

Planning

Planning	2
Planning Definitions	2
Planning States and Goals	3
Planning Actions	4
Partial Order Planning	5
Introduction to Partial Order Planning	5
Example	6
Partial Order Causal-Link Planning	7
Fixing Flaws	8
Planning Graphs	9
Introduction to Planning Graphs	9
Mutual Exclusion (Mutex) Links	10
"Simple" Planning Graph Example	11
Initial Action Level	12
Next Action Level	13
Graphplan Algorithm	14

Planning

Planning Definitions

- *Planning* is finding actions to achieve goals.
- A *fluent* is a proposition that changes over time.
- A *state* are the fluents that are true.
- The *goal* are the fluents that should be true and false.
- An *action* is described by the changes that occur.
- *Search* can perform planning. Planning states and actions map to search states and actions.
- A *progression* planner searches from the initial state to the goal. A *regression* planner searches from the goal to the initial state.

Planning States and Goals

In PDDL (Planning Domain Definition Language) states and goals are conjunctions of ground, function-free literals.

- Example: Blocks-world with a table T and three blocks named A , B , and C .
- Possible Literals:
 $on(A, T), on(A, B), on(A, C), \dots$
 $clear(A), clear(B), clear(C)$
- The state where C is on A and B is by itself:
 $clear(C) \wedge on(C, A) \wedge on(A, T)$
 $\wedge clear(B) \wedge on(B, T)$
- The goal to have A on B , and B on C :
 $on(A, B) \wedge on(B, C)$

Planning Actions

- An action schema can be specified by:
 - *name* and *parameters* of action
 - *preconditions*: what literals must be true
 - *effects*: what becomes true (*add list*) and false (*delete list*)

- Action schema to move block x from y to z :

$Action(move(x, y, z),$
 $PRECOND: clear(x) \wedge on(x, y) \wedge clear(z)$
 $EFFECT: clear(y) \wedge on(x, z) \wedge on(x, y) \wedge clear(z))$

- Create a complete, correct set of action schema for the blocks world.

CS 3793 Artificial Intelligence

Planning - 4

Partial Order Planning

5

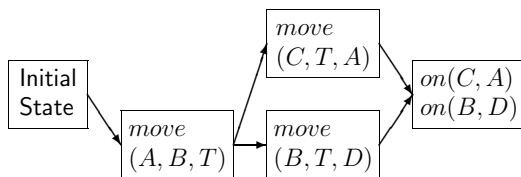
Introduction to Partial Order Planning

- Progression and regression planning require that actions be totally ordered.
- Partial order planning only specifies those orderings that are needed.
- Example: Blocks-world with a table T and four blocks, A , B , C , and D .
- $Init(clear(A) \wedge on(A, B) \wedge on(B, T) \wedge clear(C)$
 $\wedge on(C, T) \wedge clear(D) \wedge on(D, T))$
 $Goal(on(C, A) \wedge on(B, D))$

CS 3793 Artificial Intelligence

Planning - 5

Example



CS 3793 Artificial Intelligence

Planning - 6

Partial Order Causal-Link Planning

A partial-order plan consists of the following:

- A set of actions. A start step and finish step encode the initial state and goals.
- A set of orderings on the actions.
- A set of causal links. Each causal link goes from an effect of one action to a precondition of another action.

CS 3793 Artificial Intelligence

Planning - 7

Fixing Flaws

A flaw in a partial-order plan is:

- a precondition that is not supported by a causal link, or
- a causal link that is threatened by another action (the threat).

A flaw can be fixed by:

- adding a causal link, possibly adding an action to support the causal link, or
- ordering the threat before the causal link ("demotion") or after ("promotion").

CS 3793 Artificial Intelligence

Planning - 8

Planning Graphs

9

Introduction to Planning Graphs

- A *planning graph* is a sequence of levels corresponding to "time steps," alternating between "state levels" S_i and "action levels" A_i .
- It starts with state level S_0 , which contains the literals true of the initial state.
- An action is in A_i if its preconds. are in S_i . It has edges from its preconds. in S_i and to its effects in S_{i+1} .
- Also, each literal in S_i has a persistence edge to the same literal in S_{i+1} .

CS 3793 Artificial Intelligence

Planning - 9

Mutual Exclusion (Mutex) Links

- Two actions in A_i have a mutex link if a precondition or effect of one action conflicts with a precondition or effect of the other action.
- Two literals in S_{i+1} have a mutex link if one negates the other or if every pair of actions in A_i achieving them are mutex.
- An action cannot be in A_i if any two preconds. in S_i are mutex.

CS 3793 Artificial Intelligence

Planning - 10

"Simple" Planning Graph Example

Init($have(Cake)$)

Goal($have(Cake) \wedge eaten(Cake)$)

Action($eat(Cake)$,

PRECOND: $have(Cake)$

EFFECT: $eaten(Cake) \wedge \neg have(Cake)$)

Action($bake(Cake)$,

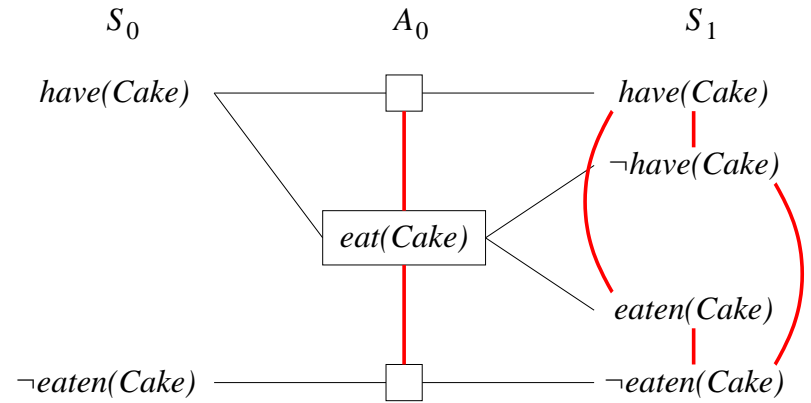
PRECOND: $\neg have(Cake)$

EFFECT: $have(Cake)$)

CS 3793 Artificial Intelligence

Planning - 11

Initial Action Level

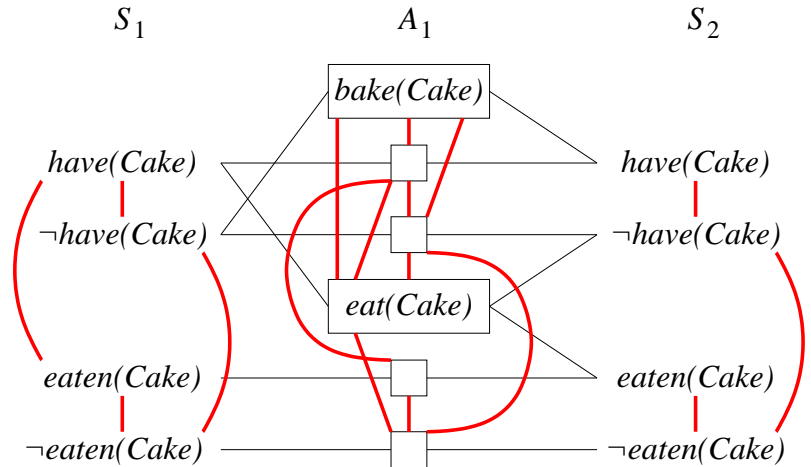


$have(Cake)$ and $eaten(Cake)$ in S_1 are mutex because keeping $have(Cake)$ is mutex with $eat(Cake)$.

CS 3793 Artificial Intelligence

Planning - 12

Next Action Level



Both actions to achieve $\neg have(Cake)$ are mutex with the one action to achieve $\neg eaten(Cake)$.

CS 3793 Artificial Intelligence

Planning - 13

Graphplan Algorithm

The Graphplan algorithm using the planning graph to search for a plan.

1. A state is a subset of literals in a state level.
2. There is an edge from a subset of S_i to a subset of S_{i-1} if there is a mutex-free subset of A_{i-1} with respective effects and preconds.
3. The initial search state is the set of goals on the last level S_n of the planning graph.
4. The goal is to reach S_0 .