CS 1571

Chapter 10 (Section 10.3)
Knowledge Representation

1

KR

- Last 3 chapters: syntax, semantics, and proof theory of propositional and first-order logic; associated knowledge-based systems
- Theorem provers
  - Prove sentences in FOL. Use inference rules, resolution rule, and resolution refutation
- Deductive retrieval systems
  - Systems based on rules (KBs in Horn form)
  - Prove theorems or infer new assertions (forward, backward chaining)

2

KR, continued

- Chapter 10: what content to put into an agent's KB
- How to represent knowledge of the world
- 10.3: Actions, Situations and Events

3

Actions, Situations, and Events

- The world is dynamic
  - What is true now may not be true tomorrow
  - Changes in the world may be triggered by our activities
- Problems
  - FOL as we had it referred to a static world. How do we represent the change in FOL?
  - How do we represent the actions that change the world?
- Planning problem
  - Find a sequence of actions that achieves some goal
  - A very complex search problem

4

The Situation Calculus

- The robot is in the kitchen.
  - in(robot,kitchen)
- It walks into the living room.
  - in(robot,livingRoom)
- Oops…
- in(robot,kitchen,2:02pm)
- in(robot,livingRoom,2:17pm)
- But what if you are not sure when it was?
- We can do something simpler than rely on time stamps…

5

Situation Calculus

- Logic for reasoning about changes in the state of the world
- The world is described by
  - *Sequences of situations of the current state*
  - *Changes from one situation to another are caused by actions*
- The situation calculus allows us to
  - Describe the initial state and a goal state
  - Build the KB that describes the effect of actions (operators)
  - Prove that the KB and the initial state lead to a goal state
  - Extracts a plan as side-effect of the proof

6

Situation Calculus Ontology

- Actions: terms, such as "forward" and "turn(right))"
- Situations: terms; initial situation s0 and all situations that are generated by applying an action to a situation. result(a,s) names the situation resulting when action a is done in situation s.

7

Situation Calculus Ontology continued

- Fluents: functions and predicates that vary from one situation to the next. By convention, the situation is the last argument of the fluent. ~holding(robot,gold,s0)
- Atemporal or eternal predicates and functions do not change from situation to situation. gold(g1). lastName(wumpus,smith). adjacent(livingRoom,kitchen).

8

Sequences of Actions

- Also useful to reason about action sequences
- All S resultSeq([],S) = S
- All A,Se,S resultSeq([A|Se],S) = resultSeq(Se,result(A,S))

resultSeq([a,b,a2,a3],so) is

result(a3,result(a2,result(b,result(a,s0))

9

Modified Wumpus World

- Won't worry about agent's orientation
- Fluent predicates: at(O,X,S) and holding(O,S)
- Initial situation: at(agent,[1,1],s0) ^ at(g1,[1,2],s0)
- But we want to exclude possibilities from the initial situation too…

10

Initial KB

- All O,X at(O,X,s0) ←→ [O=agent ^ X = [1,1]) v (O=g1 ^ X = [1,2])]
- All O ~holding(O,s0)
- Eternals:
  - gold(g1) ^ adjacent([1,1],[1,2]) ^ adjacent([1,2],[1,1]).

11

Goal: g1 is in [1,1]

At(g1,[1,1],resultSeq(
[go([1,1],[1,2]),grab(g1),go([1,2],[1,1])],
s0)

Or, planning by answering the query:

Exists S at(g1,[1,1],resultSeq(S,s0))

So, what has to go in the KB for such queries to be answered?...

12

Possibility and Effect Axioms

- Possibility axioms:
  - Preconditions → poss(A,S)
- Effect axioms:
  - poss(A,S) → changes that result from that action

13

Axioms for our Wumpus World

- For brevity: we will omit universal quantifiers that range over entire sentence. S ranges over situations, A ranges over actions, O over objects (including agents), G over gold, and X,Y,Z over locations.

14

Possibility Axioms

- The possibility axioms that an agent can
  - go between adjacent locations,
  - grab a piece of gold in the current location, and
  - release gold it is holding

15

Effect Axioms

- If an action is possible, then certain fluents will hold in the situation that results from executing the action
  - Going from X to Y results in being at Y
  - Grabbing the gold results in holding the gold
  - Releasing the gold results in not holding it

16

Frame Problem

- We run into the frame problem
- Effect axioms say what changes, but don't say what stays the same
- A real problem, because (in a non-toy domain), each action affects only a tiny fraction of all fluents

17

Frame Problem (continued)

- One solution approach is writing explicit frame axioms, such as:

At(O,X,S) ^ ~(O=agent) ^ ~holding(O,S) → at(O,X,result(Go(Y,Z),S))

- With F fluent predicates and A actions, need O(AF) frame axioms
- But if an action has at most E effects, then need only O(AE).

18

Representational Frame Problem

- What stays the same?
- A actions, F fluents, and E effects/action (worst case). Typically, E << F
- Want O(AE) versus O(AF) solution

19

Solving the Representational Frame Problem

- Instead of writing the effects of each action, consider how each fluent predicate evolves over time
- Successor-state axioms:
- Action is possible →

  (fluent is true in result state ←→
  action's effect made it true v
  it was true before and action left it alone)

20

Ramification Problem

- Implicit effects, such as: if an agent moves from X to Y, then any gold it is carrying will move too
- For our specific domain, we can solve this by writing a more general successor-state axiom for "at"

Qualification Problem

- Ensuring that all necessary conditions for an action's success have been specified. No complete solution.

Blocks World Example

- In class

23