Title: First-Order Logic

AIMA: Chapter 7 (Sections 7.4 and 7.9)

Introduction to Artificial Intelligence

CSCE 476-876, Spring 2002

URL: `www.cse.unl.edu/~choueiry/S02-476-876`

Berthe Y. Choueiry (Shu-we-ri)

`choueiry@cse.unl.edu`, `(402)472-5444`

Glossary 10:

Assigned: Mon, April 15, 2002

Due date: Mon, April 22, 2002

- Action-value rules
- Causal rules
- Diachronic
- Diagnostic rules
- Effect axioms
- Frame axioms
- Frame problem
- Qualification problem
- Ramification problem
- Situation calculus
- Successor-state axiom
- Synchronic

Look up an encyclopedia or Web, DO NOT USE AIMA)

- Description Logic
- Model-based reasoning (as an AI discipline)

Summary

- First-order logic:
  - basic elements
  - syntax
  - semantics
- Examples

Outline

Wumpus world in FOL

- Situation calculus
  - frame problem
  - qualification problem
  - ramification problem
- Simple reflex agent: pure reflex + internal state
- Model-based agent
- Goal-based agent

Logical Agents for the Wumpus World (I)

Define an interface between the environment and the agent

Perceptual sentence $\begin{cases} \text{percept} \\ \text{time at which it occurred} \end{cases}$

Percepts: Stench, Breeze, Glitter, Bump, Scream

Perceptual sentence at time step 5:

$$Percept([Stench, Breeze, Glitter, None, None], 5)$$

Interfacing with FOL KBs

```
function KB-AGENT(percept) returns an action
   static: KB, a knowledge base
           t, a counter, initially 0, indicating time

   TELL(KB, MAKE-PERCEPT-SENTENCE(percept, t))
   action ← ASK(KB, MAKE-ACTION-QUERY(t))
   TELL(KB, MAKE-ACTION-SENTENCE(action, t))
   t ← t + 1
   return action
```

- $Tell(KB, Percept([Smell, Breeze, Glitter, None, None], 5))$
- Does the KB entail any particular actions at $t = 5$?
  Make-action-query at $t = 5$: $Ask(KB, \exists\ a\ Action(a, 5))$
  Ask returns a binding list $\{a/Grab, a/..., a/...\}$
- $Tell$ informs KB which action is taken
- If σ is a possible binding of variables
  $Ask(KB, S)$ returns some/all σ such that $KB \models S\sigma$

Representing change in the world

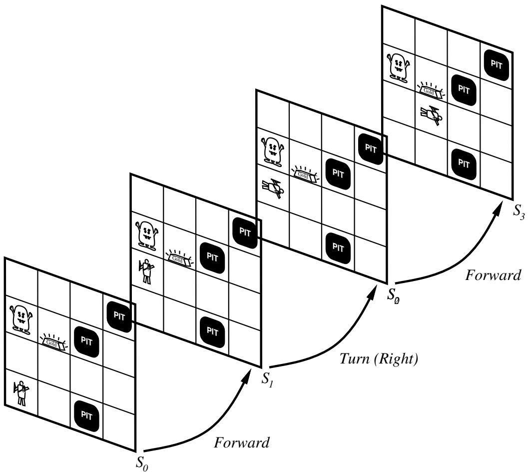
All percepts are added to KB
Percepts history provides a representation of the world
$\longrightarrow$ design rules to infer <u>actions</u> from past and present percepts

Problem: becomes tedious (big number of perceptual sentences)
Solution: maintain an internal model of the world
Problem: the world changes

Situation calculus:
— adds a situation argument to each non-eternal predicate
e.g., $Holding(Gold, Now)$ rather than just $Holding(Gold)$
— world is represented as a sequence of situations
— world evolves according to **diachronic** rules

Situation calculus (I)

Each situation is a snapshot of the state of the world
Situations are generated from previous ones by <u>actions</u>

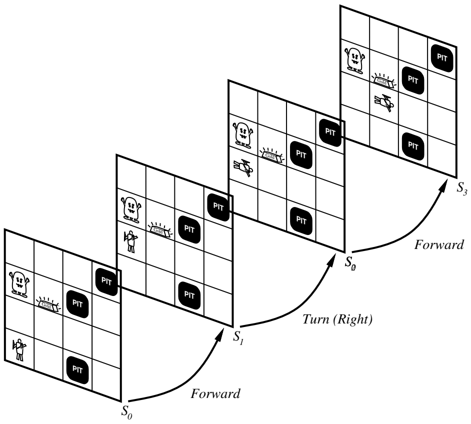


- Relations and properties that are situation dependent:
  $At(Agent, [1, 1], S_0) \wedge At(Agent, [1, 2], S_1)$
- Relations and properties not subject to change: $Wall[0, 1]$

Situation calculus (II)

S_{i+1} resulting from applying *Action* to S_i:

$Result(Action, S_i) = S_{i+1}$



Situations are connected by the *Result* function:

$Result(Forward, S_o) = S_1$

$Result(TurnRight, S_1) = S_2$

etc.

Situation calculus (III)

Actions: described by stating their effects

Effects: properties of the new situation

Effect axiom: describe changes due to action

$\forall\ s\ AtGold(s) \Rightarrow Holding(Gold, Result(Grab, s))$

$\forall\ x, s\ \neg Holding(x, Result(Release, s))$

Frame axiom: describe <u>non-changes</u> due to action

$\forall\ a, x, s\ \ Holding(x, s) \wedge (a \neq Release) \Rightarrow Holding(x, Result(a, s))$

$\forall\ s\ \ HaveArrow(s) \Rightarrow HaveArrow(Result(Grab, s))$

Situation calculus (IV)

Effect axioms + Frame axioms:

provide a complete description of how the world evolves
in response to the agent's actions
can be combined in a single axiom..

Successor-state axiom

$$\forall\, a, x, s \;\; Holding(x, Result(a, s)) \Leftrightarrow \;\; (Holding(x, s) \wedge (a \neq Release)) \vee (AtGold(s) \wedge a = Grab)$$

→ needed for every predicate that may change its value over time

→ lists all the ways in which the predicate can become $\begin{cases} \text{true} \\ \text{false} \end{cases}$

Situation calculus (V) Famous problems

Frame : find an elegant way to handle non-change

1– representation: avoid frame axioms (successor-state axioms)

2– inference: avoid repeated "copy-overs" to keep track of state

Qualification : difficulty of defining the circumstances under which an action is guaranteed to work

true descriptions of real actions require endless caveats: what if gold is slippery or nailed down or ...

Ramification : proliferation of implicit consequences of actions

real actions have many secondary consequences—what about the dust on the gold...

Logical Agents for the Wumpus World (II)

Three architectures:

1. Reflex
   classify percepts, and act accordingly
2. Model-based
   construct an internal representation of the world and use it to act
3. Goal-based
   form goals and try to achieve them

Goal-based usually $\subseteq$ model-based agents

Simple reflex agent (I)

Direct connection :
Rules directly connect percepts to actions
(similar reflexes, instincts)

$$\forall\, s, b, u, c, t\;\; Percept([s, b, Glitter, u, c], t) \Rightarrow Action(Grab, t)$$

Rules for perception : more flexible, more useful form

$$\forall\, b, g, u, c, t\;\; Percept([Stench, b, g, u, c], t) \Rightarrow Stench(t)$$
$$\forall\, s, g, u, c, t\;\; Percept([s, Breeze, g, u, c], t) \Rightarrow Breeze(t)$$
$$\forall\, s, b, u, c, t\;\; Percept([s, b, Glitter, u, c], t) \Rightarrow AtGold(t)$$

$+$ connection from predicates to action choices
$\forall t AtGold(t) \Rightarrow Action(Grab, t)$

Simple reflex agent (II) Limitations

Ideally: retrieve gold, if possible, and climb out

Pure reflex: no representation of the world

Reflex with internal state: do we have the gold already?

$\forall\, t \;\; AtGold(t) \;\wedge\; \neg Holding(Gold, t) \;\Longrightarrow\; Action(Grab, t)$

Problems:

$Holding(Gold, t)$ cannot be observed

Unable to avoid infinite loops (example)

Deducing simple properties

Synchronic rules: (same time)

relate properties of world to other properties of world:

$\forall\, l, t \;\; At(Agent, l, t) \wedge Stench(t) \Rightarrow Smelly(l)$

$\forall\, l, t \;\; At(Agent, l, t) \wedge Breeze(t) \Rightarrow Breezy(l)$

$\rightarrow$ $Smelly, Breezy$ do not need a situation argument

Causal rules: infer effect from cause

$\forall x, y Pit(x) \wedge Adjacent(x, y) \Rightarrow Breezy(y)$

does not say whether squares far away from the pits can be breezy

Diagnostic rules: infer cause from effect

$\forall y Breezy(y) \Rightarrow \exists\, x Pit(x) \wedge Adjacent(x, y)$

does not say where the pit is

Causal rules and diagnostic rules: incomplete

Model-based Agent

Diagnostic reasoning vs. model-based reasoning

Diagnostic rules: do not necessarily derive the strongest possible conclusions from the available information

Model-based reasoning: explicit model of a process and what are the effects/symptoms

> *If the axioms correctly and completely describe the way the world works and the way the percepts are produced, then the inference procedure will be correctly infer the strongest possible description of the world state give the available percepts.*

Preferences among actions

One way: use a simple scale, $Great, Good, Medium, Risky, Deadly$

Action-value system: rules state how desirable actions are

$$\forall\, a, s\ Great(a, s) \Rightarrow Action(a, s)$$

$$\forall\, a, s\ Good(a, s) \wedge (\neg \exists b Great(b, s)) \Rightarrow Action(a, s)$$

$$\forall\, a, s\ Medium(a, s) \wedge (\neg \exists b Good(b, s) \vee Great(b, s)) \Rightarrow Action(a, s)$$

$$\forall\, a, s\ Risky(a, s) \ldots \wedge (\neg \exists b Medium(b, s)\ veeGood(b, s) \vee Great(b, s))..$$

Toward a goal-based agent

Presence of an explicit goal allows the agent to work out a sequence of actions that will achieve the goal.

Three ways:

Inference : write axioms, generate all possible sentences
good solutions+ wasteful solutions

Search : best-first search to find a path to the goal
requires translating agent's knowledge into operators and state representations

Planning : a special-purpose reasoning system to reason about actions

Making plans (I)

Initial condition in KB:
$At(Agent, [1,1], S_0)$
$At(Gold, [1,2], S_0)$

Query: $Ask(KB, \exists\, s\ Holding(Gold, s))$
i.e., in what situation will I be holding the gold?

Answer: $\{s/Result(Grab, Result(Forward, S_0))\}$
i.e., go forward and then grab the gold

This assumes that the agent is interested in plans starting at S_0 and that S_0 is the only situation described in the KB

Making plans (II)

Represent plans as action sequences $[a_1, a_2, \ldots, a_n]$

$PlanResult(p, s)$ is the result of executing p in s

Then the query $Ask(KB, \exists\, p\; Holding(Gold, PlanResult(p, S_0)))$ has the solution $\{p/[Forward, Grab]\}$

Definition of $PlanResult$ in terms of $Result$:

$\forall s PlanResult([], s) = s$

$\forall a, p, s PlanResult([a|p], s) = PlanResult(p, Result(a, s))$

Planning systems are special-purpose reasoners designed to do this type of inference more efficiently than a general-purpose reasoner

Summary

First-order logic

— objects and relations are semantic primitives

— syntax: constants, functions, predicates, equality, quantifiers

Increased expressive power: sufficient to define Wumpus world

Situation calculus

— conventions for describing actions and change in FOL

— can formulate planning as inference on a situation calculus KB