Planning

Chapter 10
Overview

The blocks world

How to achieve the goal from the start?
Overview

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

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

- **Planning** is searching for *successful paths* through a state space.
Overview

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

- Planning = problem solving *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

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 unaffected.
The Blocks World

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)
The Blocks World

Specification of actions:

PICKUP(x):

\[ \text{P: } \text{CLEAR}(x) \land \text{ONTABLE}(x) \land \text{ARMEMPTY} \]

\[ \text{D: } \text{ONTABLE}(x) \land \text{ARMEMPTY} \]

\[ \text{A: } \text{HOLDING}(x) \]

PUTDOWN(x):

\[ \text{P: } \text{HOLDING}(x) \]

\[ \text{D: } \text{HOLDING}(x) \]

\[ \text{A: } \text{ONTABLE}(x) \land \text{ARMEMPTY} \]
The Blocks World

start: \( \text{ON}(B, A) \land \text{ONTABLE}(A) \land \text{ONTABLE}(C) \land \text{ONTABLE}(D) \land \text{ARMEMPTY} \)

goal: \( \text{ON}(C, A) \land \text{ON}(B, D) \land \text{ONTABLE}(A) \land \text{ONTABLE}(D) \land \text{ARMEMPTY} \)
Goal Stack Planning

Stack

Goals

Operators to satisfy the Goals

Database

Current situation

Specification of Operators/Actions

CuuDuongThanCong.com

https://fb.com/tailieudientucntt
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: $\text{ON}(B, A) \land \text{ONTABLE}(A) \land \text{ONTABLE}(C) \land \text{ONTABLE}(D) \land \text{ARMEMPTY}$

goal: $\text{ON}(C, A) \land \text{ON}(B, D) \land \text{ONTABLE}(A) \land \text{ONTABLE}(D) \land \text{ARMEMPTY}$
Goal Stack Planning

Stack

ON(C, A)
ON(B, D)
ON(C, A) \land ON(B, D) \land OTAD

CLEAR(A)
HOLDING(C)
CLEAR(A) \land HOLDING(C)
STACK(C, A)
ON(B, D) \land
ON(C, A) \land ON(B, D) \land OTAD

Database

ON(B, A)
OTACD
ARMEMPTY

ON(B, A)
OTACD
ARMEMPTY
Goal Stack Planning

Plan

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

start: \( \text{ON}(C, A) \land \text{ONTABLE}(A) \land \text{ONTABLE}(B) \land \text{ARMEMPTY} \)

goal: \( \text{ON}(A, B) \land \text{ON}(B, C) \)

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?
Linear vs. Non-Linear Planning

• Goal Stack planning is linear: satisfies subgoals sequentially, one after another.

• Non-linear planning: consider interaction among subgoals.
Homework

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