COMP 9414 Week 4 - Planning and Reasoning About Action

Author: Junji Duan

Last Update: 2025/10/12

1 W4: Planning

1.1 First-Order Logic

Def: First-Order Logic (FOL) is a formal language for stating facts about objects and their relationships, and for reasoning from those facts.

1.1.1 Syntax of First-Order Logic

Constant symbols: $a, b, \ldots$, Mary (objects)

Variables: $x, y, \ldots$

Function symbols: f, mother_of, sine, . . .

Predicate symbols: Mother, likes, . . .

Quantifiers: $\forall$ (universal); $\exists$ (existential)

1.1.2 Language of First-Order Logic

- Terms: constants, variables, functions applied to terms (refer to objects)
  - e.g. $a, f(a)$, mother_of (Mary), $\ldots$
- Atomic formulae: predicates applied to tuples of terms
  - e.g. likes(Mary, mother_of(Mary)), likes(x, a)
- Quantified formulae:
  - e.g. $\forall x$ likes $(x, a), \exists x$ likes (x, mother_of (y))
  - here the second occurrences of x are bound by the quantifier ($\forall$ in the first case, $\exists$ in the second) and y in the second formula is free
- Converting English into First-Order Logic
  - Everyone likes lying on the beach $-\forall x$ likes_lying_on_beach (x)
  - Someone likes Fido $-\exists x$ likes (x, Fido)
  - No one likes Fido $-\neg\exists x$ likes (x, Fido)(or $\forall x \neg$ likes (x, Fido))

1.1.3 Universal & Existential & Nested Quantifiers

- $\forall x\, P$ is (almost) equivalent to the conjunction of instantiations of P.
- $\exists x\, P$ is (almost) equivalent to the disjunction of instantiations of P.
- For nested quantifiers, the order of quantification is very important.

1.1.4 Defining Semantic Properties

兄弟一定是手足
$\forall x \forall y(\text{brother}(x, y) \rightarrow \text{sibling}(x, y))$
任何 x 是 y 的 brother，就必然是 y 的 sibling。

"手足"是对称的
$\forall x \forall y(\text{sibling}(x, y) \leftrightarrow \text{sibling}(y, x))$
如果 x 和 y 是手足，那么顺序互换也成立。

母亲＝女性且为家长（双向定义）
$\forall x \forall y(\text{mother}(x, y) \leftrightarrow(\text{female}(x) \wedge \text{parent}(x, y)))$

1.1.5 Scope Ambiguity & Scope of Quantifiers

量词作用域不同导致的 Scope Ambiguity(作用域歧义)

$\forall x(\text{student}(x) \rightarrow \exists y(\text{exam}(y) \wedge \text{took}(x, y)))$ 每个学生都参加了某场考试 (可以各自不同).

$\exists y(\text{exam}(y) \wedge \forall x(\text{student}(x) \rightarrow \text{took}(x, y)))$
有一场考试，所有学生都参加了同一场.

某个 Scope of Quantifiers(量词的作用域) 就是它直接支配的子公式, 某个变量名 x 被最靠近它的量词（x / x）"认领"了, 如果有好几层对同名变量的量词，离这个变量最近（嵌套最深）的那一个算数；它会遮蔽外层同名量词.

1.2 Reasoning About Action

- Semantics: Divide the world into a sequence of (notional) time points
  - Situation is a (complete) state of world at a time point

- Action is a transition between situations
- Nothing (of relevance) happens between situations

- Planner: Maintain an incomplete description of situations
  - Confusingly, also called a state of the world
  - Search for path from initial state to a goal state
  - State transitions correspond to actions
  - Major problem is to specify actions
    * 难点: 如何“正确地”刻画动作

1.2.1 Problems in Reasoning About Action

- Frame Problem
  - How to characterize what in the state does not change by performing an action
    * Problem is there are a lot of such facts
    * Both ”epistemological” and ”computational” problem

即: 当执行一个动作时，世界中绝大多数事实其实不变。如果用逻辑规则把“未改变的一切”都列出来，几乎无穷无尽，知识表示会臃肿且易错。

- Ramification Problem
  - What are the direct and indirect effects of performing an action?
    * Problem is that indirect effects depend on initial situation

即: 动作除了直接效果，还会通过因果/约束产生间接效果；这些间接效果常常取决于初始情景。间接效果可能很多、层层传递，不可能把所有后果都当作“直接效果”硬写进 Add/Delete。

- Qualification Problem
  - What preconditions are required in a specification of an action?
    * Problem is that qualifications depend on context

即: 一个动作在现实中常需要极多甚至不可穷尽的前提条件；我们不可能把所有“但”都写进动作前提。

1.3 STRIPS Planner

Action Description - name of action

- Action Description: move $(x, y, z)(x \neq y \neq z?)$

Preconditions - action can be performed in a situation only if precondition holds in situation prior to action being performed

- Preconditions: on (x, y), clear (x), clear (z)

Delete List - literals to be deleted from the state (description) after action is performed

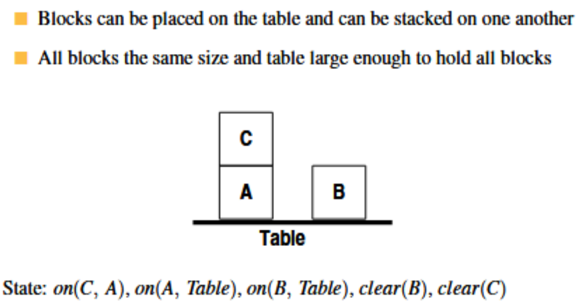
- Delete List: clear(z), on(x, y)

Add List - literals to be added to the state (description) after action is performed

- Add List: on(x, z), clear(y), clear(Table)
  - Add clear(Table) to ensure table is always clear

STRIPS Assumption - any literals in the state (description) not contained in the delete list remain the same after the action is performed (c.f. frame problem)

1.3.1 Blocks World



- Action Description: move $(x, y, z)(x \neq y \neq z$?)
- Preconditions: on (x, y), clear (x), clear (z)
- Delete List: clear (z), on (x, y)
- Add List: on(x, z), clear(y), clear(Table)
  - Add clear(Table) to ensure table is always clear

1.4 GraphPlan

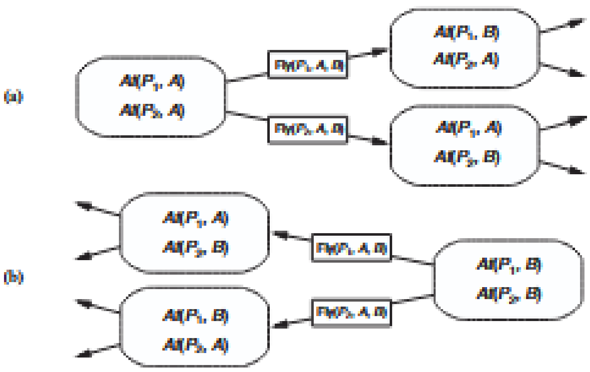
1.4.1 Planning

- Plan - sequence (or ordered set) of actions to achieve some goal
- Planner - problem solver that produces plans
- Goal - typically a conjunction of literals
- Initial State-typically a conjunction of literals
- Blocks World example for goal on $(B, C) \wedge on(C, \text{ Table })$
  - $\text{move}(C, A, \text{ Table }), \text{move}(B, \text{ Table}, C)$

1.4.2 Simple Planning Algorithms

3 Algorithms: Forward search, goal regression and non-linear.

- Forward search and goal regression.
  - Forward search: 从初始状态出发扩展动作 → 状态；优点直观，缺点：状态空间巨大。
  - Goal regression: 从目标往回“消解”需要的前置；优点能聚焦相关性，缺点：设计难、不总是奏效。



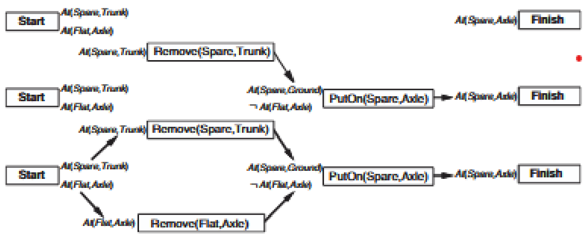
Problem with forward search is state space can be very large.

Problem with regression is that it is hard and doesn’t always work.

1.4.3 Nonlinear Planning

先做目标回归得到“部分序”计划，逐步修复冲突(threats/flaws)；最少承诺原则（least commitment)：只在必要时确定顺序，执行时按任何允许顺序进行。

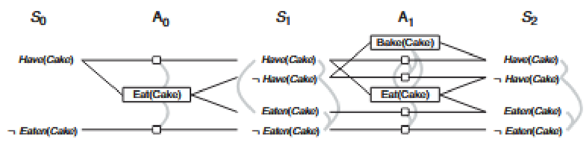
Start with goal regression and try to fix “flaws” in the plan.



Least commitment: execution can be in any permissible order.

1.4.4 Forward Search with Plan Graphs

- Only consider ”propositional” plans
- S_i contains all literals that could hold at time i
- A_i contains all actions that could have preconditions satisfied at time i
- Actions linked to preconditions
- Literals that persist from time i to time $i+1$ linked via actions
- Mutual exclusion (mutex) links between actions/literals at same time



1.4.5 Mutual Exclusion

Actions

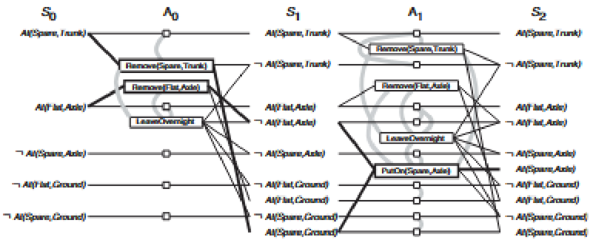
- Inconsistent effects: One action negates an effect of the other
- Interference: Effect of one action is the negation of a precondition of the other
- Competing needs: Precondition of one action is mutually exclusive with a precondition of the other

Literals

- One literal is the negation of the other
- Inconsistent support: Each possible pair of actions that could achieve the two literals is mutually exclusive

1.4.6 GraphPlan Expansion Step

- Add actions to A_i whose preconditions are at S_i
- Add "persistence actions" to A_i for literals from S_i
- Add mutex links to A_i for actions that cannot occur together
- Add effects of all actions in A_i to S_{i+1}
- Add literals to S_{i+1} for persistence actions from A_i
- Add mutex links to S_{i+1} for literals that cannot occur together



1.5 Planning as Constraint Satisfaction

CSP for each planning horizon k (vary k as needed)

- Variables
  - Create a variable for each literal and time $0, \cdots, k$
  - Create a variable for each action and time $0, \cdots, k-1$
- Constraints
  - State constraints: literals at time t
  - Precondition constraints: actions and states at time t
  - Effect constraints: actions at time t, literals at times t and $t+1$
  - Action constraints: actions at time t (mutual exclusion)
  - Initial state constraints: literals at time 0
  - Goal constraints: literals at time k