Planning and Acting

CHAPTER 13

Outline

◊ The real world
◊ Conditional planning
◊ Monitoring and replanning
The real world

Things go wrong

Incomplete information
- Unknown preconditions, e.g., Intact(Spare)?
- Disjunctive effects, e.g., Inflate(x) causes Inflate(x) ∨ SlowHiss(x) ∨ Burst(x) ∨ BrokenPump ∨ ...

Incorrect information
- Current state incorrect, e.g., spare NOT intact
- Missing/incorrect postconditions in operators

Qualification problem:
- can never finish listing all the required preconditions and possible conditional outcomes of actions
Conditional planning

- Plan to obtain information (observation actions)
- Subplan for each contingency, e.g.,
  - [Check(Tire1). If(Intact(Tire1), [Inflate(Tire1)]. [CallAAA])] 

Expensive because it plans for many unlikely cases

Monitoring/Replanning

- Assume normal states, outcomes
- Check progress during execution, replan if necessary

Unanticipated outcomes may lead to failure (e.g., no AAA card)

In general, some monitoring is unavoidable

Section 13.1 – Conditional Planning
Extending Plan Language

- Parameterized plans
  - Exact actions not known until execution
- Ex: Paint chair and table same color
  \[[\text{SenseColor(Table)}, \text{KnowsWhat}(\text{Color(Table,c)}), \text{GetPaint}(c), \text{Paint}(Chair,c)]\]
- Can achieve goal by painting both same color
  - Wastes time
- Use maintenance goal:
  - Color(Chair,c) A Color(Table,C) A (Maintain(Color(Table,x)))
    - No action can be inserted that has causal effect of changing color of the table

13.2 – Execution Monitoring
Situated Planning Agent

• Integrate planning with execution
• Activities include:
  – Execute steps that are ready
  – Refine plan to resolve deficiencies
  – Refine plan to handle new information
  – Refine plan to handle changes in environment

Example: Blocks World

(a) Start state
(b) Another agent put D on B
(c) Move(C,D) failed, C is on A instead
(d) Goal State
Plan as it would be developed by POP

After another agent moves D onto B
- unsupported links protecting Clear(B) and On(D,G) are dropped

- Link supplied by Move(D,B) replaced by link from Start
- Redundant step Move(D,B) dropped
• Move(C,D) executed and removed from plan
• But C dropped on A instead of D
• Goal precondition On(C,D) still open

Open precondition resolved by putting Move(C,D) back in
• Move(C,D) executed successfully
• Dropped from plan
• All preconditions resolved
  • causal links from Start to Finish
• Plan now complete

Preconditions for remaining plan
Replanning

Simplest: on failure, replan from scratch
Better: plan to get back on track by reconnecting to best continuation
Generates “loop until done” behavior with no explicit loop

<table>
<thead>
<tr>
<th>PRECONDITIONS</th>
<th>FAILURE RESPONSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>R/A</td>
</tr>
<tr>
<td>Have(Rod)</td>
<td>Fetch more rod</td>
</tr>
<tr>
<td>Cure(Rod,Red)</td>
<td>Repaint</td>
</tr>
</tbody>
</table>