Lecture #12 – Knowledge Representation

Outline

- Ontological engineering
- Categories and objects
- Actions, situations and events
- Mental events and mental objects
- The internet shopping world
- Reasoning systems for categories
- Reasoning with default information
- Truth maintenance systems

Ontological Engineering

- How to create more general and flexible representations.
  - Concepts like actions, time, physical objects and beliefs
  - Operates on a bigger scale than knowledge engineering
- Define general framework of concepts
  - Upper ontology
- Limitations of logic representation
  - Red, green and yellow tomatoes: exceptions and uncertainty

The Upper Ontology of the World

Differences with Special-Purpose Ontologies

- A general-purpose ontology should be applicable in more or less any special-purpose domain.
  - Add domain-specific axioms
- In any sufficiently demanding domain different areas of knowledge need to be unified.
  - Reasoning and problem solving could involve several areas simultaneously
- What do we need to express?
  - Categories, Measures, Composite objects, Time, Space, Change, Events, Processes, Physical Objects, Substances, Mental Objects, Beliefs

Categories and Objects

- KR requires the organization of objects into categories
  - Interaction at the level of the object
  - Reasoning at the level of categories
- Categories play a role in predictions about objects
  - Based on perceived properties
- Categories can be represented in two ways by first-order logic (FOL)
  - Predicates: apple(x)
  - Reification of categories into objects: apples
- Category = set of its members
Category Organization

- **Relation = inheritance:**
  - All instances of food are edible, fruit is a subclass of food and apples is a subclass of fruit then an apple is edible.
- Defines a taxonomy

FOL and Categories

- An object is a member of a category
  - \( \text{MemberOf}(BB12, \text{Basketballs}) \)
- A category is a subclass of another category
  - \( \text{SubsetOf}(\text{Basketballs}, \text{Balls}) \)
- All members of a category have some properties
  - \( \forall x (\text{MemberOf}(x, \text{Basketballs}) \Rightarrow \text{Round}(x)) \)
- A category as a whole has some properties
  - \( \text{MemberOf}(\text{Dogs}, \text{DomesticatedSpecies}) \)

Relations Between Categories

- Two or more categories are disjoint if they have no members in common:
  - \( \text{Disjoint}(s) \iff \forall c_1, c_2 \in s \land c_1 \neq c_2 \Rightarrow \text{Intersection}(c_1, c_2) = \{\} \)
  - Example: \( \text{Disjoint}([\text{animals}, \text{vegetables}]) \)
- A set of categories \( s \) constitutes an exhaustive decomposition of a category \( c \) if all members of the set \( s \) are covered by categories in \( s \):
  - \( \text{E.D.}(s, c) \iff \forall i, i \in c \Rightarrow \exists c_2 \in s \land i \in c_2 \)

Natural Kinds

- Many categories have no clear-cut definitions (chair, bush, book)
- Tomatoes: sometimes green, red, yellow, black. Mostly round.
- One solution: category Typical(Tomatoes).
  - \( \forall x, x \in \text{Typical(Tomatoes)} \Rightarrow \text{Red}(x) \land \text{Spherical}(x) \)
- We can write down useful facts about categories without providing exact definitions.
- What about “bachelor”? Quine challenged the utility of the notion of strict definition. We might question a statement such as “the Pope is a bachelor”

Physical Composition

- One object may be part of another:
  - \( \text{PartOf}(\text{Bucharest}, \text{Romania}) \)
  - \( \text{PartOf}(\text{Romania}, \text{EasternEurope}) \)
  - \( \text{PartOf}(\text{EasternEurope}, \text{Europe}) \)
- The PartOf predicate is transitive (and irreflexive), so we can infer that PartOf(Bucharest,Europe)
- More generally:
  - \( \forall x \text{ PartOf}(x, x) \)
  - \( \forall x y z \text{ PartOf}(x, y) \land \text{PartOf}(y, z) \Rightarrow \text{PartOf}(x, z) \)
- Often characterized by structural relations among parts.
  - E.g. Biped(a) \( \Rightarrow (\exists l_1, l_2, b, \text{Leg}(l_1) \land \text{Leg}(l_2) \land \text{Body}(b) \land \text{PartOf}(l_1, a) \land \text{PartOf}(l_2, a) \land \text{PartOf}(b, a) \land \text{AttachedTo}(l_1, b) \land \text{AttachedTo}(l_2, b) \land l_1 \neq l_2 \land (\forall l_3 \text{Leg}(l_3) \Rightarrow (l_3 = l_1 \lor l_3 = l_2))) \)
Artificial Intelligence

Measurements
- Objects have *height*, *mass*, *cost*, ....
- Values that we assign to these are *measures*
- Combine *Unit functions* with a *number*:
  \[ \text{Length}(L1) = \text{Inches}(1.5) = \text{Centimeters}(3.81) \]
- *Conversion* between units:
  \[ \forall i \in \text{Centimeters}, (2.54 \times i) = \text{Inches}(i). \]
- Some *measures* have no *scale*: Beauty, Difficulty, etc.
  - Most important aspect of measures: is that they are *orderable*
  - *Don't care about the actual numbers*
    - E.g. An apple can have deliciousness .9 or .1.

Actions, Events and Situations
- Reasoning about the *outcome of actions* is central to KB-agent
- How can we keep track of *location* in FOL?
  - Remember the multiple copies in PL.
- Representing *time* by *situations* (states resulting from the execution of actions).
  - *Situation calculus*

Situation Calculus
- *Actions* are logical terms
- *Situations* are logical terms consisting of
  - The initial situation \( I \)
  - All situations resulting from the action on \( I (=\text{Result}(a,s)) \)
- *Fluents* are functions and predicates that vary from one situation to the next.
  - E.g. \( \neg \text{Holding}(G1, S0) \)
- *Eternal predicates* are also allowed
  - E.g. \( \text{Gold}(G1) \)

Results of action sequences are determined by the individual actions
- *Projection task*: an SC agent should be able to deduce the outcome of a sequence of actions
- *Planning task*: find a sequence that achieves a desirable effect

Describing Change
- Simplest Situation calculus requires two axioms to describe change:
  - *Possibility axiom*: when is it possible to do the action
    \[ \text{At}(\text{Agent}, x, s) \land \text{Adjacent}(x, y) \Rightarrow \text{Poss}(\text{Go}(x, y), s) \]
  - *Effect axiom*: describe changes due to action
    \[ \text{Poss}(\text{Go}(x, y), s) \Rightarrow \text{At}(\text{Agent}, y, \text{Result}(\text{Go}(x, y), s)) \]
- What stays the same?
  - *Frame problem*: how to represent all things that stay the same?
  - *Frame axiom*: describe non-changes due to actions
    \[ \text{At}(o, x, s) \land (o \neq \text{Agent}) \land \neg \text{Holding}(o, s) \Rightarrow \text{At}(o, x, \text{Result}(\text{Go}(y, z), s)) \]
The Frame Problem

- For **fluents** and **actions** we need **A*F** frame axioms to describe other objects which are **stationary unless** they are held
  - Must write down the **effect** of each action

**Solution** - describe how each fluent changes over time

- Successor-state axiom:
  \[ \text{Pos}(a,s) \Rightarrow (\text{At}(\text{Agent}, y, \text{Result}(a,s)) \iff (a = \text{Go}(x,y)) \lor (\text{At}(\text{Agent}, y, s) \land a \neq \text{Go}(y,z)) \]

- Note that next state is completely specified by current state.

Other Problems

- How to deal with **secondary (implicit)** effects?
  - If the agent is carrying the gold and the agent moves then the gold moves too.
  - **Ramification problem**

- How to decide efficiently whether fluents hold in the future?
  - **Inferential frame problem**

Extensions

- Event calculus (when actions have a duration)
- Process categories

Mental Events and Objects

- So far, KB agents can have **beliefs** and deduce new beliefs
- What about knowledge about beliefs?
- What about knowledge about the inference process?
  - Requires a model of the mental objects in someone’s head and the processes that manipulate those objects

**Relationships** between agents and mental objects:

- \( \text{Believes}(\text{Lois}, \text{Flies}(\text{Superman})) \)
  - \( \text{Flies}(\text{Superman}) \) being a function ... a candidate for a mental object (reification)

- Agent can now **reason about the beliefs of agents**

The Internet Shopping World

- A Knowledge Engineering example
- An agent that helps a buyer to **find** product offers on the internet.
  - **IN** = product description (precise or \( \neg \) precise)
  - **OUT** = list of webpages that offer the product for sale
- **Environment** = WWW
- **Percepts** = web pages (character strings)
  - Extracting useful information required

Find Relevant Product Offers

- **RelevantOffer**(page, url, query) \( \iff \)
  - Relevant(page, url, query) \( \land \)
  - Offer(page)

- **Write axioms** to define **Offer**(x)

Find relevant pages: **Relevant**(x, y, z) ?

- Start from an initial set of stores.
- What is a relevant category?
- What are relevant connected pages?

Find Relevant Product Offers, cont

- Require rich category vocabulary
  - synonymy
  - ambiguity

- **How to retrieve pages**? **GetPage**(url) ?
  - Procedural attachment

- Compare offers (information extraction)
Reasoning systems for Categories

How to organize and reason with categories?

- Semantic networks
  - Visualize knowledge-base
  - Efficient algorithms for category membership inference

- Description logics
  - Formal language for constructing and combining category definitions
  - Efficient algorithms to decide subset and superset relationships between categories.

Semantic Networks

- Logic vs. semantic networks
- Many variations
  - All represent individual objects, categories of objects and relationships among objects.
- Allows for inheritance reasoning
  - Female persons inherit all properties from person.
  - Cfr. OO programming.
- Inference of inverse links
  - SisterOf vs. HasSister

Semantic Network Example

- Drawbacks
  - Links can only assert binary relations
  - Can be resolved by reification of the proposition as an event
- Representation of default values
  - Enforced by the inheritance mechanism

Description Logics

- Are designed to describe definitions and properties about categories
  - A formalization of semantic networks

- Principal inference task is
  - Subsumption: checking if one category is the subset of another by comparing their definitions
  - Classification: checking whether an object belongs to a category.
  - Consistency: whether the category membership criteria are logically satisfiable

Reasoning with Default Information

- “The following courses are offered: ICS101, ICS110, ICS111, EE160”
- Four (db)
  - Assume that this information is complete (not asserted ground atomic sentences are false)
  - = CLOSED WORLD ASSUMPTION
  - Assume that distinct names refer to distinct objects
  - = UNIQUE NAMES ASSUMPTION
  - Between one and infinity (logic)
  - Does not make these assumptions
  - Requires completion.
Truth Maintenance Systems

- Many inferences have default status rather than being absolutely certain
  - Inferred facts can be wrong and need to be retracted = belief revision.
  - Assume KB contains sentence P and we want to execute $\text{TELL} (\text{KB}, \neg P)$
    - To avoid contradiction: $\text{RETRACT} (\text{KB}, P)$
    - But what about sentences inferred from P?
- Truth maintenance systems are designed to handle these complications.

Summary

- Ontological engineering
- Categories and objects
- Actions, situations and events
- Mental events and mental objects
- The internet shopping world
- Reasoning systems for categories
- Reasoning with default information
- Truth maintenance systems