Encoding Partial Plans for Heuristic Search

Pascal Bercher and Susanne Biundo

Institute of Artificial Intelligence

June 10th, 2013

ulm university universität ulm

Outline

(**POCL** Planning == **P**artial-**O**rder **C**ausal-**L**ink Planning)

A planning domain is a tuple $\mathcal{D} = \langle \mathcal{V}, \mathcal{A} \rangle$ with:

- $\mathcal{V}$ is a finite set of state variables, $s \in 2^{\mathcal{V}}$ being a state,
- $\mathcal{A}$ is a finite set of actions,
  an action $a := \langle pre, add, del \rangle \in \mathcal{A}$ consists of:
  - $pre \subseteq \mathcal{V}$, the precondition of a,
  - $add \subseteq \mathcal{V}$, the add list of a,
  - $del \subseteq \mathcal{V}$, the delete list of a

A planning domain is a tuple $\mathcal{D} = \langle \mathcal{V}, \mathcal{A} \rangle$ with:

- $\mathcal{V}$ is a finite set of state variables, $s \in 2^{\mathcal{V}}$ being a state,
- $\mathcal{A}$ is a finite set of actions,
  an action $a := \langle pre, add, del \rangle \in \mathcal{A}$ is applicable:
  - in a state $s \in 2^{\mathcal{V}}$ iff $pre \subseteq s$,
  - and generates the state $(s \setminus del) \cup add$
  - (applicability of action sequences is defined as usual)

A STRIPS planning problem is a tuple $\pi = \langle \mathcal{D}, s_{init}, g \rangle$ with:

- $\mathcal{D}$ is the planning domain,
- s_{init} is the initial state,
- g is the goal description

A solution to π is an action sequence $\bar{a}$, s.t.

- $\bar{a}$ is applicable in s_{init},
- $\bar{a}$ generates a state $s' \supseteq g$

A POCL planning problem is a tuple $\pi = \langle \mathcal{D}, P_{init} \rangle$ with:

- $\mathcal{D}$ is the planning domain
- P_{init} is the initial plan (actions; partially ordered)

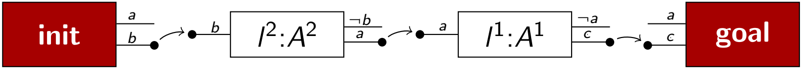
A solution to π is a plan P, s.t.:

- for every action sequence $\bar{a}$ induced by P holds:
  - $\bar{a}$ is applicable in s_{init},
  - $\bar{a}$ generates a state $s' \supseteq g$

A partial plan is a tuple $P = (PS, \prec, CL)$ with:

- PS is a finite set of plan steps,
  a plan step $l{:}a \in PS$ is a labeled action,
- $\prec$ is a strict partial order on PS,
- CL is a set of causal links between the plan steps in PS

Example for a partial plan:



$\mathcal{A} = \{A^1, A^2\}$ with: $A^1 = (a, \neg a \wedge c)$ and $A^2 = (b, \neg b \wedge a)$
initial state $s_{init} = a \wedge b$, goal description $g = a \wedge c$

Flaws	Modifications

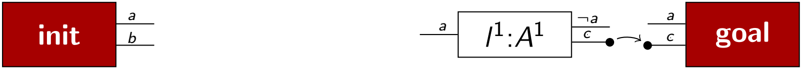
$\mathcal{A} = \{A^1, A^2\}$ with: $A^1 = (a, \neg a \wedge c)$ and $A^2 = (b, \neg b \wedge a)$
initial state $s_{init} = a \wedge b$, goal description $g = a \wedge c$

Flaws	Modifications
Open Precondition: $(a, goal)$	Causal Link: $(A^2, a, goal)$ Causal Link: $(init, a, goal)$
Open Precondition: $(c, goal)$	Causal Link: $(A^1, c, goal)$

$$\mathcal{A} = \{A^1, A^2\} \text{ with: } A^1 = (a, \neg a \wedge c) \text{ and } A^2 = (b, \neg b \wedge a)$$
$$\text{initial state } s_{init} = a \wedge b, \quad \text{goal description } g = a \wedge c$$

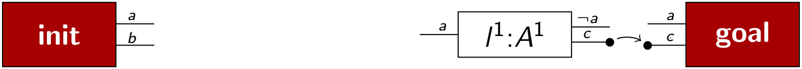
Flaws	Modifications
Open Precondition: $(a, goal)$	Causal Link: $(A^2, a, goal)$ Causal Link: $(init, a, goal)$
Open Precondition: $(c, goal)$	Causal Link: $(A^1, c, goal)$

$\mathcal{A} = \{A^1, A^2\}$ with: $A^1 = (a, \neg a \wedge c)$ and $A^2 = (b, \neg b \wedge a)$
initial state $s_{init} = a \wedge b$, goal description $g = a \wedge c$



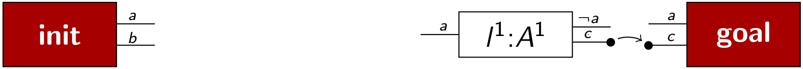
Flaws	Modifications
Open Precondition: $(a, goal)$	Causal Link: $(A^2, a, goal)$ Causal Link: $(init, a, goal)$
Open Precondition: $(c, goal)$	Causal Link: $(A^1, c, goal)$

$\mathcal{A} = \{A^1, A^2\}$ with: $A^1 = (a, \neg a \wedge c)$ and $A^2 = (b, \neg b \wedge a)$
initial state $s_{init} = a \wedge b$, goal description $g = a \wedge c$



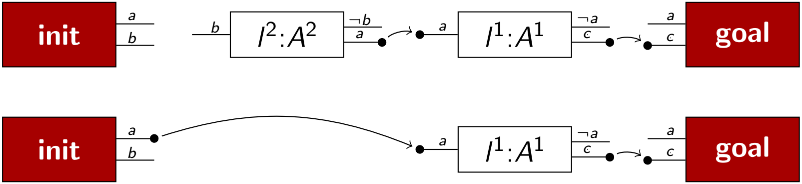
Flaws	Modifications
Open Precondition: $(a, goal)$	Causal Link: $(A^2, a, goal)$ Causal Link: $(init, a, goal)$
Open Precondition: $(a, l^1{:}A^1)$	Causal Link: $(A^2, a, l^1{:}A^1)$ Causal Link: $(init, a, l^1{:}A^1)$

$\mathcal{A} = \{A^1, A^2\}$ with: $A^1 = (a, \neg a \wedge c)$ and $A^2 = (b, \neg b \wedge a)$
initial state $s_{init} = a \wedge b$, goal description $g = a \wedge c$



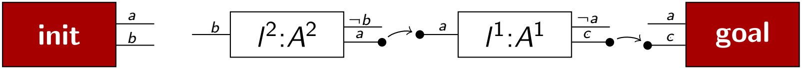
Flaws	Modifications
Open Precondition: $(a, goal)$	Causal Link: $(A^2, a, goal)$ Causal Link: $(init, a, goal)$
Open Precondition: $(a, I^1{:}A^1)$	Causal Link: $(A^2, a, I^1{:}A^1)$ Causal Link: $(init, a, I^1{:}A^1)$

$\mathcal{A} = \{A^1, A^2\}$ with: $A^1 = (a, \neg a \wedge c)$ and $A^2 = (b, \neg b \wedge a)$
initial state $s_{init} = a \wedge b$, goal description $g = a \wedge c$



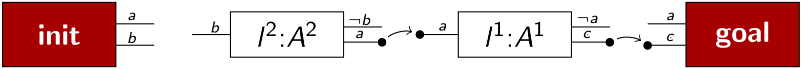
Flaws	Modifications
Open Precondition: $(a, goal)$	Causal Link: $(A^2, a, goal)$ Causal Link: $(init, a, goal)$
Open Precondition: $(a, I^1{:}A^1)$	Causal Link: $(A^2, a, I^1{:}A^1)$ Causal Link: $(init, a, I^1{:}A^1)$

$\mathcal{A} = \{A^1, A^2\}$ with: $A^1 = (a, \neg a \wedge c)$ and $A^2 = (b, \neg b \wedge a)$
initial state $s_{init} = a \wedge b$, goal description $g = a \wedge c$



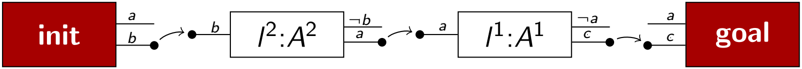
Flaws	Modifications
Open Precondition: $(a, goal)$	Causal Link: $(A^2, a, goal)$ Causal Link: $(init, a, goal)$ Causal Link: $(l^2{:}A^2, a, goal)$
Open Precondition: $(b, l^2{:}A^2)$	Causal Link: $(init, b, l^2{:}A^2)$

$\mathcal{A} = \{A^1, A^2\}$ with: $A^1 = (a, \neg a \wedge c)$ and $A^2 = (b, \neg b \wedge a)$
initial state $s_{init} = a \wedge b$, goal description $g = a \wedge c$



Flaws	Modifications
Open Precondition: $(a, goal)$	Causal Link: $(A^2, a, goal)$ Causal Link: $(init, a, goal)$ Causal Link: $(l^2{:}A^2, a, goal)$
Open Precondition: $(b, l^2{:}A^2)$	Causal Link: $(init, b, l^2{:}A^2)$

$\mathcal{A} = \{A^1, A^2\}$ with: $A^1 = (a, \neg a \wedge c)$ and $A^2 = (b, \neg b \wedge a)$
initial state $s_{init} = a \wedge b$, goal description $g = a \wedge c$



Flaws	Modifications
Open Precondition: $(a, goal)$	Causal Link: $(A^2, a, goal)$ Causal Link: $(init, a, goal)$ Causal Link: $(l^2{:}A^2, a, goal)$
Open Precondition: $(b, l^2{:}A^2)$	Causal Link: $(init, b, l^2{:}A^2)$

Algorithm 1: POCL planning procedure

$Fringe \leftarrow \{P_{init}\}$

while $Fringe \neq \emptyset$ **do**

  $P \leftarrow$ `planSel` $(Fringe)$

  $Fringe \leftarrow Fringe \setminus \{P\}$

  if $Flaws(P) = \emptyset$ **then return** P

  $f \leftarrow$ `flawSel` $(Flaws(P))$

  $Fringe \leftarrow Fringe \cup \{ \texttt{applyMod}(P, m) \mid m \in Mods(f) \}$

return *fail*

→ two choices:
plan selection (`planSel`) and **flaw selection** (`flawSel`)

Problem:

- Desired: goal-distance estimates for *plans*
- Already available: goal-distance estimates for *states*

Idea:

Encode a *plan* P by means of new planning problem $\mathcal{P}'$ s.t.:

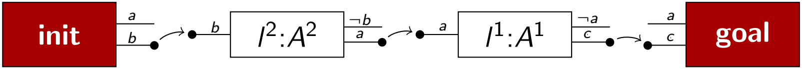
$$\begin{aligned} \text{solutions of } \mathcal{P}' \quad &\equiv \quad \text{solutions reachable from } P \\ \rightarrow \text{goal distance of } P \quad &\equiv \quad \text{goal distance of the initial state of } \mathcal{P}' \end{aligned}$$

Let $\pi = \langle\langle\mathcal{V}, \mathcal{A}\rangle, s_{init}, g\rangle$ be a planning problem and $P = (PS, \prec, CL)$ a plan.

The encoding of P is given by $\pi' = \langle\langle\mathcal{V}', \mathcal{A}'\rangle, s'_{init}, g'\rangle$ with:

- $\mathcal{A}' := \mathcal{A} \cup \mathcal{A}_{new}$, with $\mathcal{A}_{new} := \{\ enc(l{:}A) \mid l{:}A \in PS\ \}$
- $\mathcal{A}_{new}$ encodes the plan steps in PS, s.t.:
  - each $a \in \mathcal{A}_{new}$ is executable exactly once
  - the actions in $\mathcal{A}_{new}$ can only be inserted in an order consistent with the one in PS
- The goal description is altered, s.t. all actions in $\mathcal{A}_{new}$ have to be executed

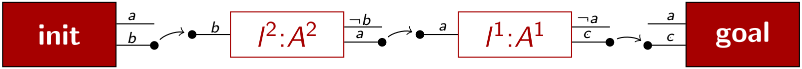
Consider the following plan P



The encoding of P is given by $\pi' = \langle\langle\mathcal{V}', \mathcal{A}'\rangle, s'_{init}, g'\rangle$ with:

- $\mathcal{V}' := \mathcal{V} \cup \{l^1, l^2\}$
- $\mathcal{A}' := \mathcal{A} \cup \{enc(l^1{:}A^1), enc(l^1{:}A^1)\}$ with
  - $enc(l^1{:}A^1) = \langle a \wedge \neg l^1 \wedge l^2, \neg a \wedge c \wedge l^1\rangle$
  - $enc(l^2{:}A^2) = \langle b \wedge \neg l^2, \neg b \wedge a \wedge l^2\rangle$
- $s'_{\text{init}} := s_{\text{init}}$
- $g' := g \cup \{l^1, l^2\}$

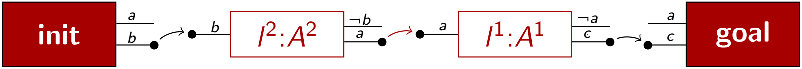
Consider the following plan P



The encoding of P is given by $\pi' = \langle\langle\mathcal{V}', \mathcal{A}'\rangle, s'_{init}, g'\rangle$ with:

- $\mathcal{V}' := \mathcal{V} \cup \{l^1, l^2\}$
- $\mathcal{A}' := \mathcal{A} \cup \{enc(l^1{:}A^1), enc(l^1{:}A^1)\}$ with
  - $enc(l^1{:}A^1) = \langle a \wedge \neg l^1 \wedge l^2, \neg a \wedge c \wedge l^1\rangle$
  - $enc(l^2{:}A^2) = \langle b \wedge \neg l^2, \neg b \wedge a \wedge l^2\rangle$
- $s'_{\text{init}} := s_{\text{init}}$
- $g' := g \cup \{l^1, l^2\}$

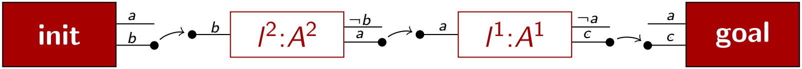
Consider the following plan P



The encoding of P is given by $\pi' = \langle\langle\mathcal{V}', \mathcal{A}'\rangle, s'_{init}, g'\rangle$ with:

- $\mathcal{V}' := \mathcal{V} \cup \{l^1, l^2\}$
- $\mathcal{A}' := \mathcal{A} \cup \{enc(l^1{:}A^1), enc(l^1{:}A^1)\}$ with
  - $enc(l^1{:}A^1) = \langle a \land \neg l^1 \land l^2, \neg a \land c \land l^1\rangle$
  - $enc(l^2{:}A^2) = \langle b \land \neg l^2, \neg b \land a \land l^2\rangle$
- $s'_{\text{init}} := s_{\text{init}}$
- $g' := g \cup \{l^1, l^2\}$

Consider the following plan P



The encoding of P is given by $\pi' = \langle\langle\mathcal{V}', \mathcal{A}'\rangle, s'_{init}, g'\rangle$ with:

- $\mathcal{V}' := \mathcal{V} \cup \{l^1, l^2\}$
- $\mathcal{A}' := \mathcal{A} \cup \{enc(l^1{:}A^1), enc(l^1{:}A^1)\}$ with
  - $enc(l^1{:}A^1) = \langle a \wedge \neg l^1 \wedge l^2, \neg a \wedge c \wedge l^1\rangle$
  - $enc(l^2{:}A^2) = \langle b \wedge \neg l^2, \neg b \wedge a \wedge l^2\rangle$
- $s'_{\text{init}} := s_{\text{init}}$
- $g' := g \cup \{l^1, l^2\}$

Let P be a plan and $\pi' = \langle\langle\mathcal{V}', \mathcal{A}'\rangle, s'_{init}, g'\rangle$ its encoding.

Then, use $h(P) := \max\{0, h_{sb}(s'_{init}) - cost(P)\}$ as heuristic.
(with h_{sb} being a heuristic for state-based planning)

Note: if h_{sb} is admissible, so is h!

Non-incremental encoding:

- a plan $P = (PS, \prec, \emptyset)$ can be encoded in $O(|\prec|) = O(|PS|^2)$
- *(a plan $P = (PS, \prec, CL)$ can be be encoded in* **P***)*

Incremental encoding:

- given P, its encoding π', and an applied modification m, the encoding of P', the successor of P w.r.t. m, can be calculated in $O(|P|)$ or $O(1)$.

Summary of Encoding:

- can be done efficiently
- enables the use of state-based heuristics in POCL planning
- provides the *first admissible heuristics* for POCL planning

Outlook:

- Empirical evaluation (not in the paper: LM cut works fine)
- Adapting heuristics, rather than blindly applying them