Planning

Chapter 10

The blocks world



How to achieve the goal from the start?

• Problem-solving is searching and moving through a state space.

- Problem-solving is searching and moving through a state space.
- Planning is searching for successful paths through a state space.

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

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



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

Actions:

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

Conditions and results:

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

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.

Specification of actions:

STACK(x, y):

P: CLEAR(y) \land HOLDING(x)

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

A: ARMEMPTY \land ON(x, y)

UNSTACK(x, y):

P: ON(x, y) \land CLEAR(x) \land ARMEMPTY

D: ON(x, y) \land ARMEMPTY

A: HOLDING(x) \land CLEAR(y)

Specification of actions:

PICKUP(x):

P: CLEAR(x) \land ONTABLE(x) \land ARMEMPTY

D: ONTABLE(x) \land ARMEMPTY

A: HOLDING(x)

PUTDOWN(x):

P: HOLDING(x)

D: HOLDING(x)

A: ONTABLE(x) \land ARMEMPTY



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



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



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.



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



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



Plan

UNSTACK(B, A)
STACK(B, D)
PICKUP(C)
STACK(C, A)



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

Sussman Anomaly (1975)

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)

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?

Linear vs. Non-Linear Planning

- Goal Stack planning is linear: satisfies subgoals sequentially, one after another.
- Non-linear planning: consider interaction among subgoals.

Homework

Excercises:

1-4, Chapter 13, Rich&Knight AI Textbook Chapter 7 of the Vietnamese Textbook