

# COMP9414: Artificial Intelligence

## Lecture 3b: Planning

Wayne Wobcke

e-mail:w.wobcke@unsw.edu.au

### This Lecture

- Reasoning About Action
- STRIPS Planner
- GraphPlan
- Planning as Constraint Satisfaction

### Planning Agent

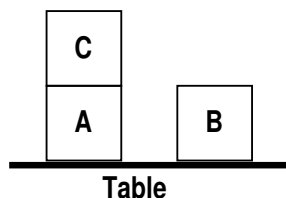
- Environment changes due to the performance of actions
- Planning scenario
  - ▶ Agent can control its environment
  - ▶ Only single agent in the environment (no interference)
  - ▶ Only changes due to agent executing actions (no evolution)
- More complex examples
  - ▶ Robocup dog
  - ▶ Delivery robot
  - ▶ Self-driving car

### 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

## The Blocks World

- Blocks can be placed on the table and can be stacked on one another
- All blocks the same size and table large enough to hold all blocks



State:  $on(C, A)$ ,  $on(A, Table)$ ,  $on(B, Table)$   $clear(B)$ ,  $clear(C)$

## Blocks World Actions (STRIPS)

- **Action Description:**  $move(x, y, z)$
- **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

## Specifying Actions (STRIPS)

- **Action Description** — name of action
- **Preconditions** — action can be performed in a situation only if precondition holds in situation prior to action being performed
- **Delete List** — literals to be deleted from the state (description) after action is performed
- **Add List** — literals to be added to the state (description) after action is performed
- **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)

Assumes actions are executed perfectly (reasonable for planning?)

## Problems in Reasoning About Action

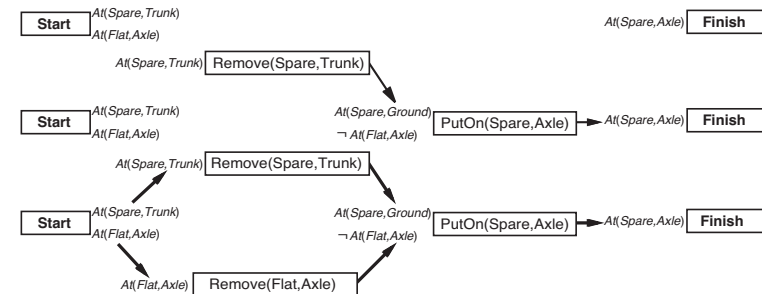
- **Frame Problem**
  - ▶ How to characterize literals of the state that are **not** changed by performing an action?
    - Problem is there are a lot of such literals
    - Both “epistemological” and “computational” problem
- **Ramification Problem**
  - ▶ What are the direct and **indirect** effects of performing an action?
    - Problem is that indirect effects depend on initial situation
- **Qualification Problem**
  - ▶ What **preconditions** are required in a specification of an action?
    - Problem is that qualifications depend on context

## 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  $on(B, C) \wedge on(C, Table)$ 
  - ▶  $move(C, A, Table), move(B, Table, C)$

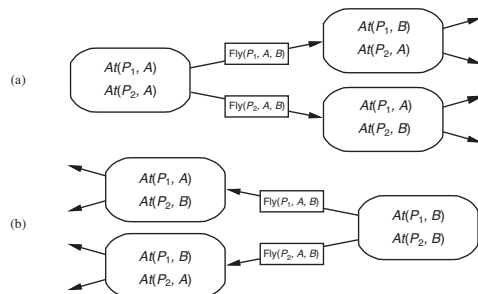
## Nonlinear Planning

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



## Simple Planning Algorithms

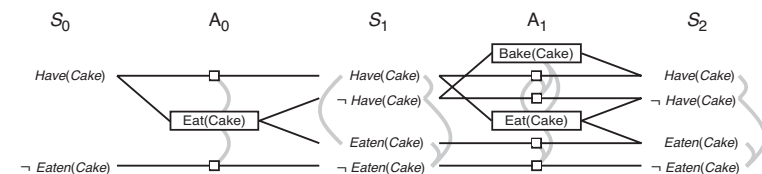
- Forward search and goal regression



- Problem with forward search is state space can be very large
- Problem with regression is that it is hard and doesn't always work

## 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



## 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

## GraphPlan Algorithm

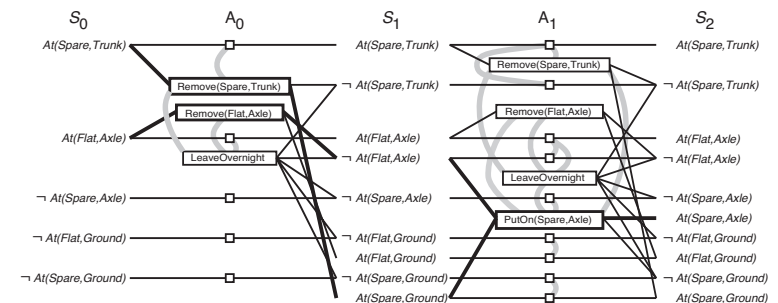
- Graph = Initial plan graph with initial state  $S_0$
- nogoods = empty set
- For  $t = 0, \dots$ 
  - ▶ If all goals are non-mutex in  $S_t$ 
    - Extract solution from graph
      - Graph as CSP with variables T/F for when action in the plan
      - Or heuristically guided regression from  $S_t$  to  $S_0$
    - If valid solution, return solution
  - ▶ If graph and nogoods didn't change then return failure
  - ▶ Expand graph to next level

## 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  $A_i$  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

## GraphPlan Example

- After expansion to Level 2



## Planning as Constraint Satisfaction

---

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

### ■ Variables

- ▶ Create a variable for each feature and time  $0, \dots, k$
- ▶ Create a variable for each action and time  $0, \dots, k-1$

### ■ 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  $k$

## Conclusion

---

- Reasoning about action interesting from philosophical point of view
- Recent advances in planning give great improvements in efficiency
- Planning makes use of CSP framework with heuristics
- Multi-agent systems, dynamic worlds much more complex