Notes on Planning

Linear Planning as Search

- To define a search problem, we need to define:

Linear Planning as Search

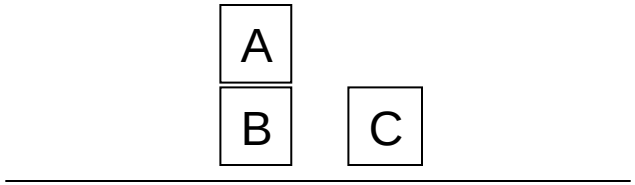
- To define a search problem, we need to define:
  - States.
  - State successors (legal “moves” for each state).
  - An initial state.
  - A goal.
  - A test for the goal.
  - (Optionally) a cost for each moves.

Defining States in Planning

- A state is:
  - A knowledge base, in some language.
    - Can be propositional, first-order, STRIPS, or anything else.
  - The knowledge base describes what is known in that particular state.
- Example: block world.
  - A state is a knowledge base.
  - Choices of language: STRIPS, first-order.

Blocks World - STRIPS

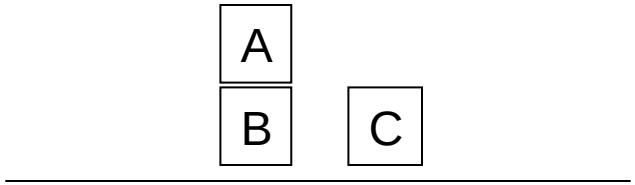
- Suppose we have:



- What do we need to include in the KB to represent that?

Blocks World - STRIPS

- Suppose we have:



- What do we need to include in the KB to represent that?

  (on A B)

  (on-table B)

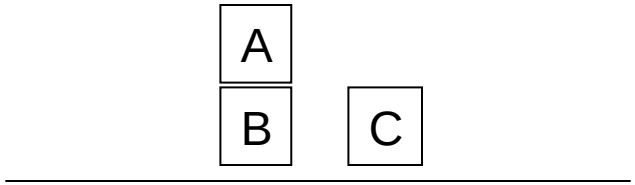
  (on-table C)

  (clear A)

  (clear C)

Blocks World - STRIPS

- Suppose we have:



- What do we need to include in the KB to represent that?

  (on A B)

  (on-table B)

  (on-table C)

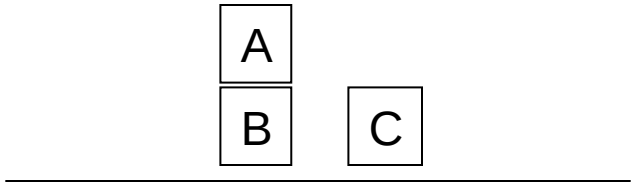
  (clear A)

  (clear C)

- How can we prove that B is not clear?

Blocks World - STRIPS

- Suppose we have:



- What do we need to include in the KB to represent that?

  (on A B)

  (on-table B)

  (on-table C)

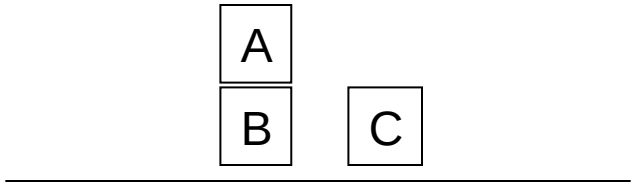
  (clear A)

  (clear C)

- How can we prove that B is not clear?
  - Using closed-world assumption.
    - The KB does not say that B is clear, => B is not clear.

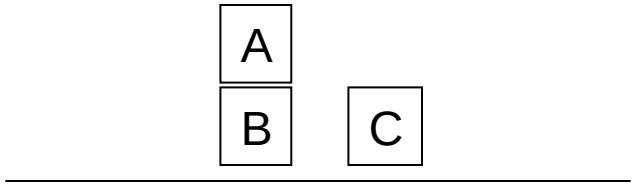
Blocks World – First Order

- Suppose we have:



- What do we need to include in the KB to represent that?

Blocks World – First Order



- Suppose we have:
- What do we need to include in the KB to represent that?

  on(A, B)

  on-table(B)

  on-table(C)

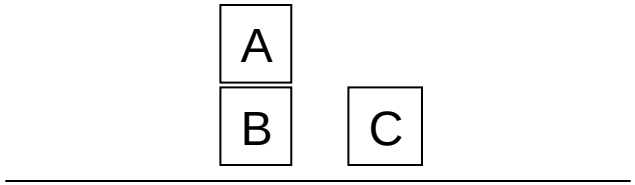
  clear(A)

  clear(C)

- How can we prove that B is not clear?
- We can’t. We need additional rules.

Blocks World – First Order

- Suppose we have:



- What do we need to include in the KB to represent that?

  on(A, B)

  on-table(B)

  on-table(C)

  clear(A)

  clear(C)

  on(x, y) => not(clear(y))

- In a first-order KB, inference is much more complicated than in a STRIPS KB.

Representing State Successors

- A state is a KB in some language.
- What are the successors of the state?

Representing State Successors

- A state is a KB in some language.
- What are the successors of the state?
  - The results of all possible actions that are legal on that state.

Goals, and Goal Tests

- A goal is:

Goals, and Goal Tests

- A goal is: a logical expression.
- How do we test if a state is a goal?

Goals, and Goal Tests

- A goal is: a logical expression.
- How do we test if a state is a goal?
  - We check if the state, which is a KB, entails the goal.
- How is that done in first-order logic?
- How is that done in STRIPS?

POP Planner

- It is still a search algorithm.
- However, there is an important difference from a linear planner: the meaning of a search state:
- Linear planner:
  - A search state is a possible state of the world.
  - The initial search state is the initial state of the world.
  - The goal state is a state that satisfies goal conditions.
- POP planner:
  - A search state is a partial plan.
  - The initial state is the empty plan, with specified initial conditions and goal conditions.
  - The goal state is a complete plan, with no open preconditions.

Successors in POP Planning

- In POP, a search state is a partial plan.
- A successor of a search state can be obtained by doing all of the above:
  - Adding an action to satisfy an open precondition.
  - Adding appropriate ordering constraints (links) between the action, its preconditions, and the open precondition(s) that it satisfies.
    - If multiple open preconditions are satisfied, there exist multiple ways for adding appropriate ordering constraints.
    - In that case, multiple successor nodes must be created.
  - Adding appropriate ordering constraints to handle clobbering.

Heuristics in POP Planning

- Assume that the cost of the plan is the number of actions in the plan.
- Heuristic 1: number of open preconditions.
  - Is this admissible?

Heuristics in POP Planning

- Assume that the cost of the plan is the number of actions in the plan.
- Heuristic 1: number of open preconditions.
  - Is this admissible? Not if a single action can satisfy multiple open preconditions.

Heuristics in POP Planning

- Assume that the cost of the plan is the number of actions in the plan.
- Heuristic 1: number of open preconditions.
  - Is this admissible? Not if a single action can satisfy multiple open preconditions.
- Heuristic 2: smallest number of actions needed to complete the plan.
  - Is this admissible?

Heuristics in POP Planning

- Assume that the cost of the plan is the number of actions in the plan.
- Heuristic 1: number of open preconditions.
  - Is this admissible? Not if a single action can satisfy multiple open preconditions.
- Heuristic 2: smallest number of actions needed to complete the plan.
  - Is this admissible? Yes.