightarrow \triangle_{\mathfrak{h}} \neg \mathsf{des}(\mathfrak{h}, l)\big)\Big),$$

$$\alpha_6 \stackrel{\text{def}}{=} \bigwedge_{a \in Act} \Big(\big(\mathsf{does}(\mathfrak{h}, a) \leftrightarrow \triangle_{\mathfrak{h}} \mathsf{does}(\mathfrak{h}, a)\big) \wedge \big(\neg \mathsf{does}(\mathfrak{h}, a) \leftrightarrow \triangle_{\mathfrak{h}} \neg \mathsf{does}(\mathfrak{h}, a)\big)\Big),$$

Example 2: Virtual coaching(cont.)

$$\begin{aligned}\alpha_7 \stackrel{\text{def}}{=} & \ \mathsf{nec}\big(\{\neg dr, \neg pw, sl\}, \mathsf{does}(\mathfrak{h},ps)\big)\wedge \\ & \ \mathsf{nec}\big(\{at, \neg dr\}, \mathsf{does}(\mathfrak{h},ps)\big)\wedge \\ & \ \mathsf{nec}\big(\{sl, gh\}, \mathsf{does}(\mathfrak{h},ps)\big)\wedge \\ & \ \mathsf{nec}\big(\{gh\}, \neg st\big)\wedge \\ & \ \mathsf{nec}\big(\{co, ow\}, sl\big).\end{aligned}$$

$$\alpha_G \stackrel{\text{def}}{=} \neg\mathsf{does}(\mathfrak{h},ps) \rightarrow \mathsf{AwareIncon}\big(\mathfrak{h}, \neg\mathsf{does}(\mathfrak{h},ps)\big).$$

Plan computed by agent $\mathfrak{m}$:

Step	Informative planning problem	Interrogative planning problem
0		$?_{\mathfrak{m},\mathfrak{h}}\mathsf{does}(\mathfrak{h},ps)$
1		$?_{\mathfrak{m},\mathfrak{h}}\mathsf{des}(\mathfrak{h}, gh), ?_{\mathfrak{m},\mathfrak{h}}co, ?_{\mathfrak{m},\mathfrak{h}}ow$
2	$!_{\mathfrak{m},\mathfrak{h}}(\emptyset,\{co, ow\},sl), !_{\mathfrak{m},\mathfrak{h}}\big(\{gh\},\{sl\},\mathsf{does}(\mathfrak{h},ps)\big)$	

Example 2: Virtual coaching(cont.)

Speaker	Utterance	Speech act
$\mathfrak{m}$	Do you practice a sport regularly?	$?_{\mathfrak{m},\mathfrak{h}}\text{does}(\mathfrak{h},ps)$
$\mathfrak{h}$	I don't	$+_{\mathfrak{m}}\neg\triangle_{\mathfrak{h}}\text{does}(\mathfrak{h},ps)$
$\mathfrak{m}$	Do you wish to be in good health?	$?_{\mathfrak{m},\mathfrak{h}}\text{des}(\mathfrak{h},\ gh)$
$\mathfrak{h}$	Yes	$+_{\mathfrak{m}}\triangle_{\mathfrak{h}}\text{des}(\mathfrak{h},\ gh)$
$\mathfrak{m}$	Do you spend quite some time in the traffic everyday as a commuter?	$?_{\mathfrak{m},\mathfrak{h}}\,co$
$\mathfrak{h}$	Yes	$+_{\mathfrak{m}}\triangle_{\mathfrak{h}}\,co$
$\mathfrak{m}$	Do you have an office work?	$?_{\mathfrak{m},\mathfrak{h}}\,ow$
$\mathfrak{h}$	Yes	$+_{\mathfrak{m}}\triangle_{\mathfrak{h}}\,ow$
$\mathfrak{m}$	You spend quite some time in the traffic everyday as a commuter and you have an office work. Therefore, your life style is sedentary!	$!_{\mathfrak{m},\mathfrak{h}}(\emptyset,\{co,\ ow\},sl)$
$\mathfrak{m}$	Your life style is sedentary. Therefore, you will not satisfy your desire to be in good health unless you practice a sport regularly!	$!_{\mathfrak{m},\mathfrak{h}}\big(\{gh\},\{sl\},$ $\text{does}(\mathfrak{h},ps)\big)$

Plan of the tutorial

1. Logical framework for cognitive planning (Emiliano)
2. QBF encoding of cognitive planning and TouIST platform (Frédéric)
3. Implementation and applications (Jorge)
4. **Formal extension and implementation for collaborative gaming (Dominique & Jorge) NOW!**
   - Extended logical framework and encoding of the Yōkai game
   - General architecture and implementation

Part 4 - Application to the Yōkai game

Introduction

Why Yōkai game?

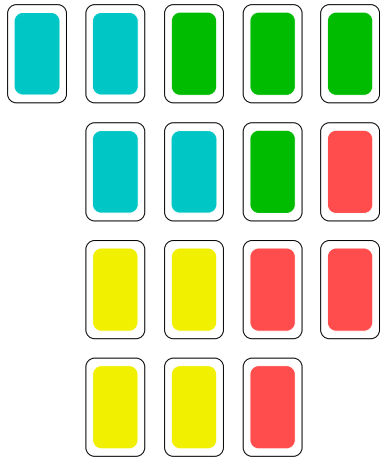
Because this game includes:

- reasoning about mental states (theory of mind),
- temporal reasoning,
- spatial reasoning.

Rules of Yōkai game [Griffon, 2019]

- 4 card colors and 4 cards of each color,
- 7 hints,
- initial state: shuffled and grouped cards face down,
- target end state (goal): group cards of the same color,
- 4 kinds of action:
  - to observe (the color of) a card,
  - to move a card (face down),
  - to active a hint,
  - to mark a card with a hint,
- constraint: communication outside of actions is not allowed.

Example of a winning final configuration



A round of play

At each round of play, each player execute 4 actions (no order constraint) :

- (to observe 1 card) $\times$ 2 ;
- to move 1 card (without creating a *separation*) ;
- to active 1 hint (x)or to mark 1 card with a hint about its color ($\Rightarrow$ marked cards cannot be moved anymore).

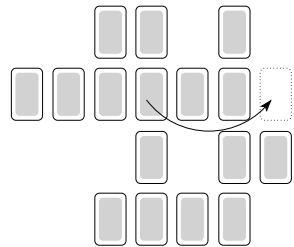
What is a *separation* (between groups of cards)?

Definition (separation)

There are at least two cards that cannot be linked by a “path”

Definition (path)

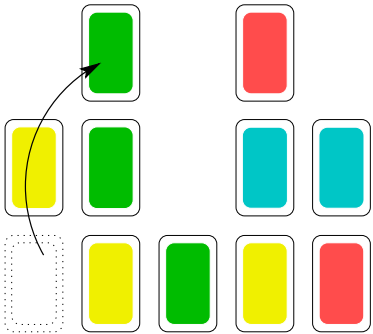
A set of positions occupied by cards which are accessed only from their sides (NOT diagonally).



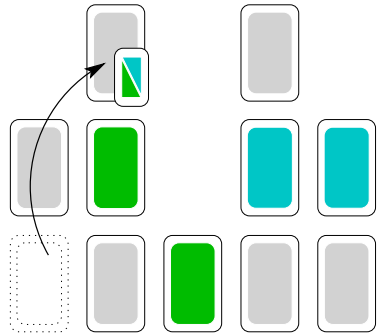
Hints

- an hint is a card containing 1 to 3 colors,
- “to mark a card” is to place a hint on this card,
- a card can only be marked with a hint including the color of the card,
- once marked, a card can no longer be observed, moved or marked.

Example of game in progress ($\mathfrak{m}$ plays against $\mathfrak{h}$)



Cards of which $\mathfrak{m}$ knows the color



Cards that $\mathfrak{m}$ believes $\mathfrak{h}$ knows the color of

End of game

- 2 ways to complete the game:
  - either because a player declares the end of the game,
  - or because there is no more hint to activate or to mark a card.

Reminder

- Each round, a hint must be activated or played (to mark a card),
- there are 7 hints,
- there are a maximum of 14 rounds of play (7 for each player).

Part 4 - Application to the Yōkai game

Formal framework extension

Temporal extension of explicit beliefs

We first define the language $\mathcal{L}_0(Atm)$ by:

$$\alpha \quad ::= \quad \underbrace{p^t \mid \triangle^t_{\mathfrak{h}}\,\alpha \mid now^{\geq t}}_{\mathcal{L}_0^T(Atm)} \mid \neg\alpha \mid \alpha_1 \wedge \alpha_2 \mid \triangle_{\mathfrak{m}}\,\alpha$$

where :

- p ranges over Atm,
- p^t is read “atomic proposition p is true at time t”,
- $now^{\geq t}$ is read “the actual time of the game play is at least t”.

Implicit beliefs (no change)

Language $\mathcal{L}(Atm)$ is defined by:

$$\varphi \quad ::= \quad \alpha \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \Box_{\mathrm{m}}\, \alpha$$

Note: only formulas from $\mathcal{L}_0$ can be in the scope of an $\Box_{\mathrm{m}}$ operator.

Semantics (reminder & new things)

- State: $S = (B, V)$ with
  - $B \subseteq \mathcal{L}_0$ is agent $\mathfrak{m}$'s belief base ,
  - $V \subseteq \mathcal{L}_0^T$ is the actual situation,
- for every $t, t' \in \mathbb{N}$:
  - $now^{\geq 0} \in V$,
  - if $now^{\geq t} \in V$ and $t' \leq t$ then $now^{\geq t'} \in V$,
  - $now^{\geq t} \in V$ iff $now^{\geq t} \in B$.

Semantics (new things)

Let $S = (B, V) \in \mathbf{S}$. Then, for every $x \in \mathcal{L}_0^T(Atm)$:

$$
\begin{aligned}
S \models x &\iff x \in V, \\
S \models \neg\alpha &\iff S \not\models \alpha, \\
S \models \alpha_1 \wedge \alpha_2 &\iff S \models \alpha_1 \text{ and } S \models \alpha_2, \\
S \models \triangle_{\mathfrak{m}}\, \alpha &\iff \alpha \in B.
\end{aligned}
$$

Dynamic extension

- Language $\mathcal{L}^{+}(Atm)$ is defined as follows:

$$\varphi \quad ::= \quad \alpha \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \Box_{\mathfrak{m}}\,\alpha \mid [+^{t}_{\mathfrak{m}}\alpha]\varphi,$$

- $[+^{t}_{\mathfrak{m}}\alpha]\varphi$ is read "φ holds after agent $\mathfrak{m}$ has privately learned that α and that the current time is at least t".
- $(S, Cxt) \models [+^{t}_{\mathfrak{m}}\alpha]\varphi \iff (S^{+^{t}_{\mathfrak{m}}\alpha}, Cxt) \models \varphi$ with

$$S^{+^{t}_{\mathfrak{m}}\alpha} = (B^{+^{t}_{\mathfrak{m}}\alpha}, V^{+^{t}_{\mathfrak{m}}\alpha}),$$
$$V^{+^{t}_{\mathfrak{m}}\alpha} = V \cup \{now^{\geq t'} : t' \leq t\},$$
$$B^{+^{t}_{\mathfrak{m}}\alpha} = B \cup \{\alpha\} \cup \{now^{\geq t'} : t' \leq t\}.$$

Theorems

- Checking satisfiability of $\mathcal{L}(Atm)$ formulas in the class $\mathbf{M}$ is an NP-complete problem.
- Checking satisfiability of $\mathcal{L}^{+}(Atm)$ formulas in the class $\mathbf{M}$ is an NP-complete problem.
- Let $\varphi \in \mathcal{L}^{+}(Atm)$ and let $\Sigma \subset \mathcal{L}_0(Atm)$ be finite. Then

  $\Sigma \models_{\mathbf{M}} \varphi$ if and only if $\models_{\mathbf{M}} \bigwedge_{\alpha \in \Sigma} \Box_{\mathfrak{m}} \alpha \rightarrow \varphi$.

  (Logical consequence problem with a finite set of premises can be reduced to the satisfiability problem.)

Action selection (rappel)

- **Belief expansion events:** $Act_{\mathfrak{m}} = \{+^t_{\mathfrak{m}}\alpha : \alpha \in \mathcal{L}_0, t \in \mathbb{N}\}$
- **Executability preconditions:** $\mathcal{P}(\epsilon)$ for $\epsilon \in Act_{\mathfrak{m}}$
- **Successful occurrence of an event:** $\langle\!\langle +^t_{\mathfrak{m}}\alpha \rangle\!\rangle \varphi \quad \stackrel{\text{def}}{=} \quad \mathcal{P}(+^t_{\mathfrak{m}}\alpha) \wedge [+^t_{\mathfrak{m}}\alpha]\varphi$
- **Action selection problem (ASP):** $\langle \Sigma, Op, \alpha_G \rangle$ where:
  - $\Sigma \subset \mathcal{L}_0(Atm)$ ($\mathfrak{m}$'s belief base),
  - $Op \subset Act_{\mathfrak{m}}$ ($\mathfrak{m}$'s action repertoire),
  - $\alpha_G \in \mathcal{L}_0(Atm)$ ($\mathfrak{m}$'s goal).
- **Solution to ASP:** $\epsilon \in Op$ such that $\Sigma \models_{\mathbf{M}} \langle\!\langle \epsilon \rangle\!\rangle \Box_{\mathfrak{m}} \alpha_G$.

Update, then revision

- Belief update : $\mathfrak{m}$'s explicit beliefs evolve from a time t to $t+1$.
- Belief revision : $\mathfrak{m}$ learns a new fact and adds it to its belief base.

Beliefs update

- $Upd(\Sigma_c, \Sigma_h, \Sigma_f) = (\Sigma'_c, \Sigma'_h, \Sigma'_f)$ if and only if:
  - $\Sigma'_c = \Sigma_c$,
  - $\Sigma'_h = incrt(\Sigma_h)$,
  - $\Sigma'_f = incrt(\Sigma_f) \cup \{now^{\geq 0}\}$.
- where for each finite $X \subseteq \mathcal{L}_0$, $incrt(X) = \{incrt(\alpha) : \alpha \in X\}$ such that:
  - $incrt(p^t) = p^{t+1}$
  - $incrt(\triangle^t_{\mathfrak{h}}\, \alpha) = \triangle^{t+1}_{\mathfrak{h}} incrt(\alpha)$
  - $incrt(\triangle_{\mathfrak{m}}\, \alpha) = \triangle_{\mathfrak{m}} incrt(\alpha)$
  - $incrt(now^{\geq t}) = now^{\geq t+1}$
  - $incrt(\neg\alpha) = \neg incrt(\alpha)$
  - $incrt(\alpha_1 \wedge \alpha_2) = incrt(\alpha_1) \wedge incrt(\alpha_2)$.

Belief revision (reminder)

- $Rev(\Sigma_c, \Sigma_h, \Sigma_f, \Sigma_{input}) =$
  - $(\Sigma_c, \Sigma_h, \Sigma_f)$ if $\Sigma_c \cup \Sigma_{input}$ is not propositionally consistent ;
  - $(\Sigma'_c, \Sigma'_h, \Sigma'_f)$ otherwise, with $\Sigma'_c = \Sigma_c$, $\Sigma'_f = \bigcap_{X \in MCS(\Sigma_c, \Sigma_f, \Sigma_{input})} X$, $\Sigma'_h = \bigcap_{X \in MCS(\Sigma_c \cup \Sigma'_f, \Sigma_h, \emptyset)} X$.
- So:
  - Σ_c is not modified,
  - Σ_{input} is added to Σ_f (iff Σ_{input} and Σ_c are consistent),
  - belief revision satisfies minimal change for factual information.

Part 4 - Application to the Yōkai game

Logical modeling of Yōkai

Useful sets

Let be the following sets:

$$TIME = \{0, 1, .., end\} \text{ where } end = 56$$
$$TIME^* = TIME \setminus \{0\}$$
$$GRID = \{1, \ldots, 32\} \times \{1, \ldots, 32\},$$
$$IPOS = \big\{(l, c) \in GRID : l, c \in \{15, ..., 18\}\big\},$$
$$COLORS = \{\mathrm{r}, \mathrm{g}, \mathrm{b}, \mathrm{y}\},$$
$$HINTS = 2^{COLORS} \setminus \big\{\{\}, \{\mathrm{r}, \mathrm{g}, \mathrm{b}, \mathrm{y}\}\big\},$$
$$CARDS = \{1, \ldots, 16\}$$
$$CARDS^n = \{X \in 2^{CARDS} : |X| = n\} \text{ with } n \in \mathbb{N}$$

Atomic propositions

- $col^t_{x,c}$: card x is of color c at time t,
- $pos^t_{x,p}$: card x is at position p at time t,
- $active^t_h$: hint h is enabled at time t,
- $mark^t_{x,h}$: card x is marked with hint h at time t,
- $legMov^t_{x,p}$: to move card x to position p at time t is legal.

Initial factual belief base

$$\Sigma_f^W = \{now^{\geq 0}\} \cup \bigcup_{x \in CARDS} \{pos^0_{x,\sigma(x)}\} \cup \bigcup_{\substack{x \in CARDS \\ c \in COLORS}} \{\neg \triangle_{\mathfrak{m}}\, col^0_{x,c}\} \cup$$

$$\bigcup_{h \in HINTS} \{\neg active^0_h\} \cup \bigcup_{\substack{x \in CARDS \\ h \in HINTS}} \{\neg mark^0_{x,h}\}$$

$$\Sigma_f^{\mathfrak{h}} = \bigcup_{\alpha \in \Sigma_f^W} \{\triangle^0_{\mathfrak{h}}\, \alpha\} \cup \bigcup_{\substack{x \in CARDS \\ c \in COLORS}} \{\neg \triangle^0_{\mathfrak{h}}\, col^0_{x,c}\}$$

$$\Sigma_f = \Sigma_f^W \cup \Sigma_f^{\mathfrak{h}}$$

Initial hypothetical belief base

$$\Sigma_h = \bigcup_{\substack{x \in CARDS \\ h \in HINTS \\ c \in COLORS \setminus h}} \{mark^0_{x,h} \rightarrow \neg col^0_{x,c} \ , \ \triangle^0_{\mathfrak{h}} \, mark^0_{x,h} \rightarrow \triangle^0_{\mathfrak{h}} \neg col^0_{x,c}\}$$

Example

1) $\Sigma_h \supseteq \{mark^0_{1,\{\mathrm{g,r}\}} \rightarrow \neg col^0_{1,\mathrm{b}}, mark^0_{1,\{\mathrm{g,r}\}} \rightarrow \neg col^0_{1,\mathrm{y}}\}$.

2) $mark^0_{1,\{\mathrm{g,r}\}} \in \Sigma_f$.

So, we can deduce that $\Box_{\mathfrak{m}} \left((col^0_{1,\mathrm{g}} \vee col^0_{1,\mathrm{r}}) \wedge \neg col^0_{1,\mathrm{b}} \wedge \neg col^0_{1,\mathrm{y}} \right)$ holds.

Core belief base (integrity constraints)

- Rules of the game:

$$\Sigma_c^{ic} = \bigcup_{t \in TIME} \left\{ \bigwedge_{x \in CARDS} \bigvee_{p \in GRID} pos^t_{x,p}\,, \right. \qquad \text{(ICP1)}$$

$$\bigwedge_{\substack{x \in CARDS \\ p,p' \in GRID : p \neq p'}} \neg(pos^t_{x,p} \wedge pos^t_{x,p'})\,, \qquad \text{(ICP2)}$$

$$\bigwedge_{\substack{p \in GRID \\ x,x' \in CARDS : x \neq x'}} \neg(pos^t_{x,p} \wedge pos^t_{x',p})\,, \qquad \text{(ICP3)}$$

$$\left. \ldots \right\}$$

Core belief base (frame axioms)

- Description of what does not change:

$$\text{posFA}_X^t \stackrel{\text{def}}{=} \bigwedge_{\substack{x \in CARDS \setminus X \\ p \in GRID}} (pos_{x,p}^t \leftrightarrow pos_{x,p}^{t-1})$$

$$\text{colFA}_X^t \stackrel{\text{def}}{=} \bigwedge_{\substack{x \in CARDS \setminus X \\ c \in COLORS}} \Big((col_{x,c}^t \leftrightarrow col_{x,c}^{t-1}) \wedge (\neg \triangle_{\mathfrak{m}}\, col_{x,c}^t \leftrightarrow \neg \triangle_{\mathfrak{m}}\, col_{x,c}^{t-1}) \Big)$$

$$\text{hintFA}_H^t \stackrel{\text{def}}{=} \bigwedge_{h \in HINTS \setminus H} (active_h^t \leftrightarrow active_h^{t-1}) \wedge \bigwedge_{\substack{x \in CARDS \\ h \in HINTS \setminus H}} (mark_{x,h}^t \leftrightarrow mark_{x,h}^{t-1})$$

Core belief base (successor state axioms)

- Computation of a future state:

$$\Sigma_c^{ssa} = \bigcup_{t \in TIME^*} \Bigg\{ \bigwedge_{\substack{x \in CARDS \\ p \in GRID}} (pos_{x,p}^t \wedge \neg pos_{x,p}^{t-1} \rightarrow \mathsf{posFA}_{\{x\}}^t \wedge \mathsf{colFA}_{\emptyset}^t \wedge \mathsf{hintFA}_{\emptyset}^t)\,, \qquad \text{(SSA4)}$$

$$\bigwedge_{\substack{x \in CARDS \\ c \in COLORS}} (col_{x,c}^t \wedge \neg \triangle_{\mathfrak{m}}\, col_{x,c}^{t-1} \rightarrow \mathsf{posFA}_{\emptyset}^t \wedge \mathsf{colFA}_{\{x\}}^t \wedge \mathsf{hintFA}_{\emptyset}^t)\,, \qquad \text{(SSA5)}$$

$$\bigwedge_{h \in HINTS} (active_h^t \wedge \neg active_h^{t-1} \rightarrow \mathsf{posFA}_{\emptyset}^t \wedge \mathsf{colFA}_{\emptyset}^t \wedge \mathsf{hintFA}_{\{h\}}^t)\,, \qquad \text{(SSA6)}$$

$$\bigwedge_{\substack{x \in CARDS \\ h \in HINTS}} (mark_{x,h}^t \wedge \neg mark_{x,h}^{t-1} \rightarrow \mathsf{posFA}_{\emptyset}^t \wedge \mathsf{colFA}_{\emptyset}^t \wedge \mathsf{hintFA}_{\{h\}}^t) \Bigg\} \qquad \text{(SSA7)}$$

Core belief base

- Finally:

$$\Sigma_c = \Sigma_c^{ic} \cup \Sigma_c^{ssa}$$

Action repertoire

- $ACT \subseteq EVT$ of agent $\mathfrak{m}$ is defined as follows:

$$+_{\mathfrak{m}}^{\mathsf{col}^t_{x,c}} \stackrel{\text{def}}{=} +_{\mathfrak{m}}^{t}\Big(col^t_{x,c} \wedge \triangle_{\mathfrak{m}}\, col^t_{x,c} \wedge (\triangle^t_{\mathfrak{h}} \bigvee_{c' \in COLORS} \triangle_{\mathfrak{m}}\, col^t_{x,c'}) \wedge \triangle^t_{\mathfrak{h}}\, now^{\geq t}\Big)$$

$$+_{\mathfrak{m}}^{\mathsf{pos}^t_{x,p}} \stackrel{\text{def}}{=} +_{\mathfrak{m}}^{t}\Big(pos^t_{x,p} \wedge \triangle_{\mathfrak{m}}\, pos^t_{x,p} \wedge \triangle^t_{\mathfrak{h}}\, pos^t_{x,p} \wedge \triangle^t_{\mathfrak{h}}\, now^{\geq t}\Big)$$

$$+_{\mathfrak{m}}^{\mathsf{actHint}^t_{h}} \stackrel{\text{def}}{=} +_{\mathfrak{m}}^{t}\Big(active^t_{h} \wedge \triangle_{\mathfrak{m}}\, active^t_{h} \wedge \triangle^t_{\mathfrak{h}}\, active^t_{h} \wedge \triangle^t_{\mathfrak{h}}\, now^{\geq t}\Big)$$

$$+_{\mathfrak{m}}^{\mathsf{markHint}^t_{x,h}} \stackrel{\text{def}}{=} +_{\mathfrak{m}}^{t}\Big(mark^t_{x,h} \wedge \triangle_{\mathfrak{m}}\, mark^t_{x,h} \wedge \triangle^t_{\mathfrak{h}}\, mark^t_{x,h} \wedge \triangle^t_{\mathfrak{h}}\, now^{\geq t}\Big)$$

Actions preconditions

$$\mathcal{P}(+_{\mathfrak{m}}^{\mathsf{col}_{x,c}^{t+1}}) \stackrel{\text{def}}{=} now^{=t} \wedge \Box_{\mathfrak{m}} \bigwedge_{h \in HINTS} \neg mark_{x,h}^{t}$$

$$\mathcal{P}(+_{\mathfrak{m}}^{\mathsf{pos}_{x,p}^{t+1}}) \stackrel{\text{def}}{=} now^{=t} \wedge \Box_{\mathfrak{m}} \bigwedge_{h \in HINTS} \neg mark_{x,h}^{t} \wedge \Box_{\mathfrak{m}}\, legMov_{x,p}^{t}$$

$$\mathcal{P}(+_{\mathfrak{m}}^{\mathsf{actHint}_{h}^{t+1}}) \stackrel{\text{def}}{=} now^{=t}$$

$$\mathcal{P}(+_{\mathfrak{m}}^{\mathsf{markHint}_{x,h}^{t+1}}) \stackrel{\text{def}}{=} now^{=t} \wedge \Box_{\mathfrak{m}}\, active_{h}^{t} \wedge \bigvee_{c \in h} \Box_{\mathfrak{m}}\, col_{x,c}^{t} \wedge \Box_{\mathfrak{m}} \bigwedge_{\substack{h' \in HINTS \\ h' \neq h}} \neg mark_{x,h'}^{t}$$

Goals modeling (observe a card around X)

- goal = motivational attitude for action,
- $X \subseteq CARDS : 1 \leq |X| \leq 3$, a set of cards of the same color,

$$\alpha^t_{obsArroundCards(X)} \stackrel{\text{def}}{=} \bigvee_{\substack{x' \in CARDS \\ x' \notin X}} \Big(\Big(\bigwedge_{c \in COLORS} \neg \triangle_{\mathfrak{m}}\, col^t_{x',c} \Big) \wedge \mathsf{nbgCards}^t_{X \cup \{x'\}} \wedge \Big(\bigvee_{c \in COLORS} \triangle_{\mathfrak{m}}\, col^{t+1}_{x',c} \Big) \Big)$$

$$\mathcal{P}_g(\alpha^t_{obsArroundCards(X)}) \stackrel{\text{def}}{=} \mathsf{nbgCards}^t_X$$

Goals modeling (observe a card whose color is unknown)

Let $x \in CARDS$:

$$\alpha^t_{obsUnknownColorCard(x)} \stackrel{\text{def}}{=} \bigvee_{c \in COLORS} \triangle_{\mathfrak{m}}\, col^{t+1}_{x,c}$$

$$\mathcal{P}_g(\alpha^t_{obsUnknownColorCard(x)}) \stackrel{\text{def}}{=} \bigwedge_{c \in COLORS} \neg \triangle_{\mathfrak{m}}\, col^{t}_{x,c}$$

Goals modeling (observe a card randomly)

- Default goal for moving ($x \in CARDS$):

$$\alpha^t_{obsRandomly(x)} \stackrel{\text{def}}{=} \bigvee_{c \in COLORS} \triangle_{\mathfrak{m}}\, col^{t+1}_{x,c}$$

$$\mathcal{P}_g(\alpha^t_{obsRandomly(x)}) \stackrel{\text{def}}{=} \top$$

How to manage the different goal?

- With meta algorithms generating an ordered list of goals
- getSubsetsGtLeq$(X, \tau_m, \tau_M) = \{X' \in 2^X : \tau_m < |X'| \leq \tau_M\}$

```
1: function tryToObserve(S, α_g^t)
2:     success ← false
3:     while S ≠ ∅  &&  !success do
4:         x ← pop(S)
5:         if P_g(α_{g(x)}^t) then
6:             success ← checkPlan(α_{g(x)}^t)
7:         end if
8:     end while
9:     return success
10: end function
```

```
1: ▷ Parameters computation for α^t_{obsArroundCards(X)}
2: S ← { }
3: for all c ∈ COLORS do
4:     X_c ← {x ∈ CARDS : △_m col^t_{x,c} ∈ Σ_f}
5:     S ← S ∪ getSubsetsGtLeq(X_c, 0, |X_c|)
6: end for
7: S' ← sortByDecreasingSize(S)
8:                          ▷ ∀X_i, X_{i+1} ∈ S', |X_i| ≥ |X_{i+1}|
```

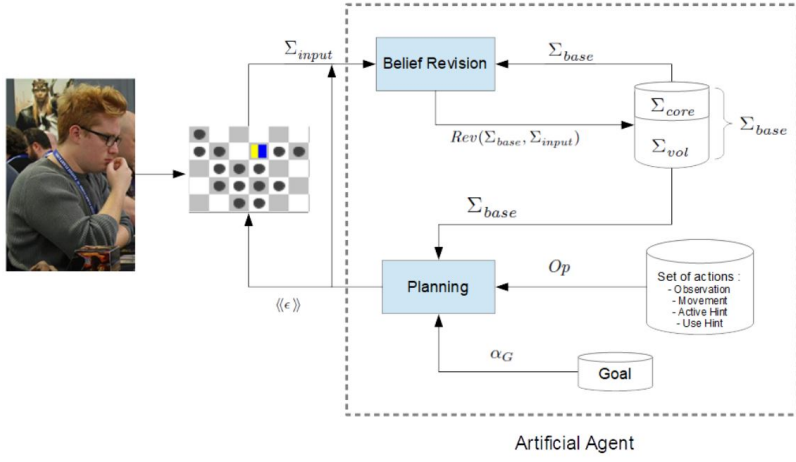
How to manage the different goal?

1: **if** !tryToObserve($S', \alpha^t_{obsArroundCards}$) **then**
2: $X \leftarrow \{x \in CARDS : \exists c \in COLORS, \triangle_{\mathtt{m}}\ col^t_{x,c} \in \Sigma_f\}$
3: $X' \leftarrow CARDS \setminus X$ ▷ The set of cards whose color is unknown
4: **if** !tryToObserve($X', \alpha^t_{obsUnknownColorCard}$) **then**
5: tryToObserve($CARDS, \alpha^t_{obsRandomly}$)
6: **end if**
7: **end if**

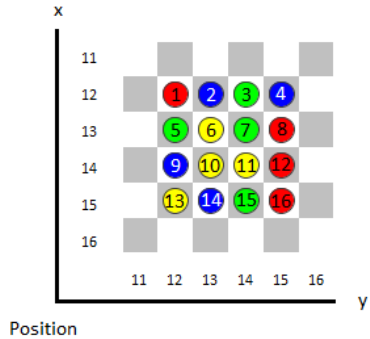
Part 4 - Application to the Yōkai game

Implementation

System Architecture

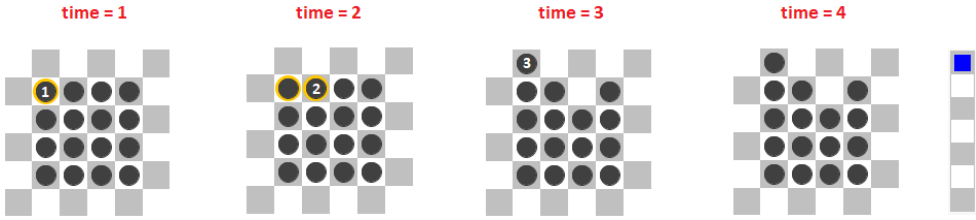


Cards distribution



- Source code:
  `https://github.com/iritlab/yokai`

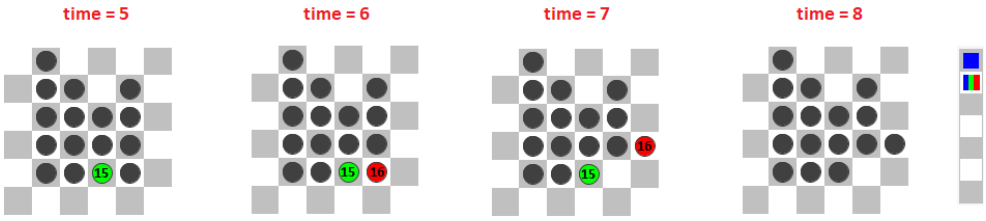
Agent $\mathfrak{m}$'s first round




```
1: col_1_R_t4 and tm_col_1_R_t4 and (th_tm_col_1_G_t4 or th_tm_col_1_Y_t4 or
                                     th_tm_col_1_B_t4 or th_tm_col_1_R_t4)
2: col_2_B_t4 and tm_col_2_B_t4 and (th_tm_col_2_G_t4 or th_tm_col_2_Y_t4 or
                                     th_tm_col_2_B_t4 or th_tm_col_2_R_t4)
3: pos_3_p_11_12_t4 and tm_pos_3_p_11_12_t4 and th_pos_3_p_11_12_t4
4: act_1_t4 and tm_act_1_t4 and th_act_1_t4
```

Σ_m at time point 4

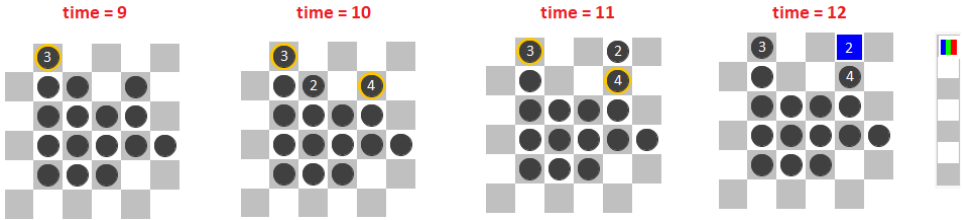
Agent $\mathfrak{h}$'s first round



5: (th_col_15_R_t8 or th_col_15_G_t8 or th_col_15_Y_t8 or th_col_15_B_t8)
6: (th_col_16_R_t8 or th_col_16_G_t8 or th_col_16_Y_t8 or th_col_16_B_t8)
7: pos_16_p_14_16_t8 and tm_pos_16_p_14_16_t8 and th_pos_16_p_14_16_t8
8: act_6_t8 and tm_act_6_t8 and th_act_6_t8

Σ_m at time point 8

Grouping cards



.
2: col_2_B_t10 and tm_col_2_B_t10 and (th_tm_col_2_G_t10 or th_tm_col_2_Y_t10 or th_tm_col_2_B_t10 or th_tm_col_2_R_t10)
.
9: col_3_G_t10 and tm_col_3_G_t10 and (th_tm_col_3_G_t10 or th_tm_col_3_Y_t10 or th_tm_col_3_B_t10 or th_tm_col_3_R_t10)
10: col_4_B_t10 and tm_col_4_B_t10 and (th_tm_col_4_G_t10 or th_tm_col_4_Y_t10 or th_tm_col_4_B_t10 or th_tm_col_4_R_t10)

$\Sigma_{\dots}$ at time point 10

Action selection

```
(not ([m](
not tm_col_5_G_t10 and
not tm_col_6_Y_t10 and
.
col_1_R_t10 and tm_col_1_R_t10 and (th_tm_col_1_G_t10 or th_tm_col_1_Y_t10 or th_tm_col_1_B_t10 or th_tm_col_1_R_t10) and
col_2_B_t10 and tm_col_2_B_t10 and (th_tm_col_2_G_t10 or th_tm_col_2_Y_t10 or th_tm_col_2_B_t10 or th_tm_col_2_R_t10) and
act_1_t10 and tm_act_1_t10 and th_act_1_t10 and
.
;;SSA11
((pos_2_p_11_15_t11 and not pos_2_p_11_15_t10) =>
((pos_1_p_12_12_t11 <=> pos_1_p_12_12_t10) and ((pos_3_p_11_12_t11 <=> pos_3_p_11_12_t10) and
((pos_4_p_12_15_t11 <=> pos_4_p_12_15_t10) and ((pos_5_p_13_12_t11 <=> pos_5_p_13_12_t10) and
((pos_6_p_13_13_t11 <=> pos_6_p_13_13_t10)))))))
.
)=> ([m] now_t10 and [m] (not mark_2_1_t10 and not mark_2_2_t10 and
          not mark_2_3_t10 and not mark_2_4_t10 and not mark_2_5_t10 and
          not mark_2_6_t10 and not mark_2_7_t10) and
    plus(pos_2_p_11_15_t11 and tm_pos_2_p_11_15_t11 and th_pos_2_p_11_15_t11,
        [m]((pos_2_p_11_15_t11 and pos_4_p_12_15_t11) or (pos_2_p_12_14_t11 and pos_4_p_12_15_t11) or
        (pos_2_p_12_16_t11 and pos_4_p_12_15_t11) or (pos_4_p_11_13_t11 and pos_2_p_12_13_t11) or
        (pos_4_p_12_14_t11 and pos_2_p_12_13_t11))))))
```

Planning formula for grouping blue cards 2 and 4

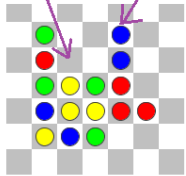
Belief revision

Integrity constraint ICP2

```
.
.
(not (pos_2_p_12_13_t11 and pos_2_p_12_11_t11) and
(not (pos_2_p_12_13_t11 and pos_2_p_10_12_t11) and
(not (pos_2_p_12_13_t11 and pos_2_p_11_11_t11) and
(not (pos_2_p_12_13_t11 and pos_2_p_11_13_t11) and
.
.
(not (pos_2_p_12_13_t11 and pos_2_p_11_15_t11) and
(not (pos_2_p_12_13_t11 and pos_2_p_12_14_t11) and
(not (pos_2_p_12_13_t11 and pos_2_p_12_16_t11) and
(not (pos_2_p_12_13_t11 and pos_2_p_13_11_t11) and
```

Initial position

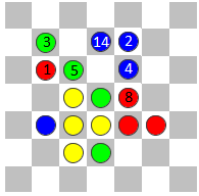
Final position



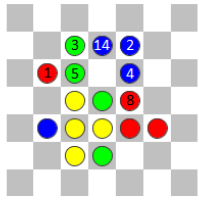
Hierarchy of goals

Sets of cards known by the machine :

1 B|2 4 14
2 R|1 8
3 G|3 5
4 B|4 14
5 B|2 4
6 B|2 14
7 Y|6

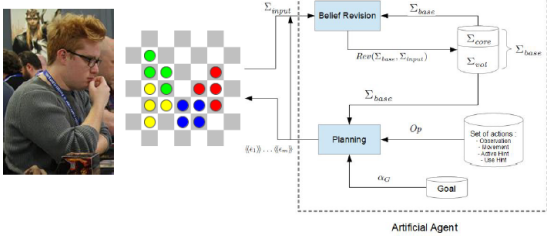


Human round at time t

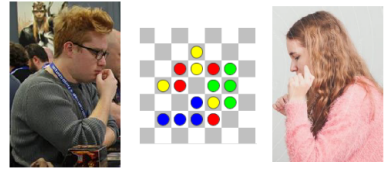


Machine round at time t+3

Demo

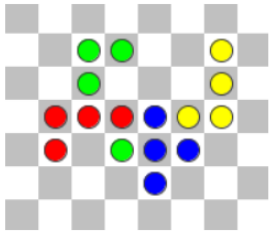

Belief Revision
Planning
Goal
Set of actions :
- Castes/skin
- Movement
- Active Hint
- Use Hint
Artificial Agent

ℌ - 𝔪

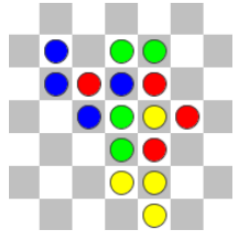


ℌ - ℌ

Experiments



Case 01 - Failed game



Case 02 - Failed game

Figure: Two different scenarios when players lost the game

Experiments(cont.)

Experiment set	Game	ɦ-ɱ		ɦ-ɦ	
		# 4-y-groups	Score	# 4-y-groups	Score
1	1	1	-	1	-
	2	1	-	2	-
	3	4	8	2	-
	4	1	-	2	-
	5	3	-	1	-
	6	4	8	4	9
	7	4	7	3	-
	8	2	-	2	-
	9	1	-	1	-
	10	1	-	0	-
Total:		22		18	

Table: Total # 4-y-groups in experiment set 1

Experiments(cont.)

Config. type	Experiment set									
	1	2	3	4	5	6	7	8	9	10
ɦ-ɱ	22	28	33	32	28	30	19	26	23	26
ɦ-ɦ	18	23	21	23	24	25	17	19	21	25

Table: Total # 4-y-groups by experiment set

Experiments(cont.)

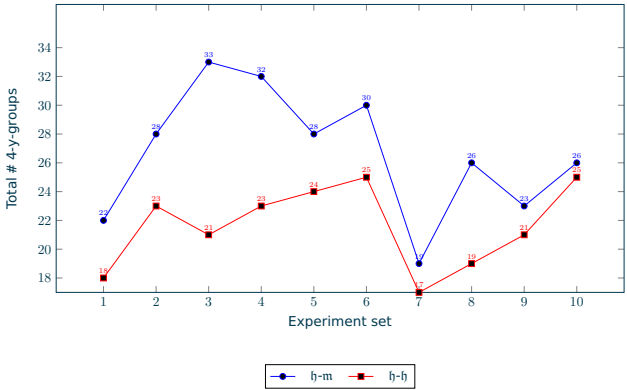


Figure: Total # 4-y-groups by experiment set

Bibliography

Griffon, J. (2019).
Règles du jeu de Yōkai.
https://www.play-in.com/pdf/rules_games/yokai_regles_fr.pdf.