AI Planning
Lecture 3

Rostislav Horčík

Czech Technical University in Prague
Faculty of Electrical Engineering
xhorcik@fel.cvut.cz

Propositional representation

Grounding

Action schema $a[\vec{v}] = \langle \text{pre}_a[\vec{v}], \text{add}_a[\vec{v}], \text{del}_a[\vec{v}] \rangle$

To find applicable actions in state s, we must find all tuple of objects $\vec{b}$ s.t. $\text{pre}_a[\vec{b}] \subseteq s$.

This is NP-hard problem (evaluation of conjuctive query).

Thus most planners ground the first-order representation to the propositional level.

Grounding: for each action schema $a[\vec{v}]$ precompute all possible grounded actions $a[\vec{b}]$ for a tuple of objects $\vec{b}$.

Example

Objects – locations l_1, l_2, l_3, truck t_1, package p_1

Predicates – unary L, T, P, binary At, In

Action schema – pick$[t, p, l]$

- $\text{pre}_{\text{pick}}[t, p, l] = \{T(t), P(p), L(l), At(t, l), At(p, l)\}$
- $\text{add}_{\text{pick}}[t, p, l] = \{In(p, t)\}$
- $\text{del}_{\text{pick}}[t, p, l] = \{At(p, l)\}$

Grounding: $pick[t_1, p_1, l_1], pick[t_1, p_1, l_2], pick[t_1, p_1, l_3]$

STRIPS

Definition

A STRIPS planning task is a tuple $\Pi = \langle F, A, s_0, g \rangle$ where

- F is a set of facts,
- the initial state $s_0 \subseteq F$,
- the goal $g \subseteq F$,
- and A is a set of actions.

Each action $a \in A$ is a triple $a = \langle \mathrm{pre}_a, \mathrm{add}_a, \mathrm{del}_a \rangle$ of three sets of facts.

Π induces LTS $\Theta_\Pi = \langle S, A, T, G \rangle$ where

- $S = 2^F = \{s \mid s \subseteq F\}$,
- $G = \{s \subseteq F \mid g \subseteq s\}$,
- $s \xrightarrow{a} t$ iff $\mathrm{pre}_a \subseteq s$ and $t = (s \setminus \mathrm{del}_a) \cup \mathrm{add}_a$.

Example

Let $\Pi = \langle F, A, s_0, g \rangle$ be a STRIPS planning task where

- $F = \{b, c, d\}$,
- $s_0 = \{b\}$,
- $g = \{c, d\}$,
- actions A consists of actions

$$a_1 = \langle \{b\}, \{d\}, \emptyset \rangle$$
$$a_2 = \langle \{b, d\}, \{c\}, \{b\} \rangle$$
$$a_3 = \langle \{b\}, \{c\}, \emptyset \rangle$$

Complexity

Theorem (Bylander 94)

The problem deciding whether there is a plan for a given STRIPS planning task is PSPACE-complete.

Mutex group

Definition

A set of facts M is said to be a mutex group if $|M \cap s| \leq 1$ for all reachable states s.

Example

$M = \{At(t_1, l_1), At(t_1, l_2), At(t_1, l_3)\}$ is a mutex group.

Truck t_1 can be at most in a single location.

To represent a state, it suffices to store only which atom from M holds (or if none of them).

FDR states

Suppose $M_1, \ldots, M_k$ is a family of pairwise disjoint mutex groups such that $\bigcup_{i=1}^{k} M_i = F$.

We can represent any reachable state s as a function $\nu_s \colon \{1, \ldots, n\} \to F \cup \{\bot\}$ such that $\nu_s(i) \in M_i \cup \{\bot\}$.

Definition

Let V be a set of variables each $v \in V$ with its domain $\mathrm{dom}(v)$.

- A partial function $s \colon V \to \bigcup_{v \in V} \mathrm{dom}(v)$ is called a partial state if $s(v) \in \mathrm{dom}(v)$ for each $v \in V$.
- If s is total, we call s a state.
- Let s, t be two partial states. We say that t extends s if $s \subseteq t$.

Definition

FDR planning task (aka SAS^+) is $\Pi = \langle V, A, s_0, g \rangle$ where

- V is a set of variables,
- the initial state s_0,
- the goal g is a partial state,
- A is a set of actions.

Each action $a \in A$ is a pair $a = \langle \text{pre}_a, \text{eff}_a \rangle$ where $\text{pre}_a, \text{eff}_a$ are partial states.

LTS induced by FDR

FDR task Π induces an LTS $\Theta_\Pi = \langle S, A, T, G \rangle$ where

- S is the set of all states,
- $G = \{s \in S \mid g \subseteq s\}$,
- A is the set of actions from Π,
- For $s, t \in S$ and $a \in A$, there is a transition $s \xrightarrow{a} t$ iff $\mathrm{pre}_a \subseteq s$ and

$$t(v) = \begin{cases} \mathrm{eff}_a(v) & \text{if } \mathrm{eff}_a(v) \text{ is defined,} \\ s(v) & \text{otherwise.} \end{cases}$$

Example

FDR task $\Pi = \langle V, A, s_0, g\rangle$ where

- $V = \{v_1, v_2\}$ with $\text{dom}(v_1) = \{X, Y, Z\}$ and $\text{dom}(v_2) = \{0, 1\}$,
- $s_0 = \{\langle v_1, X\rangle, \langle v_2, 1\rangle\}$,
- $g = \{\langle v_1, Z\rangle\}$,
- $A = \{a_1, \ldots, a_5\}$:

1. $\text{pre}_{a_1} = \{\langle v_1, X\rangle, \langle v_2, 1\rangle\}$ and $\text{eff}_{a_1} = \{\langle v_1, Y\rangle, \langle v_2, 0\rangle\}$.
2. $\text{pre}_{a_2} = \{\langle v_1, Y\rangle, \langle v_2, 1\rangle\}$ and $\text{eff}_{a_2} = \{\langle v_1, X\rangle, \langle v_2, 0\rangle\}$.
3. $\text{pre}_{a_3} = \{\langle v_1, Y\rangle, \langle v_2, 1\rangle\}$ and $\text{eff}_{a_3} = \{\langle v_1, Z\rangle, \langle v_2, 0\rangle\}$.
4. $\text{pre}_{a_4} = \{\langle v_1, Z\rangle, \langle v_2, 1\rangle\}$ and $\text{eff}_{a_4} = \{\langle v_1, Y\rangle, \langle v_2, 0\rangle\}$.
5. $\text{pre}_{a_5} = \{\langle v_2, 0\rangle\}$ and $\text{eff}_{a_5} = \{\langle v_2, 1\rangle\}$.

STRIPS2FDR

Let $\Pi = \langle F, A, s_0, g \rangle$ be a STRIPS task.

Define $\Pi^{\mathrm{FDR}} = \langle F, A^{\mathrm{FDR}}, s_0^{\mathrm{FDR}}, g^{\mathrm{FDR}} \rangle$ where

- $\mathrm{dom}(p) = \{0, 1\}$ for $p \in F$,
- $s_0^{\mathrm{FDR}} = \{\langle p, 1 \rangle \mid p \in s_0\} \cup \{\langle p, 0 \rangle \mid p \notin s_0\}$,
- $g^{\mathrm{FDR}} = \{\langle p, 1 \rangle \mid p \in g\}$,
- $A^{\mathrm{FDR}} = \{a^{\mathrm{FDR}} \mid a \in A\}$ where
  - $\mathrm{pre}_{a^{\mathrm{FDR}}} = \{\langle p, 1 \rangle \mid p \in \mathrm{pre}_a\}$ and
  - $\mathrm{eff}_{a^{\mathrm{FDR}}} = \{\langle p, 1 \rangle \mid p \in \mathrm{add}_a\} \cup \{\langle p, 0 \rangle \mid p \in \mathrm{del}_a\}$,
- $\mathrm{cost}(a^{\mathrm{FDR}}) = \mathrm{cost}(a)$ for $a \in A$.

FDR2STRIPS

Let $\Pi = \langle V, A, s_0, g \rangle$ be an FDR task.

Define $\Pi^{\text{STR}} = \langle F, A^{\text{FDR}}, s_0, g \rangle$ where

- $F = \{\langle v, d \rangle \mid v \in V, d \in \text{dom}(v)\}$,
- for $a \in A$ we have
  - $\text{pre}_{a^{\text{FDR}}} = \text{pre}_a$,
  - $\text{add}_{a^{\text{FDR}}} = \text{eff}_a$, and
  - $\text{del}_{a^{\text{FDR}}} = \{\langle v, d \rangle \mid \langle v, e \rangle \in \text{eff}_a, d \neq e\}$.

Simulations

How to construct heuristic

Let Π be a planning task Π, Θ_Π its induced LTS and s a state.

We relax/simplify Θ_Π so that its relaxed version Θ' can be solved in a reasonable time.

Next, we need a map $\alpha \colon S \to S'$ translating states in Θ to Θ'.

Finally, we find an optimal $\alpha(s)$-plan π' for Θ'.

Heuristic value $h(s) = \text{cost}(\pi')$.

Simulation

Definition

Let $\Theta = \langle S, A, T, G \rangle$ and $\Theta' = \langle S', A', T', G' \rangle$ be two LTSs. A pair $\langle R, \beta \rangle$ where $R \subseteq S \times S'$ and $\beta \colon A \to A'$ is called a simulation of Θ by Θ' if for all $s, t \in S$ and $s' \in S'$, we have

1. if $s \; R \; s'$ and $s \in G$, then $s' \in G'$,
2. if $s \; R \; s'$ and $s \xrightarrow{a} t$, then there is $t' \in S'$ such that $s' \xrightarrow{\beta(a)} t'$ and $t \; R \; t'$,
3. $\mathrm{cost}(\beta(a)) \leq \mathrm{cost}(a)$ for all $a \in A$.

We extend the map $\beta \colon A \to A'$ to sequences of actions. If $\pi = a_1, \ldots, a_n$, then $\beta(\pi) = \beta(a_1), \ldots, \beta(a_n)$.

Simulation preserves plans

Lemma

Let $\langle R, \beta \rangle$ *be a simulation of an LTS* $\Theta = \langle S, A, T, G \rangle$ *by an LTS* $\Theta' = \langle S', A', T', G' \rangle$. *Further, let* $s_0 \in S$ *and* $s'_0 \in S'$ *such that* $s_0 \; R \; s'_0$. *If* π *is a* s_0*-plan for* Θ*, then* $\beta(\pi)$ *is a* s'_0*-plan for* Θ' *as well.*

Corollary

Let $\langle R, \beta \rangle$ *be a simulation of an LTS* Θ *by an LTS* Θ' *and* $s_0 \; R \; s'_0$. *Let* π' *be an optimal* s'_0*-plan for* Θ' *and* $\mathrm{cost}(\pi')$ *its cost. Then* $\mathrm{cost}(\pi') \leq \mathrm{cost}(\pi)$ *for any* s_0*-plan* π *for* Θ.

Admissible heuristic

Let $\langle R, \beta \rangle$ be a simulation of Θ by Θ' and $\alpha \colon S \to S'$.

α is compatible with R if $\alpha \subseteq R$.

α is LTS homomorphism if $\alpha = R$.

Theorem

Let Π be a planning task, $\Theta_\Pi = \langle S, A, T, G \rangle$ its LTS, $\Theta' = \langle S', A', T', G' \rangle$ an LTS, h' the perfect heuristic for Θ', $\langle R, \beta \rangle$ a simulation of Θ_Π by Θ', and $\alpha \colon S \to S'$ compatible with R. Define $h(s) = h'(\alpha(s))$ for $s \in S$. Then h is admissible.

Delete relaxation

Definition

Let $\Pi = \langle F, A, s_0, g \rangle$ be a STRIPS task. For action $a = \langle \text{pre}_a, \text{add}_a, \text{del}_a \rangle \in A$, the corresponding delete relaxed action $a^+ = \langle \text{pre}_a, \text{add}_a, \emptyset \rangle$. The cost $\text{cost}(a^+) = \text{cost}(a)$.

The delete relaxation of Π is the STRIPS task $\Pi^+ = \langle F, A^+, s_0, g \rangle$ where $A^+ = \{a^+ \mid a \in A\}$.

Lemma

Let $\Pi = \langle F, A, s_0, g \rangle$ be a STRIPS task, $\Pi^+ = \langle F, A^+, s_0, g \rangle$ its delete relaxation, and $\beta : A \to A^+$ defined by $\beta(a) = a^+$. Then $\langle \subseteq, \beta \rangle$ is a simulation of Θ_Π by Θ_{Π^+}.

For a state s in Π, we define a heuristic $h^+(s) = h^*_+(s)$ where h^*_+ is the perfect heuristic for Π^+.

Corollary
h^+ is admissible.

Theorem
h^+ is consistent.

Plan existence

Let $\Pi^+ = \langle F, A^+, s_0, g \rangle$ be a delete relaxation.

For a state $s \subseteq F$, let $A_s = \{a \in A \mid \text{pre}_a \subseteq s\}$.

We define an operator $\Gamma \colon 2^F \to 2^F$ by

$$\Gamma(s) = s \cup \bigcup_{a \in A_s} a^+(s) = s \cup \bigcup_{a \in A_s} \text{add}_a.$$

$$s \subseteq \Gamma(s) \subseteq \Gamma(\Gamma(s)) \subseteq \cdots$$

The above sequence has a fixed point, i.e., there is $k \in \mathbb{N}$ such that $\Gamma^{k+1}(s) = \Gamma^k(s)$.

Theorem

Let Π be a STRIPS task. The plan existence problem for Π^+ belongs to P.

Theorem

Let Π be a STRIPS task. The decision problem whether $h^+(s) \leq m$ for a given $m \in \mathbb{N}$ is NP-complete.