FAI LAB 10: Classical Planning

Paolo Morettin

2024-25

Classical planning

- Fully-observable, deterministic, static, single agent
- Like Ch.3, but with a **structured/factored** state representation

Classical planning

- Fully-observable, deterministic, static, single agent
- Like Ch.3, but with a **structured/factored** state representation

Given:

- The **initial state** S
- A **goal** G (a set of states)
- A set of **actions**

Find a *totally/partially ordered sequence of actions from S to G* (**plan**)

Planning Domain Definition Language (PDDL)

State: a conjunction of fluents

- **Fluent**: conjunction of function-less ground atoms
- **Closed-world assumption**: non-mentioned fluents are false
- **Unique-name assumption**: different constants $\rightarrow$ different objects

$$At(Paolo, Blackboard) \land \neg At(Paolo, Bed) \land \neg At(Paolo, PC)$$

Planning Domain Definition Language (PDDL)

State: a conjunction of fluents

- **Fluent**: conjunction of function-less ground atoms
- **Closed-world assumption**: non-mentioned fluents are false
- **Unique-name assumption**: different constants $\rightarrow$ different objects

$$At(Paolo, Blackboard)$$

Planning Domain Definition Language (PDDL)

State: a conjunction of fluents

- **Fluent**: conjunction of function-less ground atoms
- **Closed-world assumption**: non-mentioned fluents are false
- **Unique-name assumption**: different constants $\rightarrow$ different objects

Goal:

- Conjunction of literals (variables implicitly $\exists$-quantified)
- Set of states

Planning Domain Definition Language (PDDL)

State: a conjunction of fluents

- **Fluent**: conjunction of function-less ground atoms
- **Closed-world assumption**: non-mentioned fluents are false
- **Unique-name assumption**: different constants $\rightarrow$ different objects

Goal:

- Conjunction of literals (variables implicitly $\exists$-quantified)
- Set of states

Action schemata:

- Action name / variables (implicitly $\forall$-quantified)
- Precondition / effect - conjunction of literals

$$\begin{aligned}&\mathit{Action}(\mathit{Move}(p,x,y),\\&\qquad \mathit{PRE}: \mathit{At}(p,x) \wedge \mathit{CanMove}(p) \wedge \mathit{Reachable}(x,y),\\&\qquad \mathit{EFF}: \mathit{At}(p,y) \wedge \neg \mathit{At}(p,x))\end{aligned}$$

Planning Domain Definition Language (PDDL)

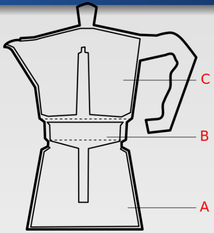
$$\begin{aligned}&\mathit{Action}(\mathit{Move}(p, x, y),\\ &\qquad \mathit{PRE} : \mathit{At}(p, x) \land \mathit{CanMove}(p) \land \mathit{Reachable}(x, y)\\ &\qquad \mathit{EFF} : \mathit{At}(p, y) \land \neg \mathit{At}(p, x))\end{aligned}$$

Executing actions:

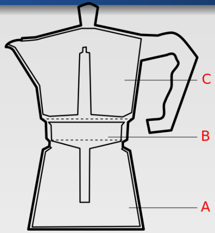
- Actions schema are **instantiated** by grounding variables
- **Executable** action: the ground precondition holds
- Add list ($\mathit{Add}(a)$): the list of positive ground literals in EFF
- Delete list ($\mathit{Del}(a)$): the list of negative ground literals in EFF
- $\mathit{Result}(s, a) = (s - \mathit{Del}(a)) \cup \mathit{Add}(a))$

$$\begin{aligned}&s1 = \mathit{CanMove}(\mathit{Paolo}) \land \mathit{At}(\mathit{Paolo}, \mathit{Lab}) \land \mathit{Reachable}(\mathit{Lab}, \mathit{Bed})\\ &\mathit{Result}(s1, \mathit{Move}(\mathit{Paolo}, \mathit{Lab}, \mathit{Bed})) = s2\\ &s2 = \mathit{CanMove}(\mathit{Paolo}) \land \mathit{At}(\mathit{Paolo}, \mathit{Bed}) \land \mathit{Reachable}(\mathit{Lab}, \mathit{Bed})\end{aligned}$$

PDDL encoding



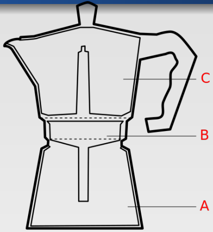
PDDL encoding



$$\begin{aligned}\text{Initial:}\;\; & \mathit{Contains}(A, \mathit{None}) \wedge \mathit{Contains}(B, \mathit{None}) \\ & \wedge \mathit{Contains}(C, \mathit{None}) \wedge \mathit{Ingredient}(\mathit{Water}) \\ & \wedge \mathit{Ingredient}(\mathit{GroundCoffee})\end{aligned}$$

$$\text{Goal:}\;\; \mathit{Contains}(C, \mathit{Coffee})$$

PDDL encoding

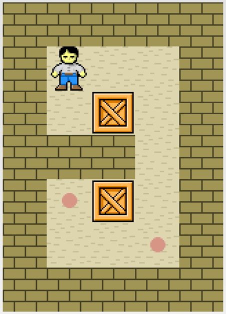


Initial: $Contains(A, None) \wedge Contains(B, None)$
$\wedge\ Contains(C, None) \wedge Ingredient(Water)$
$\wedge\ Ingredient(GroundCoffee)$

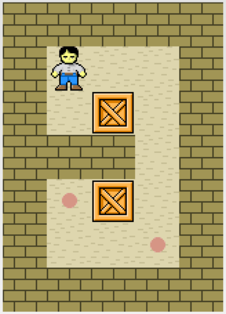
Goal: $Contains(C, Coffee)$

$Action(Fill(c, i),$
$\quad PRE: Contains(c, None) \wedge Ingredient(i),$
$\quad EFF: Contains(c, i) \wedge \neg Contains(c, None) \wedge \neg Ingredient(i))$
$Action(Empty(c, i),$
$\quad PRE: Contains(c, i) \wedge (i \neq None),$
$\quad EFF: Contains(c, None) \wedge \neg Contains(c, i))$
$Action(MakeCoffee(),$
$\quad PRE: Contains(A, Water) \wedge Contains(B, GroundCoffee) \wedge Contains(C, None),$
$\quad EFF: Contains(A, None) \wedge Contains(B, WetGroundCoffee) \wedge Contains(C, Coffee) \wedge$
$\quad \neg Contains(A, Water) \wedge \neg Contains(B, GroundCoffee) \wedge \neg Contains(C, None))$

PDDL encoding

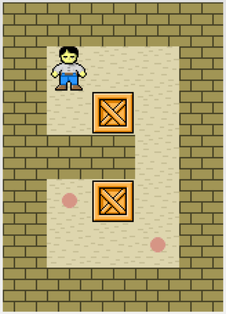


PDDL encoding



$$\text{Initial: } \mathit{Player}(1,5) \wedge \mathit{Box}(2,4) \wedge \mathit{Box}(2,2)$$
$$\wedge \bigwedge_{x=0}^{4} [\mathit{Wall}(x,0) \wedge \mathit{Wall}(x,6)]$$
$$\wedge \bigwedge_{y=0}^{6} [\mathit{Wall}(0,y) \wedge \mathit{Wall}(4,y)]$$
$$\wedge\, \mathit{Wall}(1,3) \wedge \mathit{Wall}(2,3)$$

PDDL encoding



Initial: $Player(1,5) \wedge Box(2,4) \wedge Box(2,2)$

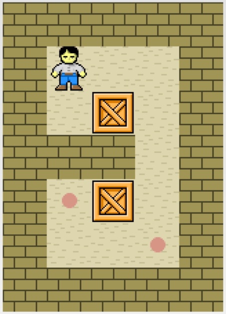
$$\wedge \bigwedge_{x=0}^{4} [Wall(x,0) \wedge Wall(x,6)]$$

$$\wedge \bigwedge_{y=0}^{6} [Wall(0,y) \wedge Wall(4,y)]$$

$$\wedge\ Wall(1,3) \wedge Wall(2,3)$$

Goal: $Box(1,2) \wedge Box(3,1)$

PDDL encoding

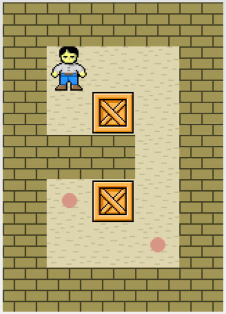


$$\text{Initial: } \mathit{Player}(1,5) \wedge \mathit{Box}(2,4) \wedge \mathit{Box}(2,2)$$
$$\wedge \bigwedge_{x=0}^{4} [\mathit{Wall}(x,0) \wedge \mathit{Wall}(x,6)]$$
$$\wedge \bigwedge_{y=0}^{6} [\mathit{Wall}(0,y) \wedge \mathit{Wall}(4,y)]$$
$$\wedge\, \mathit{Wall}(1,3) \wedge \mathit{Wall}(2,3)$$

$$\text{Goal: } \mathit{Box}(1,2) \wedge \mathit{Box}(3,1)$$

$$\begin{aligned}&\mathit{Action}(\mathit{MoveUp}(x,y),\\ &\qquad \mathit{PRE}: \mathit{Player}(x,y) \wedge \neg \mathit{Wall}(x,y+1) \wedge \neg \mathit{Box}(x,y+1),\\ &\qquad \mathit{EFF}: \mathit{Player}(x,y+1) \wedge \neg \mathit{Player}(x,y))\end{aligned}$$

PDDL encoding



$$\text{Initial: } \mathit{Player}(1,5) \wedge \mathit{Box}(2,4) \wedge \mathit{Box}(2,2)$$
$$\wedge \bigwedge_{x=0}^{4} [\mathit{Wall}(x,0) \wedge \mathit{Wall}(x,6)]$$
$$\wedge \bigwedge_{y=0}^{6} [\mathit{Wall}(0,y) \wedge \mathit{Wall}(4,y)]$$
$$\wedge\, \mathit{Wall}(1,3) \wedge \mathit{Wall}(2,3)$$

$$\text{Goal: } \mathit{Box}(1,2) \wedge \mathit{Box}(3,1)$$

$\mathit{Action}(\mathit{MoveUp}(x,y),$

$\quad \mathit{PRE} : \mathit{Player}(x,y) \wedge \neg \mathit{Wall}(x,y+1) \wedge \neg \mathit{Box}(x,y+1),$

$\quad \mathit{EFF} : \mathit{Player}(x,y+1) \wedge \neg \mathit{Player}(x,y))$

$\mathit{Action}(\mathit{PushUp}(x,y),$

$\quad \mathit{PRE} : \mathit{Player}(x,y) \wedge \mathit{Box}(x,y+1) \wedge \neg \mathit{Wall}(x,y+2) \wedge \neg \mathit{Box}(x,y+2),$

$\quad \mathit{EFF} : \mathit{Player}(x,y+1) \wedge \neg \mathit{Player}(x,y) \wedge \mathit{Box}(x,y+2) \wedge \neg \mathit{Box}(x,y+1))$

Forward vs. backward planning

- **Forward**: Start from the initial state, try unifying action schemas with the current state
  - Reasoning on states
  - Very large branching factor $\rightarrow$ heuristic are very important
- **Backward**: Start from the goal, compute sub-goals g' from action a

$$Pos(g') = (Pos(g) - Add(a)) \cup Pos(Precond(a))$$
$$Neg(g') = (Neg(g) - Del(a)) \cup Neg(Precond(a))$$

  - Reasoning on **set of** states
  - Harder to devise effective heuristics

Forward vs. backward planning

$$\begin{aligned}\text{Initial: } & \mathit{Contains}(A, \mathit{None}) \wedge \mathit{Contains}(B, \mathit{None}) \\ & \wedge \mathit{Contains}(C, \mathit{None}) \wedge \mathit{Ingredient}(\mathit{Water}) \\ & \wedge \mathit{Ingredient}(\mathit{GroundCoffee})\end{aligned}$$

$$\text{Goal: } \mathit{Contains}(C, \mathit{Coffee})$$

$\mathit{Action}(\mathit{Fill}(c, i),$

$\quad \mathit{PRE} : \mathit{Contains}(c, \mathit{None}) \wedge \mathit{Ingredient}(i),$

$\quad \mathit{EFF} : \mathit{Contains}(c, i) \wedge \neg\mathit{Contains}(c, \mathit{None}) \wedge \neg\mathit{Ingredient}(i))$

$\mathit{Action}(\mathit{Empty}(c, i),$

$\quad \mathit{PRE} : \mathit{Contains}(c, i) \wedge (i \neq \mathit{None}),$

$\quad \mathit{EFF} : \mathit{Contains}(c, \mathit{None}) \wedge \neg\mathit{Contains}(c, i))$

$\mathit{Action}(\mathit{MakeCoffee}(),$

$\quad \mathit{PRE} : \mathit{Contains}(A, \mathit{Water}) \wedge \mathit{Contains}(B, \mathit{GroundCoffee}) \wedge \mathit{Contains}(C, \mathit{None}),$

$\quad \mathit{EFF} : \mathit{Contains}(A, \mathit{None}) \wedge \mathit{Contains}(B, \mathit{WetGroundCoffee}) \wedge \mathit{Contains}(C, \mathit{Coffee}) \wedge$

$\quad \neg\mathit{Contains}(A, \mathit{Water}) \wedge \neg\mathit{Contains}(B, \mathit{GroundCoffee}) \wedge \neg\mathit{Contains}(C, \mathit{None})$

Planning Graph

- Interleaves two kind of layers:
  - Sets of states S_i represented by ground literals
  - A_i: Ground actions applicable at S_i
- Build forward from the initial state until convergence
- Contains **persistence**/**maintenance**/**no-ops** actions
- **Mutex** links among nodes of the same layer

Planning Graph

Create S_0 by adding the literals in the initial state

Repeat for $i = 0, 1, 2, \ldots$

Create A_i

- Add ground actions whose PRECOND unify with non-mutex literals in S_i
- Add no-op actions for every literal in S_i
- Add mutex links between actions in A_i

Create S_{i+1}

- For every ground action in A_i create and connect literals of the EFFECT
- For no-op action in A_i, create and connect it to the same literal
- Add mutex links between literals in S_{i+1}

...until $S_k = S_{k+1}$

Planning Graph

Mutex actions:

- Inconsistent effects
- Interference (one action's EFFECT negates the other's PRECOND)
- Inconsistent preconditions

$\rightarrow$ Any pair that **can't execute in any order with the same result**

Mutex literals:

- One is the negation of the other
- All ways of obtaining them are pairwise mutex

Planning Graph exercise

$Action(WearSocks, PRE : \neg Shoes, EFF : Socks)$
$Action(WearShoes, PRE : \neg Shoes, EFF : Shoes)$

$Action(UnwearSocks, PRE : Socks \land \neg Shoes, EFF : \neg Socks)$
$Action(UnwearShoes, PRE : Shoes, EFF : \neg Shoes)$

$Initial : -$
$Goal : Socks \land Shoes$

(Longer) Planning Graph exercise

$Action(WearSocks, PRE: \neg Shoes \wedge \neg Socks, EFF: Socks)$
$Action(WearPants, PRE: \neg Shoes \wedge \neg Pants, EFF: Pants)$
$Action(WearShoes, PRE: \neg Shoes, EFF: Shoes)$

$Action(UnwearSocks, PRE: Socks \wedge \neg Shoes, EFF: \neg Socks)$
$Action(UnwearPants, PRE: Pants \wedge \neg Shoes, EFF: \neg Pants)$
$Action(UnwearShoes, PRE: Shoes, EFF: \neg Shoes)$

$Initial: -$
$Goal: Pants \wedge Socks \wedge Shoes$

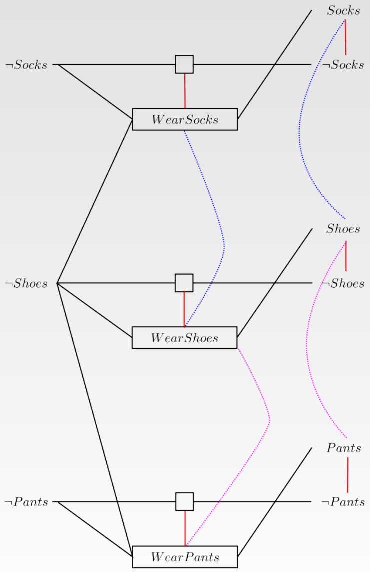
(Longer) Planning Graph exercise

$\neg Socks$

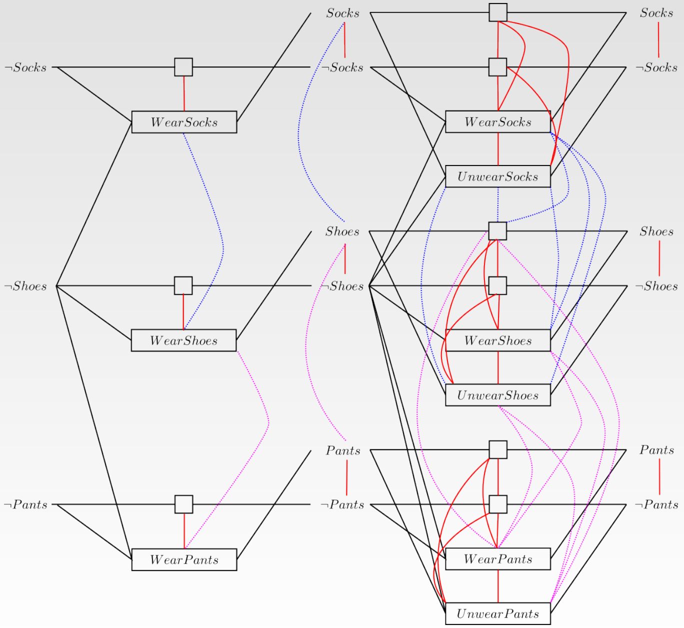
$\neg Shoes$

$\neg Pants$

(Longer) Planning Graph exercise



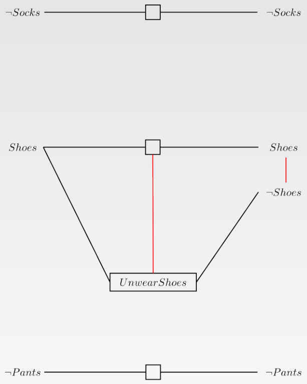
(Longer) Planning Graph exercise



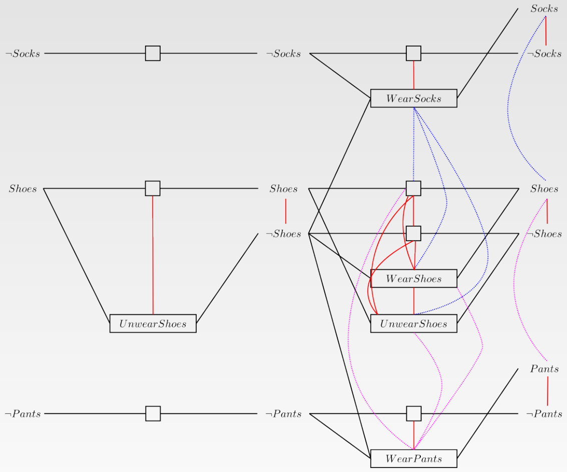
(Even longer) Planning Graph exercise

As before, but we start with *Shoes*

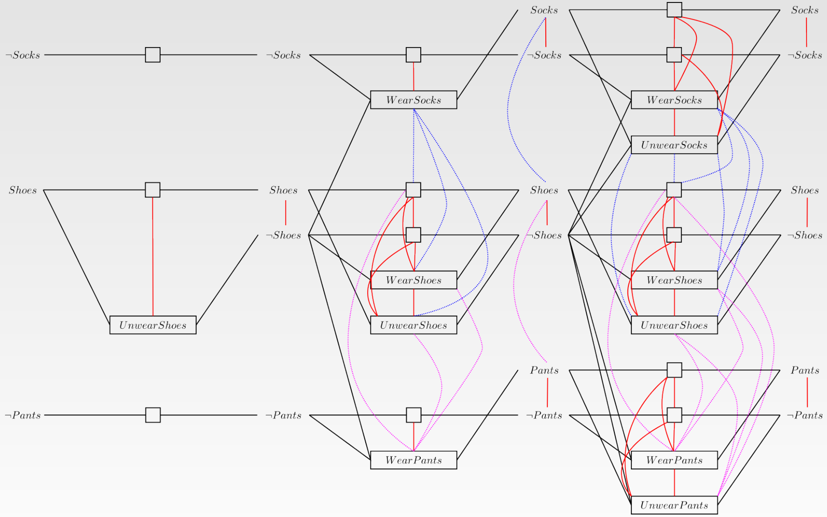
(Even longer) Planning Graph exercise



(Even longer) Planning Graph exercise



(Even longer) Planning Graph exercise



More exercises

- Use PDDL to model a variant of the moka exercise where A can contain any liquid and B any powder. Goal: make the Undergrad Coffee (coffee made with coffee in place of water).
- Hanoi tower
- ...