Planning and Acting
Outline

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

START

~Flat(Spare) Intact(Spare) Off(Spare)
On(Tire1) Flat(Tire1)

FINISH

On(x) ~Flat(x)

On(x)
Remove(x)

Off(x) ClearHub

Off(x) ClearHub
Puton(x)

On(x) ~ClearHub

Intact(x) Flat(x)

~Flat(x)

Inflate(x)
Things go wrong

Incomplete information

Unknown preconditions, e.g., $\text{Intact(Spare)}$?
Disjunctive effects, e.g., $\text{Inflate}(x)$ causes
\[ \text{Inflated}(x) \lor \text{SlowHiss}(x) \lor \text{Burst}(x) \lor \text{BrokenPump} \lor \ldots \]

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
Solutions

Conditional planning

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

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
Conditional planning

[... , If(p, [then plan], [else plan]), ...]

Execution: check $p$ against current KB, execute "then" or "else"

Conditional planning: just like POP except
  if an open condition can be established by observation action
    add the action to the plan
  complete plan for each possible observation outcome
  insert conditional step with these subplans

CheckTire(x)

KnowsIf(Intact(x))
Conditional planning example

Start
On(Tire1)
Flat(Tire1)
Inflated(Spare)

Finish
On(x)
Inflated(x)
(True)
Conditional planning example
Conditional planning example

Start
- On(Tire1)
- Flat(Tire1)
- Inflated(Spare)

Check(Tire1)
- Intact(Tire1)

Inflate(Tire1)
- Intact(Tire1)
- (Intact(Tire1))

Finish
- On(Tire1)
- Inflated(Tire1)
- (Intact(Tire1))
Conditional planning example

Start

On(Tire1)
Flat(Tire1)
Intact(Tire1)
Inflated(Spare)

Check(Tire1)

Intact(Tire1)

Flat(Tire1)

Inflate(Tire1)

(Tire1)

On(Tire1)

Inflated(Tire1)

Finish

On( x )

Inflated( x )

(~Intact(Tire1))
Conditional planning example

Start

On(Tire1)
Inflated(Spare)
Flat(Tire1)
Intact(Tire1)
Check(Tire1)

Finish

On(Tire1)
Inflated(Tire1)
Intact(Tire1)
Inflate(Tire1)

Spare
Finish

Start

On(Tire1)
Inflated(Spare)
Flat(Tire1)
Intact(Tire1)
Check(Tire1)

Finish

On(Tire1)
Inflated(Tire1)
Intact(Tire1)
Inflate(Tire1)

Spare
Finish

Start

On(Tire1)
Inflated(Spare)
Flat(Tire1)
Intact(Tire1)
Check(Tire1)

Finish

On(Tire1)
Inflated(Tire1)
Intact(Tire1)
Inflate(Tire1)

Spare
Finish

Start

On(Tire1)
Inflated(Spare)
Flat(Tire1)
Intact(Tire1)
Check(Tire1)

Finish

On(Tire1)
Inflated(Tire1)
Intact(Tire1)
Inflate(Tire1)

Spare
Finish

Start

On(Tire1)
Inflated(Spare)
Flat(Tire1)
Intact(Tire1)
Check(Tire1)

Finish

On(Tire1)
Inflated(Tire1)
Intact(Tire1)
Inflate(Tire1)

Spare
Finish
Conditional planning example
Monitoring

Execution monitoring
   "failure" = preconditions of remaining plan not met
   preconditions = causal links at current time

Action monitoring
   "failure" = preconditions of next action not met
   (or action itself fails, e.g., robot bump sensor)

In both cases, need to replan
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

### Failure Response

<table>
<thead>
<tr>
<th>PRECONDITIONS</th>
<th>FAILURE RESPONSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>N/A</td>
</tr>
<tr>
<td>Have(Red)</td>
<td>Fetch more red</td>
</tr>
<tr>
<td>Color(Chair,Red)</td>
<td>Repaint</td>
</tr>
</tbody>
</table>

### Preconditions

- Color(Chair,Blue)
- ~Have(Red)
- Get(Red)
- Have(Red)
- Paint(Red)
- Color(Chair,Red)