Foundations of AI
9. Knowledge Representation: Modeling with Logic

Concepts, Actions, Time, & all the rest

Wolfram Burgard and Luc De Raedt

Contents

- Knowledge Representation and Reasoning
- Concepts & Description logics
- Logical agents for the WUMPUS World
- Situation calculus
- The frame problem
- Time intervals and their description

Knowledge Representation and Reasoning

- Often, our agents need knowledge before they can start to act intelligently
- They then also need some reasoning component to exploit the knowledge they have
- Examples:
  - Knowledge about the important concepts in a domain
  - Knowledge about actions one can perform in a domain
  - Knowledge about temporal relationships between events
  - Knowledge about the world and how properties are related to actions

Categories and Objects

- We need to describe the objects in our world using categories
- Necessary to establish a common category system for different applications (in particular on the web)
- There are a number of quite general categories everybody and every application uses
**The Upper Ontology:**
**A General Category Hierarchy**

- AbstractObjects
- Numbers
- RepresentationalObjects
- GeneralizedEvents
  - intervals
  - Place
  - PhysicalObjects
  - Processes

- Categories
  - Sentences
  - Measurements
- Moments
  - Things
  - Stuff
  - Animals
  - Agents
  - Solids
  - Liquids
  - Gases
- Humans

**Description Logics**

- How to describe more specialized things?
- Use definitions and/or necessary conditions referring to other already defined concepts:
  - a parent is a human with at least one child
  - More complex description:
    - a proud-grandmother is a human, which is female with at least two children that are in turn parents whose children are all doctors

**Reasoning Services in Description Logics**

- **Subsumption:** Determine whether one description is more general than (subsumes) the other
- **Classification:** Create a subsumption hierarchy
- **Satisfiability:** Is a description satisfiable?
- **Instance relationship:** Is a given object instance of a concept description?
- **Instance retrieval:** Retrieve all objects for a given concept description

**Special Properties of Description Logics**

- Semantics of description logics can be given using ordinary PL1
  - Alternatively, DLs can be considered as modal logics
- Reasoning for most DLs is much more efficient than for PL1
- Nowadays, W3C standards such as OWL (formerly DAML+OIL) are based on description logics
Logic-Based Agents That Act

```plaintext
function KB (Agent1 perceive) returns an action
    static: KB, a knowledge base
    t, a counter, initially 0, indicating time
    TELL(KB, MAKE-PERCEPT-Sentence perceive, t)
    action ← ASK(KB, MAKE-ACTION-QUERY(T))
    TELL(KB, MAKE-ACTION-Sentence(action, t))
    t ← t + 1
    return action
```

Query (Make-Action-Query): $\exists x \text{Action}(x, t)$

A variable assignment for $x$ in the WUMPUS world example should give the following answers: $\text{turn(right)}$, $\text{turn(left)}$, $\text{forward}$, $\text{shoot}$, $\text{grab}$, $\text{release}$, $\text{climb}$

09/9

Reflex Agents

... only react to percepts.

Example of a percept statement (at time 5):

$\text{Percept(stench, breeze, glitter, none, none, 5)}$

1. $\forall v, g, u, c, t [\text{Percept(stench, b, g, u, c, t)} \Rightarrow \text{Stench}(t)]$
2. $\forall v, g, u, c, t [\text{Percept(s, breeze, g, u, c, t)} \Rightarrow \text{Breeze}(t)]$
3. $\forall v, g, u, c, t [\text{Percept(s, b, glitter, u, c, t)} \Rightarrow \text{AtGold}(t)]$
... 

2. Step: Choice of action

$\forall t [\text{AtGold}(t) \Rightarrow \text{Action(grab, t)}]$

... 

Note: Our reflex agent does not know when it should climb out of the cave and cannot avoid an infinite loop.

09/10

Model-Based Agents

... have an internal model

- of all basic aspects of their environment,
- of the executability and effects of their actions,
- of further basic laws of the world, and
- of their own goals.

Important aspect: How does the world change?

$\Rightarrow \text{Situation calculus}$: (McCarthy, 63).

09/11

Situation Calculus

- A way to describe dynamic worlds with PL1.
- States are represented by terms.
- The world is in state $s$ and can only be altered through the execution of an action: $do(a, s)$ is the resulting situation, if $a$ is executed.
- Actions have preconditions and are described by their effects.
- Relations whose truth value changes over time are called fluents. Represented through a predicate with two arguments: the fluent and a state term. For example, $At(x, s)$ means, that in situation $s$, the agent is at position $x$. $Holding(y, s)$ means that in situation $s$, the agent holds object $y$.
- Atemporal or eternal predicates, e.g., $\text{Portable}(gold)$.

09/12
Example: WUMPUS-World

Let $s_0$ be the initial situation and

\[
s_1 = \text{do(forward, } s_0) \\
s_2 = \text{do(turn(right), } s_1) \\
s_3 = \text{do(forward, } s_2)
\]

Description of Actions

Preconditions: In order to pick something up, it must be both present and portable:

\[
\forall x, s[\text{Poss(grab}(x), s) \iff \text{Present}(x, s) \land \text{Portable}(x)]
\]

In the WUMPUS-World:

\[
\text{Portable}(\text{gold}), \forall s[\text{AtGold}(s) \Rightarrow \text{Present}(\text{gold}, s)]
\]

Positive effect axiom:

\[
\forall x, s[\text{Poss(grab}(x), s) \Rightarrow \text{Holding}(x, \text{do(grab}(x), s))]\]

Negative effect axiom:

\[
\forall x, s[\sim \text{Holding}(x, \text{do(release}(x), s))]
\]

The Frame Problem

We had: $\text{Holding}(\text{gold}, s_0)$.

Following situation: $\sim \text{Holding}(\text{gold}, \text{do(release}(\text{gold}), s_0))$?

We had: $\sim \text{Holding}(\text{gold}, s_2)$.

Following situation: $\sim \text{Holding}(\text{gold}, \text{do(turn(right), } s_0))$?

- We must also specify which fluents remain unchanged!
- The frame problem: Specification of the properties that do not change as a result of an action.

$\Rightarrow$ Frame axioms must also be specified.

Number of Frame Axioms

\[
\forall a, x, s[\text{Holding}(x, s) \land (a \neq \text{release}(x)) \Rightarrow \text{Holding}(x, \text{do}(a, s))]
\]

\[
\forall a, x, s[\sim \text{Holding}(x, s) \land ((a \neq \text{grab}(x)) \lor \sim \text{Poss(grab}(x), s) \Rightarrow \sim \text{Holding}(x, \text{do}(a, s))]
\]

Can be very expensive in some situations, since $O(|F| \times |A|)$ axioms must be specified, $F$ being the set of fluents and $A$ being the set of actions.
**Successor-State Axioms**

A more elegant way to solve the frame problem is to fully describe the successor situation:

- true after action ⇔ [ action made it true ∨ already true and the action did not falsify it ]

Example for **grab** :

∀a, x, s[ Holding(x, do(a, s))
⇔ [(a = grab(x)) ∧ Poss(a, s)) ∨ (Holding(x, a) ∧ a ≠ release(x))] ]

Can also be automatically compiled by only giving the effect axioms (and then applying explanation closure). Here we suppose that only certain effects can appear.

**Limits of this Version of Situation Calculus**

- No explicit time. We cannot discuss how long an action will require, if it is executed.
- Only one agent. In principle, however, several agents can be modeled.
- No parallel execution of actions.
- Discrete situations. No continuous actions, such as moving an object from A to B.
- Closed world. Only the agent changes the situation.
- Determinism. Actions are always executed with absolute certainty.
  → Nonetheless, sufficient for many situations.

**Qualitative Descriptions of Temporal Relationships**

We can describe the temporal occurrence of event/actions:

- **absolute** by using a date/time system
- **relative** with respect to other event occurrences
- **quantitatively**, using time measurements (5 secs)
- **qualitatively**, using comparisons (before/overlaps)

**Allen’s Interval Calculus**

- Allen proposed a calculus about relative order of time intervals
- Allows us to describe, e.g.,
  - Interval I occurs before interval J
  - Interval J occurs before interval K
- and to conclude
  - Interval I occurs before interval K
  → 13 jointly exhaustive and pair-wise disjoint relations between intervals
Allen’s 13 Interval Relations

- $I < J, J > I$ before/after
- $I \cap J, J \cap I$ meets
- $I \cap J, J \cap I$ overlaps
- $I \cup J, J \cup I$ starts
- $I \cup J, J \cup I$ during
- $I \cup J, J \cup I$ finishes
- $I = J$

Examples

- Using Allen’s relation system one can describe temporal configurations as follows:
  - $X < Y, Y o Z, Z > X$

- One can also use disjunctions (unions) of temporal relations:
  - $X (<, m) Y, Y (o, s) Z, Z > X$

Reasoning in Allen’s Relations System

How do we reason in Allen’s system
- Checking whether a set of formulae is satisfiable
- Checking whether a temporal formula follows logically

- Use a constraint propagation technique for CSPs with infinite domains (3-consistency), based on composing relations

Constraint Propagation

$X \rightarrow (<, m) \rightarrow Y$

$X < Y s Z = X Z$
$X < Y o Z = X Z$
$X m Y s Z = X Z$
$X m Y o Z = X Z$

Do that for every triple until nothing changes anymore, then CSP is 3-consistent
Concluding Remarks:
Use of Logical Formalisms

- In many (but not all) cases, full inference in PL1 is simply too slow (and therefore too unreliable).
- Often, special (logic-based) representational formalisms are designed for specific applications, for which specific inference procedures can be used. Examples:
  - Description logics for representing conceptual knowledge.
  - James Allen’s time interval calculus for representing qualitative temporal knowledge.
  - Planning: Instead of situation calculus, this is a specialized calculus (STRIPS) that allows us to address the frame problem.

→ Generality vs. efficiency
→ In every case, logical semantics is important!

09/25