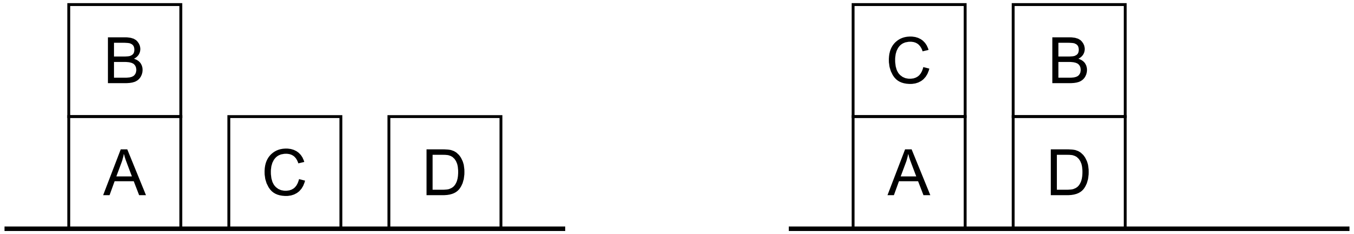
Planning

Overview

The blocks world



start

goal

How to achieve the goal from the start?

Overview

- Problem-solving is a search through a state space.

Overview

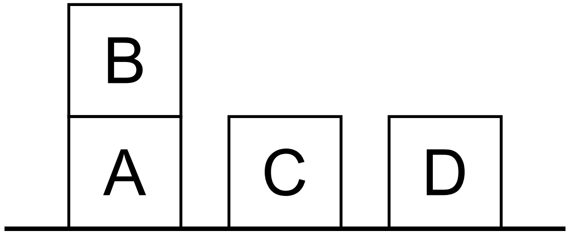
- Planning is the process of computing several steps of a problem-solving procedure before executing them.

- Planning = problem sloving in advance.

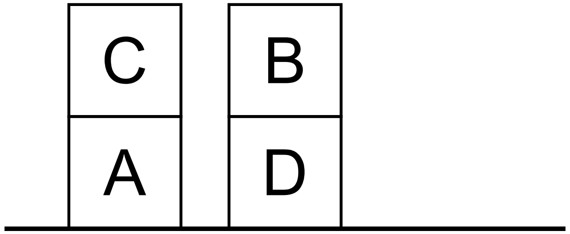
Overview

- Planning is important if solutions cannot be undone.
- If the universe is not predictable, then a plan can fail
  ⇒ dynamic plan revision.

The Blocks World



start



goal

Planning = generating a sequence of actions to achieve the goal from the start

The Blocks World

Actions:

- UNSTACK(A, B)
- STACK(A, B)
- PICKUP(A)
- PUTDOWN(A)

The Blocks World

Conditions and results:

- ON(A, B)
- ONTABLE(A)
- CLEAR(A)
- HOLDING(A)
- ARMEMPTY

The Blocks World

Specification of actions:

- PRECONDITION: list of predicates that must be true for an operator to be applied.
- ADD: list of new predicates that an operator causes to become true.
- DELETE: list of old predicates that an operator causes to become false.
- Predicates not in ADD nor DELETE are unaffacted.

The Blocks World

Specification of actions:

STACK(x, y):

P: CLEAR(y) $\wedge$ HOLDING(x)

D: CLEAR(y) $\wedge$ HOLDING(x)

A: ARMEMPTY $\wedge$ ON(x, y)

UNSTACK(x, y):

P: ON(x, y) $\wedge$ CLEAR(x) $\wedge$ ARMEMPTY

D: ON(x, y) $\wedge$ ARMEMPTY

A: HOLDING(x) $\wedge$ CLEAR(y)

The Blocks World

Specification of actions:

PICKUP(x):

P: CLEAR(x) ∧ ONTABLE(x) ∧ ARMEMPTY

D: ONTABLE(x) ∧ ARMEMPTY

A: HOLDING(x)

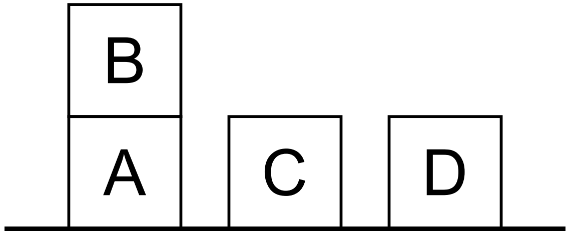
PUTDOWN(x):

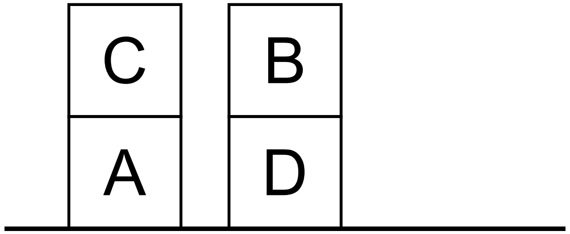
P: HOLDING(x)

D: HOLDING(x)

A: ONTABLE(x) ∧ ARMEMPTY

The Blocks World

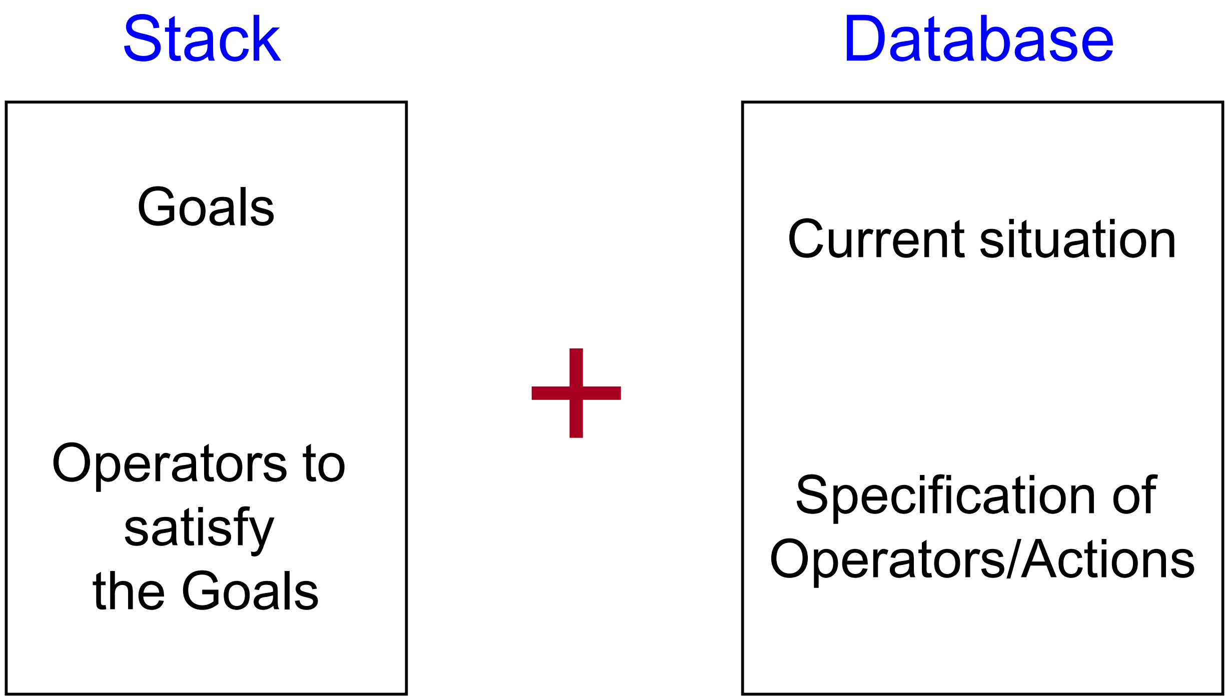




start: ON(B, A) ∧
ONTABLE(A) ∧
ONTABLE(C) ∧
ONTABLE(D) ∧
ARMEMPTY

goal: ON(C, A) ∧
ON(B, D) ∧
ONTABLE(A) ∧
ONTABLE(D) ∧

Goal Stack Planning



Stack

Goals

Operators to satisfy the Goals

+

Database

Current situation

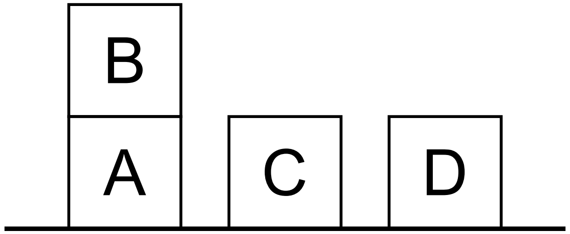
Specification of Operators/Actions

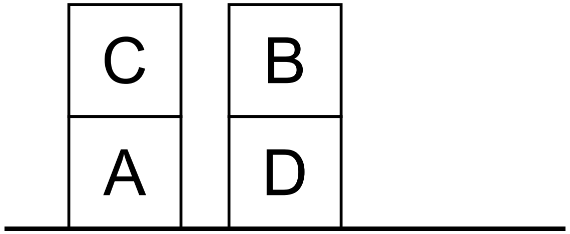
Goal Stack Planning

Push the original goal to the stack. Repeat until the stack is empty:

- If stack top is a compound goal, push its unsatisfied subgoals to the stack.
- If stack top is a single unsatisfied goal, replace it by an operator that makes it satisfied and push the operator's precondition to the stack.
- If stack top is an operator, pop it from the stack, execute it and change the database by the operation's affects.
- If stack top is a satisfied goal, pop it from the stack.

Goal Stack Planning





start: ON(B, A) ∧
ONTABLE(A) ∧
ONTABLE(C) ∧
ONTABLE(D) ∧
ARMEMPTY

goal: ON(C, A) ∧
ON(B, D) ∧
ONTABLE(A) ∧
ONTABLE(D) ∧

Goal Stack Planning

Stack	Database
ON(C, A)	ON(B, A)
ON(B, D)	OTACD
ON(C, A) ∧ ON(B, D) ∧ OTAD	ARMEMPTY

⬇

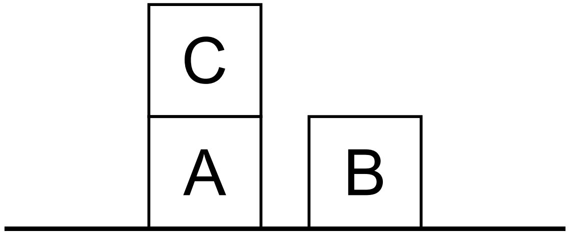
Stack	Database
CLEAR(A)	ON(B, A)
HOLDING(C)	OTACD
CLEAR(A) ∧ HOLDING(C)	ARMEMPTY
STACK(C, A)	
ON(B, D) ∧	
ON(C, A) ∧ ON(B, D) ∧ OTAD	

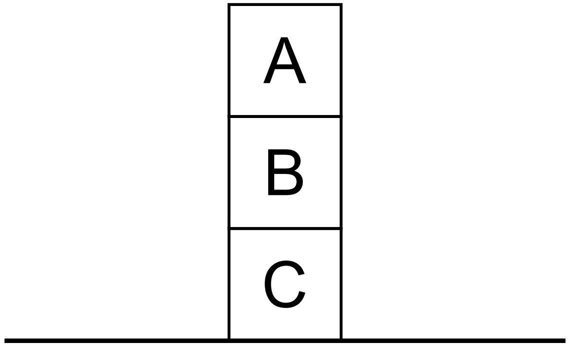
Goal Stack Planning

Plan

1. UNSTACK(B, A)
2. STACK(B, D)
3. PICKUP(C)
4. STACK(C, A)

Goal Stack Planning





start: ON(C, A) ∧
ONTABLE(A) ∧
ONTABLE(B) ∧
ARMEMPTY

goal: ON(A, B) ∧
ON(B, C)

Sussman Anomaly (1975)

Goal Stack Planning

Plan

1. UNSTACK(C, A)
2. PUTDOWN(C)
3. PICKUP(A)
4. STACK(A, B)
5. UNSTACK(A, B)
6. PUTDOWN(A)
7. PICKUP(B)
8. STACK(B, C)
9. PICKUP(A)
10. STACK(A, B)

Goal Stack Planning

Plan

1. UNSTACK(C, A)
2. PUTDOWN(C)
3. PICKUP(A)
4. STACK(A, B)
5. UNSTACK(A, B)
6. PUTDOWN(A)
7. PICKUP(B)
8. STACK(B, C)
9. PICKUP(A)
10. STACK(A, B)

Questions

- Why stacks used?
- Why a compound goal retained in the stack with its subgoals?.
- Does the order of subgoals in the stack matter?.