On Conformant Planning in A-Prolog

By
Michael Gelfond and A. Ricardo Morales

Conformant Planning

Given an incomplete specification of the initial state find a plan that leads to a state satisfing the given goals for each possible completion of the initial state.

Examples:

- Finding a drug therapy that covers all possible cases (without bad drug interactions).
- Bomb in the toilet.

Representation

Consider an action description *D* of language *B*

D consists of sentences of the form:

(1) *a* causes *l* if *p*.

(2) *l* if *p*.

(3) impossible *a* if *p*.

Where *a* is an action symbol, *l* is a fluent literal and *p* is a list of fluent literals.

Representation

We translate *D* into a logic program C_D, called the *cautious translation* of *D*.

C_D is obtained from *D* by replacing each:

- Rule (1) by:

  h(*l*,T+1) ← o(*a*,T), h(*p*,T).

  ab($\overline{l}$,T) ← o(*a*,T), not ¬h(*p*,T).

- Rule (2) by

  h(*l*,T+1) :-← h(*p*,T).

  ab(*l*,T) :-← ab(*p*,T).

- Rule (3) by

  ← o(*a*,T), not ¬h(*p*,T).

Representation

And including the inertia axiom:

- $h(L,T+1) \leftarrow h(L,T), \text{not } ab(L,T).$

Properties of C_D

- C_D is sound:
  - Models of C_D + “incomplete initial situation” + “planning module” contain conformant plans.

- C_D is complete (*):
  - If C_D + “incomplete initial situation” + “planning module” has no models then no conformant plan exists.

(*)

- C_D has no rules of type "*l* if *p*"
- If C_D has rules "*a* causes *l* if p_1." and "*a* causes *l* if p_2.", and p_1 is false in a completion σ of the initial state then p_2 is also false in σ.

Example: Bomb in the Toilet

dunk(P,E) causes ¬armed(P) if ¬clogged(E).

dunk(P,E) causes clogged(E).

flush(E) causes ¬clogged(E).

impossible dunk(P,E) if clogged(E).

Example: Bomb in the Toilet

```
-h(armed(P),T+1) :- o(dunk(P,E),T),
                   -h(cloged(E),T).

ab(armed(P),T) :- o(dunk(P,E),T),
                  not h(cloged(E),T).

h(cloged(E),T+1) :- o(dunk(P,E),T).
ab(cloged(E),T) :- o(dunk(P,E),T).

-h(cloged(E),T+1) :- o(flush(E),T).
ab(cloged(E),T) :- o(flush(E),T).

:- o(dunk(P,E),T), not -h(cloged(E),T),
```

Experimental Results (C-Plan)

P : T	CModels	SModels	DLV	C-Plan	planlenght
2 : 1	0.562	0.431	0.279	0.942	3
2 : 5	0.253	0.207	0.194	0.778	1
2 : 10	0.272	0.286	0.298	0.752	1
4 : 1	1.311	1.069	0.841	2.086	7
4 : 5	0.257	0.239	0.223	0.643	1
4 : 10	0.311	0.385	0.396	0.972	1
6 : 1	2.712	2.831	20.437	19.923	11
6 : 5	3.856	13.535	10.132	6.369	3
6 : 10	0.354	0.476	0.527	3.778	1
8 : 1	9.234	16.997	TIME	685.437	15
8 : 5	3.347	32.040	31.967	107.405	3
8 : 10	0.390	0.570	0.685	69.106	1
10 : 1	119.382	832.282	TIME	TIME	19
10 : 5	6.227	62.474	78.429	TIME	3
10 : 10	0.459	173.350	0.936	3.677	1

Experimental Results (DLVk)

P : T	CModels	DLVk	planlenght
2 : 2	0.329	0.174	1
3 : 2	0.264	0.100	3
4 : 2	0.268	0.097	3
5 : 2	0.314	0.126	5
6 : 2	0.380	0.298	5
7 : 2	0.543	3.103	7
8 : 2	1.977	94.765	7
9 : 2	78.689	TIME	9
10 : 2	114.262	TIME	9
2 : 3	0.223	0.112	1
3 : 3	0.225	0.070	1
4 : 3	0.292	0.107	3
5 : 3	0.327	0.119	3
6 : 3	0.413	0.208	3
7 : 3	1.702	0.589	5
8 : 3	1.840	2.406	5
9 : 3	1.813	95.239	5
10 : 3	4.788	TIME	7
2 : 4	0.232	0.065	1
3 : 4	0.228	0.029	1
4 : 4	0.240	0.077	1
5 : 4	0.482	0.146	3
6 : 4	0.625	0.168	3
7 : 4	0.995	0.284	3
8 : 4	1.576	1.277	3
9 : 4	5.636	3.842	5
10 : 4	3.092	376.950	5

THE END.

M

Properties of CD

- C_D is complete
-
-
- C_D is sound (under certain condition *)
-
-
- * There is at most one rule (2) with body p