On Delete Relaxation in Partial-Order Causal-Link Planning

Pascal Bercher and Thomas Geier
and Felix Richter and Susanne Biundo

Institute of Artificial Intelligence

November 5th, 2013

ulm university universität uulm

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:

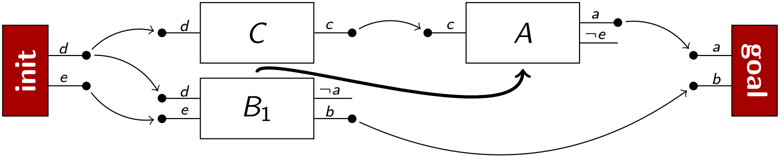
- $\mathcal{V}$ is a finite set of state variables, $s \in 2^{\mathcal{V}}$ being a state,
- $\mathcal{A}$ is a finite set of actions,
  an action $a := \langle pre, add, del \rangle \in \mathcal{A}$ consists of:
  - $pre \subseteq \mathcal{V}$, the precondition of a,
  - $add \subseteq \mathcal{V}$,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

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

- $\mathcal{D}$ is the planning domain
- P_{init} is the initial plan



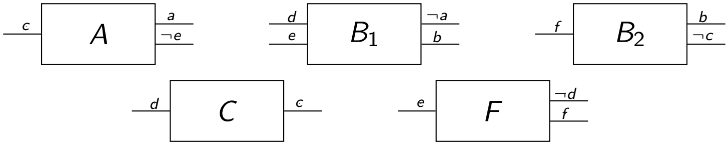
A POCL planning problem is a tuple $\mathcal{P} = \langle \mathcal{D}, P_{init} \rangle$ with:

- $\mathcal{D}$ is the planning domain
- P_{init} is the initial plan

A solution to $\mathcal{P}$ is a plan P, s.t.:

- every precondition is supported by a causal link
- there are no causal threats

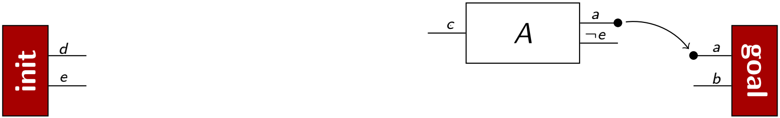
Planning domain $\mathcal{D}$:



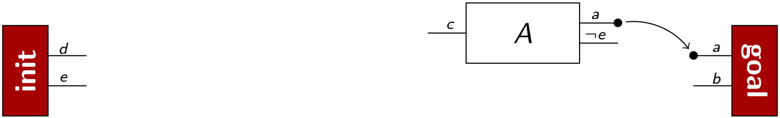
Initial plan P_{init}:

flaws:	modifications:
open precondition: (a:goal)	A
open precondition: (b:goal)	B_1, B_2

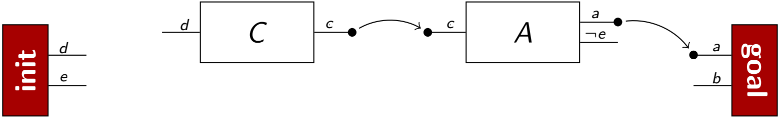
flaws:	modifications:
open precondition: (a:goal)	A
open precondition: (b:goal)	B_1, B_2



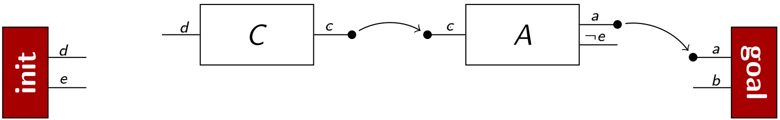
flaws:	modifications:
open precondition: (b:goal)	B_1, B_2
open precondition: (c:A)	C



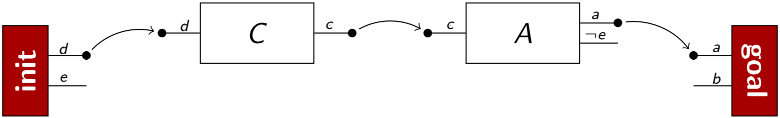
flaws:	modifications:
open precondition: (b:goal)	B_1, B_2
open precondition: (c:A)	C



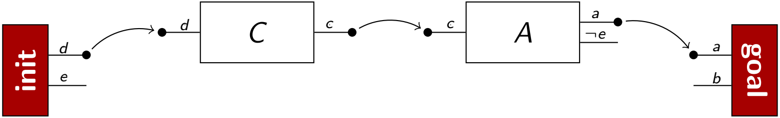
flaws:	modifications:
open precondition: (b:goal)	B_1, B_2
open precondition: (d:C)	init



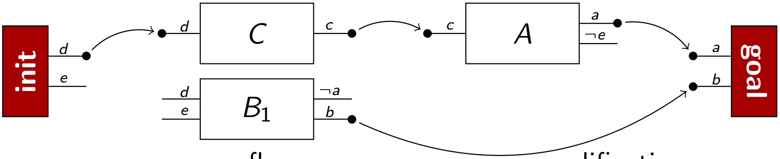
flaws:	modifications:
open precondition: (b:goal)	B_1, B_2
open precondition: (d:C)	init



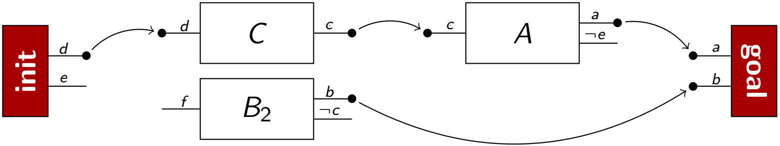
flaws:	modifications:
open precondition: (b:goal)	B_1, B_2



flaws:	modifications:
open precondition: (b:goal)	B_1, B_2



flaws:	modifications:
open precondition: (d:B_1)	init
open precondition: (e:B_1)	init
causal threat: (B_1:$\neg a$),(A,goal)	$B_1 < A$

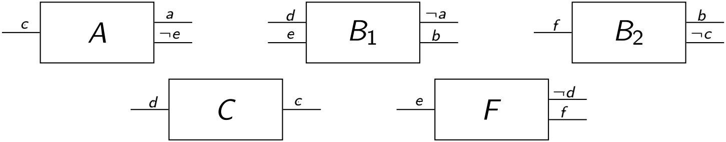


flaws:	modifications:
open precondition: (f:B_2)	F
causal threat: (B_2:$\neg c$),(C,A)	$B_2 < C, A < B_2$

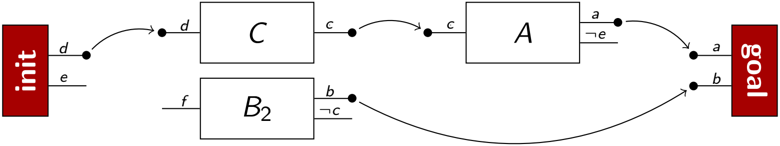
We need a well-informed plan-to-solution distance
(in terms of number of actions/modifications)

- **How hard is that estimate? (theoretically)**
- **How to estimate? (practically)**

Planning domain $\mathcal{D}$:

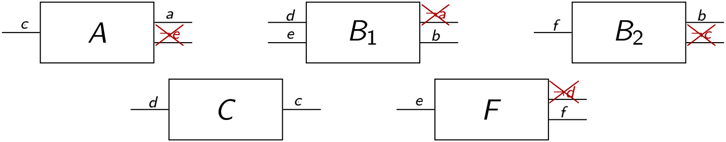


Plan to estimate:

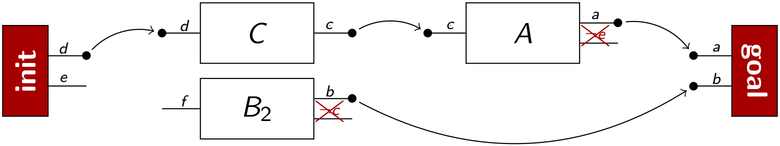


flaws:	modifications:
open precondition: (f:B_2)	F
causal threat: (B_2:$\neg c$),(C,A)	$B_2 < C, A < B_2$

Planning domain $\mathcal{D}$:

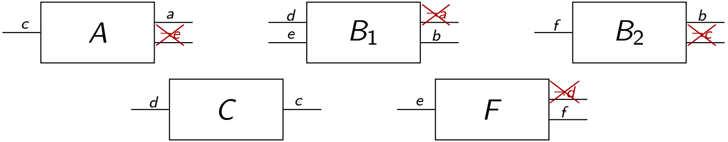


Plan to estimate:

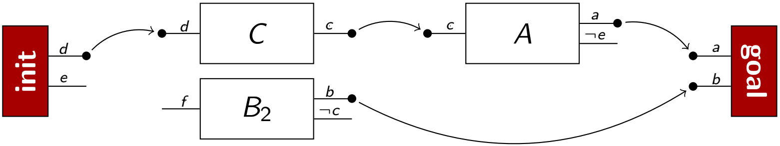


flaws:	modifications:
open precondition: (f:B_2)	F
~~causal threat: (B_2:$\neg c$),(C,A)~~	~~$B_2 < C$, $A < B_2$~~

Planning domain $\mathcal{D}$:



Plan to estimate:



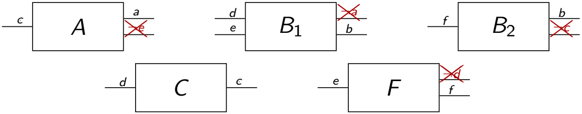
flaws:	modifications:
open precondition: (f:B_2)	F
causal threat: (B_2:$\neg c$),(C,A)	$B_2 < C, A < B_2$

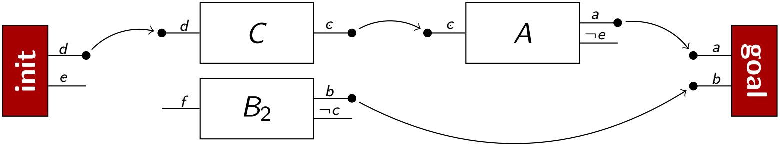
Theorem:

It is in **NP** to decide whether there is a solution for a relaxed planning domain and a non-relaxed plan.

Proof:

- Guess a non-conflicting linearization
- Apply all relaxed actions in initial state
- Apply the first plan step in the resulting state
- Continue until the last plan step (including goal) is applied



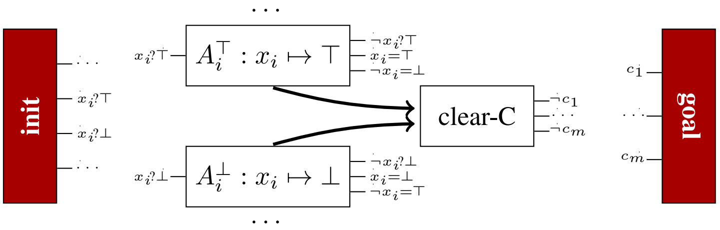


Theorem:

It is **NP-hard** to decide whether there is a solution for a relaxed planning domain and a non-relaxed plan.

Proof:

- Reduction from CNF-SAT. $X = \{x^1, \ldots, x^n\}$ is a set of boolean variables. $C = \{c^1, \ldots, c^m\}$ is a set of clauses; each c^j represents a disjunction of variables of X.
- Construct a delete relaxed domain and a non-relaxed plan, s.t. its solutions are isomorphic to the CNF-SAT's solutions.
- The order of $A_i^\top$ and $A_i^\perp$ encodes the truth assignment of $x^i \in X$. For $x^i \in c^j$, an action $C_{ij} = \langle\{x_i = \top\}, \{c_j\}, \emptyset\rangle$ encodes truth assignment of clause $c^j \in C$.



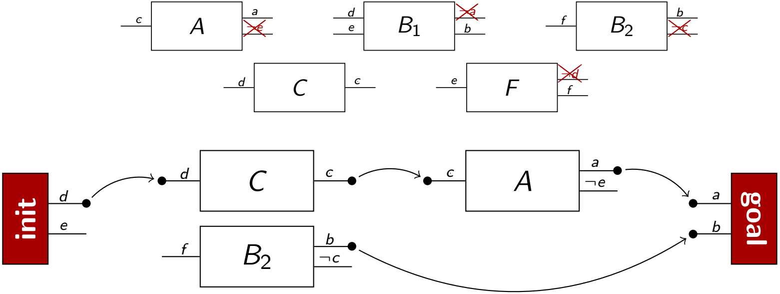
Idea:

- Sample a fixed number of linearizations
  (this approximates the "guessing" part")
- For each linearization: build relaxed solution as in the membership proof
- Heuristic estimate is cheapest solution

Properties:

- Does not ignore present causal links:
  - Only non-conflicting relaxed actions are used
  - Only non-conflicting linearizations are used
- Does not ignore negative effects of present plan steps

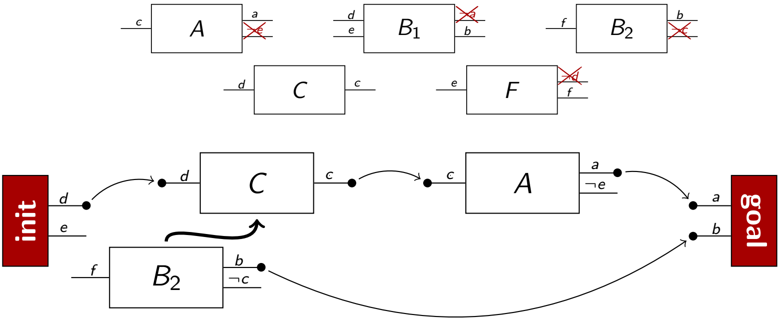
Example:



Properties:

- Does not ignore present causal links:
  - Only non-conflicting relaxed actions are used
  - Only non-conflicting linearizations are used
- Does not ignore negative effects of present plan steps

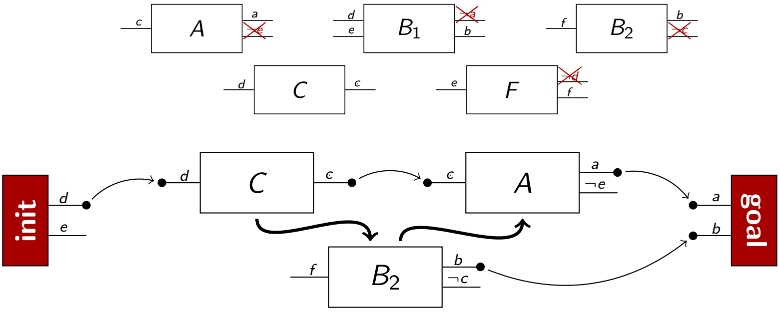
Example:



Properties:

- Does not ignore present causal links:
  - Only non-conflicting relaxed actions are used
  - Only non-conflicting linearizations are used
- Does not ignore negative effects of present plan steps

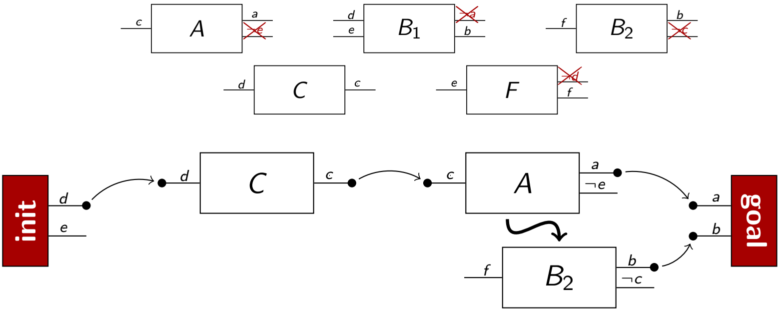
Example:



Properties:

- Does not ignore present causal links:
  - Only non-conflicting relaxed actions are used
  - Only non-conflicting linearizations are used
- Does not ignore negative effects of present plan steps

Example:



Setting:

- Evaluation was done on 20 domains taken from the first 5 International Planning Competitions (IPCs).
- Comparison was done with:
  - Add and Relax Heuristic (for POCL planning)
  - 12 variants of SampleFF
    (1, 3, 10, 30 samples, action filter feature on/off)

Results:

- number of solved problem instances overall (of 446)
  Add: 292, Relax: 227/194, Sample-FF: 127 to 187
- For one problem, only Sample-FF disproved it
- Depending on the Sample-FF configuration, 1% to 46% of all plans had only invalid linearizations

Summary:

- Delete relaxation only for the domain is **NP-complete**
- We developed a new heuristic for POCL planning (Sample-FF)
  - pro: more pruning power/tighter estimates
  - con: sampled linearizations are often *all* invalid

Outlook:

- Solve the "con" problem
- More intelligent sampling. Using causal graphs?
- Lifting the heuristic

Domain	*n*	**Add**	**Relax***	**Relax**	**Sample-FF**											
					front: $\bot$ end: $\bot$				front: $\bot$ end: $\top$				front: $\top$ end: $\top$			
					1	3	10	30	1	3	10	30	1	3	10	30
grid	5	0	0	0	0	0	0	0	0	0	0	0	1	**1**	**1**	**1**
gripper	20	14	20	7	1	1	1	1	1	2	1	1	2	**3**	**3**	2
logistics	20	12	8	7	**8**	5	6	6	6	7	6	5	0	0	1	1
movie	30	30	30	30	**30**	**30**	**30**	**30**	**30**	**30**	**30**	**30**	**30**	**30**	**30**	**30**
mystery	20	8	8	9	10	11	9	9	10	11	10	9	**12**	**12**	11	11
mystery-prime	20	3	3	3	3	4	**6**	5	5	4	4	4	**6**	**6**	**6**	**6**
blocks	21	4	5	7	5	5	**6**	**6**	4	3	3	3	5	3	2	0
logistics	28	28	28	27	22	23	23	**24**	21	19	20	21	15	13	14	15
miconic	100	100	49	39	**40**	**40**	37	35	39	41	37	32	15	16	18	20
depot	22	2	2	1	1	1	1	1	0	1	1	1	2	2	**3**	2
driverlog	20	7	9	7	**11**	9	10	9	9	10	9	8	8	7	9	7
rover	20	20	18	19	13	14	**15**	**15**	11	11	12	11	7	9	9	9
zeno-travel	10	4	5	3	3	**5**	4	**5**	4	3	4	4	1	1	1	1
airport	20	18	15	9	10	**11**	**11**	**11**	7	10	10	10	7	8	6	4
pipesworld-noTankage	10	8	1	2	2	3	**5**	3	2	1	2	1	1	4	3	**5**
pipesworld-Tankage	10	1	1	1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0	0
satellite	20	16	7	7	**7**	5	6	6	5	5	7	5	1	2	3	3
pipesworld	10	1	1	1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0	0
storage	20	7	6	4	6	7	9	8	6	7	7	6	9	9	**10**	**10**
tpp	20	19	11	11	**8**	7	6	7	6	6	6	6	5	6	6	6
total	446	292	227	194	182	183	187	183	168	173	171	159	127	132	136	133

Variables: $X = \{x^1, \dots, x^n\}$, Clauses: $C = \{c^1, \dots, c^m\}$

$$\mathcal{V} = \{x_i?\top, x_i?\bot, x_i{=}\top, x_i{=}\bot \mid i \in \{1, \dots, n\}\} \cup \{c_i \mid i \in \{1, \dots, m\}\}$$

$$\mathcal{A} = \{dr\text{-}A_i^\top, dr\text{-}A_i^\bot \mid i \in \{1, \dots, n\}\} \cup \{dr\text{-clear-C}\} \cup$$
$$\{C_{ij}^\top \mid c^j \in C, x^i \in c^j\} \cup \{C_{ij}^\bot \mid c^j \in C, \neg x^i \in c^j\}$$

$P = (PS, \prec, CL)$ and

$$PS = \{a_0, a_\infty, \text{clear-C}\} \cup \{A_i^\top, A_i^\bot \mid i \in \{1, \dots, n\}\}$$
$$\prec = \{(l_i^\top, l_C), (l_i^\bot, l_C) \mid i \in \{1, \dots, n\}\}$$
$$CL = \emptyset$$
$$a_0 = (\emptyset, \{x_i?\top, x_i?\bot \mid i \in \{1, \dots, n\}\}, \emptyset)$$
$$a_\infty = (\{c_i \mid i \in \{1, \dots, m\}\}, \emptyset, \emptyset)$$
$$A_i^\top = (\{x_i?\top\}, \{x_i{=}\top\}, \{x_i?\top, x_i{=}\bot\}), dr\text{-}A_i^\top = (\{x_i?\top\}, \{x_i{=}\top\}, \emptyset)$$
$$A_i^\bot = (\{x_i?\bot\}, \{x_i{=}\bot\}, \{x_i?\bot, x_i{=}\top\}), dr\text{-}A_i^\bot = (\{x_i?\bot\}, \{x_i{=}\bot\}, \emptyset)$$
$$\text{clear-C} = (\emptyset, \emptyset, \{c_1, \dots, c_m\}), dr\text{-clear-C} = (\emptyset, \emptyset, \emptyset)$$
$$C_{ij}^\top = (\{x_i{=}\top\}, \{c_j\}, \emptyset), C_{ij}^\bot = (\{x_i{=}\bot\}, \{c_j\}, \emptyset)$$