Generation of Strong Cyclic Plans with Incomplete Information and Sensing

joint work with: Luca Iocchi, Riccardo Rosati

⇒ Incomplete information about the world

⇒ Partial observability of the world
Incomplete information

The exact state of the world is not completely known by the agent

⇒ **incomplete initial state**

Moreover, the agent acts in an unpredictable environment, where other (unknown) agents may unpredictably change the state of the world

⇒ **presence of exogenous actions/effects** (i.e., the state of the world is non-inertial)
Partial observability

• The agent is able to execute **sensing actions** to acquire knowledge about the environment

• However, only some properties of the world can be sensed by the agent

• Moreover, the executability of such sensing actions is in general context-dependent

⇒ partial observability
Strong Cyclic Conditional planning

Conditional planning = *nondeterministic control problem over belief space.*

Strong Cyclic Conditional (or contingent) planning: plan with

- ordinary actions
- sensing actions
- conditional (if–then–else) statement (sensing)
- loop (while) statement (sensing)
Our proposal

1. definition of \( \mathcal{KL} \), a language for representing planning domains with incomplete information and partial observability

2. semantics of \( \mathcal{KL} \) expressed through sets of belief states

3. definition of the notion of strong cyclic conditional plan in the \( \mathcal{KL} \) framework

4. algorithm for computing strong cyclic conditional plans in \( \mathcal{KL} \)

5. implementation and experimentation of the algorithm
The language $\mathcal{KL} – Syntax$

- initial state = formula $K\phi_{\text{init}}$
- goal = formula $K\phi_{\text{goal}}$
- precondition for action $a = \text{pre}_a : K\alpha$
- effect of ordinary action $a = \text{post}_a : \bigwedge_i (K\alpha_i \rightarrow K\beta_i)$
- effect of sensing action $a = \text{sense}_a : P$
- forgetting effect of action $a = \text{post}_a : \neg KP$
- static axiom = formula $\alpha$
The language $\mathcal{KL}$ – Semantics

- **belief state** = set of propositional interpretations

- A belief state $b$ is denoted by an epistemic formula $K\phi_b$ such that $b$ is the set of interpretations satisfying $\phi_b$

- Initial belief state = $K\phi_{\text{init}}$

- The **transition function** $f_a(b) = b_a^o$ is defined through the combination of: deterministic effects, sensing effects, forgetting effects and static axioms
\textbf{\(\mathcal{KL}\) specification: Cleaning example}

Cleaning Robot searching and cleaning objects out of a set of rooms.

<table>
<thead>
<tr>
<th>Action  (a)</th>
<th>(pre_\alpha)</th>
<th>(post_\alpha)</th>
<th>(sense_\alpha)</th>
</tr>
</thead>
<tbody>
<tr>
<td>enter_Rj</td>
<td>(Kin(C))</td>
<td>(Kin(R_j))</td>
<td></td>
</tr>
<tr>
<td>exit_Rj</td>
<td>(Kin(R_j))</td>
<td>(Kin(C'))</td>
<td></td>
</tr>
<tr>
<td>takeout_Rj</td>
<td>(Kin(R_j) \land K\text{obj_in}(R_j))</td>
<td>(Kin(C') \land \neg K\text{obj_in}(R_j))</td>
<td></td>
</tr>
<tr>
<td>scan_Rj</td>
<td>(Kin(R_j))</td>
<td></td>
<td>(obj_in(R_j))</td>
</tr>
</tbody>
</table>

Without knowing how many objects are in every room, a conditional plan (without loops) solution does not exist.
Strong cyclic planning in $\mathcal{KL}$

- searching in the belief states space
- using heuristics to find linear paths (i.e. partial plans considering only one effect of sensing)
- merging partial plans to (possibly cyclic) conditional plans.

Epistemic reasoning framework that provides the semantics for $\mathcal{KL}$ [Locchi, Nardi, Rosati (KR-2000)] allows for **verifying equivalent belief states** and guarantees **soundness** and **completeness**.
Cyclic conditional plan for the Cleaning domain
Experimental results for conditional planning

Experimental results show that explicit representation of the agent knowledge ($\mathcal{K}$-Planner and PKS) significantly improve performance of reasoning, with respect to representations based on possible worlds.

<table>
<thead>
<tr>
<th>Rooms</th>
<th>First Linear Path</th>
<th>Complete Linear Path</th>
<th>PKS [PeBa02]</th>
<th>MBP [BeCi01]</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>60</td>
<td>3</td>
<td>-</td>
<td>10</td>
</tr>
<tr>
<td>20</td>
<td>240</td>
<td>8</td>
<td>-</td>
<td>300</td>
</tr>
<tr>
<td>30</td>
<td>1040</td>
<td>35</td>
<td>-</td>
<td>11500</td>
</tr>
<tr>
<td>40</td>
<td>3230</td>
<td>60</td>
<td>-</td>
<td>?</td>
</tr>
<tr>
<td>50</td>
<td>7900</td>
<td>85</td>
<td>-</td>
<td>?</td>
</tr>
</tbody>
</table>

Cleaning domain

<table>
<thead>
<tr>
<th>Illnesses</th>
<th>$\mathcal{K}$-Planner</th>
<th>PKS [PeBa02]</th>
<th>MBP [BeCi01]</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1</td>
<td>-</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>-</td>
<td>300</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>-</td>
<td>11500</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>-</td>
<td>?</td>
</tr>
<tr>
<td>20</td>
<td>65</td>
<td>80</td>
<td>?</td>
</tr>
<tr>
<td>50</td>
<td>620</td>
<td>1610</td>
<td>?</td>
</tr>
<tr>
<td>100</td>
<td>4620</td>
<td>20390</td>
<td>?</td>
</tr>
</tbody>
</table>

Medical domain
Conclusions

• definition of a language $\mathcal{KL}$ for representing planning domains with incomplete information and partial observability

• algorithm for computing strong cyclic conditional plans in $\mathcal{KL}$

• cyclic plan generation in domains where a conditional plan solution does not exist

• efficient implementation using explicit knowledge representation and heuristic search