Contingency Planning and Online Replanning

CSE 4308/5360 – Artificial Intelligence I
University of Texas at Arlington

Planning in More Complicated Domains

- Our discussion of planning so far assumed a fully-observable, deterministic world.
  - The state of the world is fully known.
  - The effects of any action are known in advance.
  - Nothing ever goes wrong.
- Obviously, in real life these assumptions are often violated.
  - Some aspects of the current state may be unknown.
  - Some facts that we think are true may actually be false.
    - You think your car is in the parking lot, but it has been stolen :(
  - Some actions may not have the expected effect.

Three Planning Approaches

- Sensorless planning (also called conformant planning).
  - Handles domains where the state of the world is not fully known.
  - Comes up with plans that work in all possible cases.
- Contingent planning (also called conditional planning).
  - Handles domains where the effects of an action are not deterministic.
  - Approach: plan ahead for different possible results of each action.
- Execution monitoring and online replanning.
  - Handles domains where the effects of an action are not deterministic, or where, more generally, things in the world may unexpectedly change.
  - While executing the plan, before performing each action, monitor the environment.
  - If the enviroment is different than expected, replan.

Sensorless (Conformant) Planning

- Handles domains where the state of the world is not fully known.
- Comes up with plans that work in all possible cases.

Sensorless (Conformant) Planning

- Example:
  - You have a wall made of bricks.
  - You have a can of white paint.
  - Action: *Paint(brick)*, effect: *Color(brick, white)*.
  - Goal: every brick should be painted white.

Sensorless (Conformant) Planning

- Example:
  - You have a wall made of bricks.
  - You have a can of white paint.
  - Action: *Paint(brick)*, effect: *Color(brick, white)*.
  - Goal: every brick should be painted white.
- In a fully observable domain, you could:
  - Know the initial color of every brick.
  - Make a plan to paint all the bricks that are not white initially.
  - No need to paint bricks that are already white.

Sensorless (Conformant) Planning

- Example:
  - You have a wall made of bricks.
  - You have a can of white paint.
  - Action: *Paint(brick)*, effect: *Color(brick, white)*.
  - Goal: every brick should be painted white.
- Suppose the world is not fully observable.
  - We actually cannot observe the color of a brick.
- Suppose that the world is deterministic.
  - The effects of an action are known in advance.
- What plan would ensure achieving the goal?

Sensorless (Conformant) Planning

- Example:
  - You have a wall made of bricks.
  - You have a can of white paint.
  - Action: *Paint(brick)*, effect: *Color(brick, white)*.
  - Goal: every brick should be painted white.
- Suppose the world is not fully observable.
  - We actually cannot observe the color of a brick.
- Suppose that the world is deterministic.
  - The effects of an action are known in advance.
- What plan would ensure achieving the goal?
  - Paint all bricks, regardless of their initial color (which we don't know anyway).
  - It may be overkill, since some bricks may already be white, but it is the only plan that guarantees achieving the goal.

Sensorless (Conformant) Planning

- Limitations:

Sensorless (Conformant) Planning

- Limitations:
  - While there are a few domains simple enough to allow for sensorless planning:
  - Many real world domains are too complicated for this approach, and you can't come up with plans that work regardless of what the state of the world is.

Contingent Planning

- Also called conditional planning.
- Handles domains where the effects of an action are not deterministic.
- Approach: plan ahead for different possible results of each action.

Contingent Planning

- Example: this was the definition of actions for the blocks world:

  Action(Move(block, from, to),
  PRECOND: On(block, from) AND Clear(block) AND Clear(to)
  EFFECT: On(block, to) AND NOT(On(block, from)) AND
  Clear(from) AND NOT(Clear(to))

  Action(MoveToTable(block, from),
  PRECOND: On(block, from) AND Clear(block)
  EFFECT: On(block, Table) AND NOT(On(block, from)) AND Clear(from)

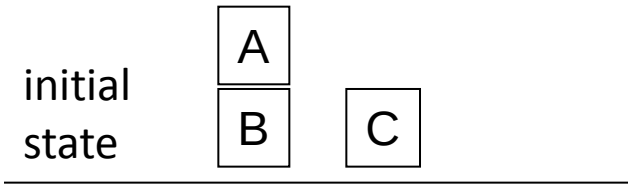
Contingent Planning

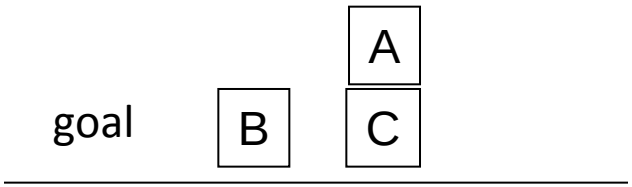
- Suppose now that we have a non-deterministic blocks-world.
- Moving a block may be successful or not.
  - If successful, the block ends up where we wanted.
  - If unsuccessful, the block falls on the table.
- To model this, we need to allow the effects of an action to include disjunctions:

Action(Move(block, from, to),
PRECOND: On(block, from) AND Clear(block) AND Clear(to)
EFFECT: NOT(On(block, from)) AND Clear(from)
AND ((On(block, to) AND NOT(Clear(to))) OR (On(block, table))))

Action(MoveToTable(block, from),
PRECOND: On(block, from) AND Clear(block)
EFFECT: On(block, Table) AND NOT(On(block, from)) AND Clear(from)

Contingent Planning





- We also need to allow plans to have if statements.
- Example:

move(A, B, C)

if *on(A, C)* **then** *done*

else:

???

Contingent Planning

- We also need to allow plans to have if statements.
- Example:

move(A, B, C)

if *on(A, C)* **then** *done*

else:

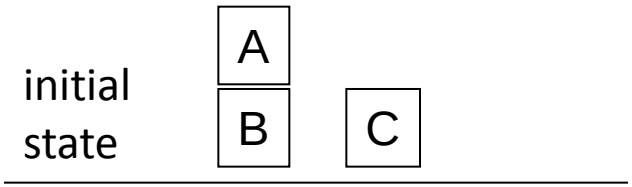
move(A, table, C)

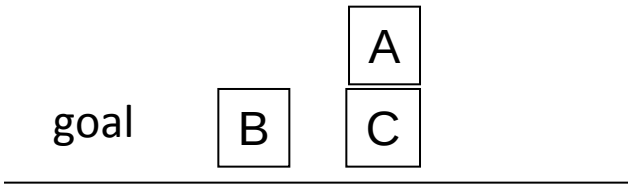
if *on(A, C)* **then** *done*

else:

???

Contingent Planning





- We also need to allow plans to have if statements.
- Example:

move(A, B, C)

if *on(A, C)* **then** *done*

else:

move(A, table, C)

if *on(A, C)* **then** *done*

else:

move(A, table, C)

if *on(A, C)* **then** *done …*

Contingent Planning

- Theoretically, the execution of such a plan may never terminate.
- In practice, one of the attempts should be successful, and the plan should eventually terminate.
- This behavior matches real-world plan execution:
  - Usually there is no 100% guarantee that a plan will work.
  - In practice, well-made plans tend to work.
- Contingent plans look like decision trees. At each node, we choose a subtree based on a condition.

Online Replanning

- Handles domains where:
  - The effects of an action are not deterministic.
  - More generally, things in the world may unexpectedly change.
- While executing the plan:
  - Before performing each action, monitor the environment.
  - If the enviroment is different than expected, replan.

Online Replanning

- Example: the non-deterministic blocks world, again.
- As before, moving a block may be successful or not.
  - If successful, the block ends up where we wanted.
  - If unsuccessful, the block falls on the table.

Action(Move(block, from, to),
PRECOND: On(block, from) AND Clear(block) AND Clear(to)
EFFECT: On(block, to) AND NOT(On(block, from)) AND
Clear(from) AND NOT(Clear(to))

Action(MoveToTable(block, from),
PRECOND: On(block, from) AND Clear(block)
EFFECT: On(block, Table) AND NOT(On(block, from)) AND
Clear(from)

Online Replanning

- In online replanning, we do not have to modify the action definitions.
- The plan we make is the same as in the deterministic case:

Online Replanning

- In online replanning, we do not have to modify the action definitions.
- The plan we make is the same as in the deterministic case:

move(A, B, C)

- When executing the plan:
- Before executing *move(A, B, C)*, we check if the state of the world is as expected.
- If so, we execute the action.
- After we execute the action, we check if the goal is indeed achieved.
  - If not, we replan.

Online Replanning

- Before executing the next action, the agent monitors the enviroment.
- There are different choices as to exactly what to monitor.
  - **Action monitoring:** verify that the preconditions of the next action still hold.
  - **Plan monitoring:** verify that the remaining plan will still succeed, given the current state of the world.
  - **Goal monitoring:** check to see if, given the current state of the world, there is a better plan to follow.
- Action monitoring is the most simple and efficient.
- Goal monitoring can take advantage of unexpected changes that may make it easier to achieve the goal.

Contingency Planning vs. Online Replanning

- The two approaches can be applicable on the same problem.
- They both handle cases where the effects of an action are not deterministic.
- What are the relative pros and cons?
- When would it make more sense to use one approach, vs. the other approach?
- What are the pitfalls of each approach?

Contingency Planning vs. Online Replanning

- Contingency planning becomes infeasible when there are too many possibilities.
  - At some point, the required plan may be too large to fit in memory, or to be able to compute in a reasonable amount of time.
  - Online replanning can help in such cases: focuses resources on analyzing the current situation, as opposed to planning for every possible situation.
- Online planning requires additional computational effort during plan execution.
  - In real-time environments, online planning may be too slow to be useful.
  - In such cases, contingency planning is a better approach: it does more computations offline, so that during execution we don't have to spend as much time computing new plans.
- Obviously, there may be cases where both approaches are too computationally expensive.

Contingency Planning vs. Online Replanning

- How do humans approach this dilemma?
- We typically have conditional plans to handle problems that have high probability to occur.
- We typically do not plan for every possible thing that could go wrong.
  - If something unexpected happens, we adjust our plan on the fly.
- Example: suppose we plan to drive from point A to point B.
  - If we know that the preferred route is often closed due to construction, it makes sense to have an alternate route in mind before starting the trip.
  - However, we typically don't make plans in advance about every possible street that might be blocked, flat tires, dead batteries, etc. We deal with those situations as they happen.

Contingency Planning vs. Online Replanning

- Summary:
  - They both have advantages and disadvantages.
  - In some domains, it may be best to include both approaches:
  - Use contingency planning to be prepared for high-probability situations.
  - Use online replanning to handle any unpredicted problems that arise in practice.