Knowledge Engineering

Semester 2, 2004-05

Michael Rovatsos
mrovatso@inf.ed.ac.uk

School of informatics
The University of Edinburgh

Lecture 6 – Further Issues in Ontological Reasoning
1st February 2005

Where are we?

Last time . . .

- we discussed basics of ontologies
  - definitions, examples
  - formalising certain kinds of knowledge
  - problems: multiple inheritance, frame problem(s), intrinsic/extrinsic properties of objects

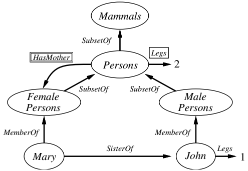
Today . . .

- Category reasoning systems
  - Semantic Networks
  - Description Logics
- Reasoning with default information

Semantic Networks

- Idea: represent information about categories and their attributes graphically
- Historical dispute between graphical notations (such as semantic networks) and logic
- Semantic networks with well-defined semantics can be regarded as a kind of logic, but often more convenient
- Particularly well-suited for representing inheritance information (but problem of multiple inheritance)

Example

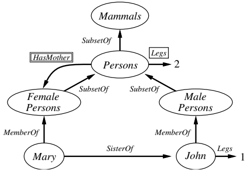


Semantic Networks

Many different notations, but most common elements include:

- Nodes for object and category names (here ovals)
- Edges for relations among objects/categories
- Different types of edge labels depending on whether relation is of type
  - object-object (e.g. $SisterOf(Mary, John)$)
  - object-category (e.g. $John \in MalePersons$))
  - category-object (e.g. $\forall x\ x \in Persons \Rightarrow Legs(x, 2)$)
  - category-category (e.g. $\forall x\ x \in Persons \Rightarrow [\forall y\ HasMother(x, y) \Rightarrow y \in FemalePersons]$)

Example

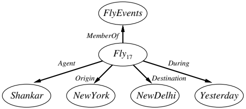


Semantic Networks

- Note distinction between relations concerning categories and those referring to their members
- Inheritance reasoning very simple: follow links from object we want to retrieve information about until suitable relation is found (e.g. $Legs(Mary, ?)$)
- Dealing with multiple inheritance problems: simple kind of **default reasoning** (use attributes of super-class unless overridden by more specific category)
- Inability to model relations with more than two arguments
  ➡ can be partially solved by **reification** of relations

Semantic Networks – Relation Reification

Reify event $Fly(Shankar, NewYork, NewDelhi)$ as object Fly_{17} belonging to category $FlyEvents$:



Fairly awkward: Can we do this for the entire category of "fly events"?

Expressiveness

- Reification enables representation of any function-free, ground atomic sentence
- But no disjunction, negation, existential quantification
  ➡ not full expressiveness of first-order logic
- Trade-off: Possible to introduce all elements of FOL, but tractability problems
- Common middle solution: **procedural attachments** for particular predicates

Description Logics

- Description Logics are notations used to facilitate definitions and descriptions of categories
- Idea: Describe what semantic networks mean while retaining taxonomic structure as an organising principle
- Main tasks:
  - Subsumption: Is a category a subset of another category?
  - Classification: Does an object belong to a category?
  - Consistency: Is a category definition consistent?
- CLASSIC: Make statements about categories (treated as objects in FOL sense)

Description Logics

Example: "All men with at least three sons who are unemployed and married to doctors and at least two daughters who are all professors in physics or math departments"

$$\begin{aligned}&And(Man, AtLeast(3, Son), AtMost(2, Daughter),\\ &\quad All(Son, And(Unemployed, Married, All(Spouse, Doctor))),\\ &\quad All(Daughter, And(Prof, Fills(Department, Physics, Math))))\end{aligned}$$

But: no direct description of subset relation! (subset information derived from descriptions)

Description Logics

Advantages:

- Easy to describe categories directly, without speaking about their members
- Tractability of inference

Disadvantages:

- No negation, disjunction only limited as enumeration over objects (but not descriptions)
- Even in this simple notation, subsumption can be exponential in tehe worst case!

Importance: Foundation for Semantic Web logics!

Reasoning with Default Information

Already discussed defaults, but what is their semantics?

- Closed-world assumption
- Negation as failure
- Circumscription
- Default logic

Closed-World Assumptions

- Example: Suppose we know *"John loves Mary"* and *"Jack loves Mary"*, what would a reasonable answer to the query *"Who loves Mary?"* be?
  - *"John and Jack love Mary"*
  - ... and maybe some other guys do, too?!?
  - How about *"Does Jim love Mary?"*
- **Closed-world assumption (CWA):** assume that information provided is *complete* (sentences not assumed to be true are false)
- **Unique names assumption (UNA):** objects with different names are different
- In FOL, these (seemingly intuitive) assumptions have to be made explicit

Completion

Procedure to express CWA and UNA (for Horn clauses):

1. Gather all clauses with same predicate name P and same arity
2. Translate $P(t_1, \dots t_n) \Leftarrow Body$ to $P(v_1, \dots, v_n) \Leftarrow \exists w_1 \dots w_n \, [v_1, \dots, v_n] = [t_1, \dots, t_n] \wedge Body$ (v_i new variables, w_i original ones)
3. Combine the results into one big disjunctive clause
4. Replace "$\Rightarrow$" by equivalence "$\Leftrightarrow$"

Example

1. $Loves(x, Mary) \Leftarrow Rich(x)$, $Loves(Jack, Mary)$
2. Result:

$$\begin{aligned} Loves(v_1, v_2) &\Leftarrow \exists x \, [v_1, v_2] = [x, Mary] \wedge Rich(x) \\ Loves(v_1, v_2) &\Leftarrow [v_1, v_2] = [Jack, Mary] \end{aligned}$$

3.-4. Result:

$$\begin{aligned} Loves(v_1, v_2) \Leftrightarrow & \exists x \, [v_1, v_2] = [x, Mary] \wedge Rich(x) \\ & \vee [v_1, v_2] = [Jack, Mary] \end{aligned}$$

Does this remind you of anything? ➡ Successor-state axioms

Closed-World Assumption/Negation as failure

- Can use similar scheme for UNA (how? problem?)
- CWA allows for construction of **minimal models** (but, e.g. in case of "=", these can be "maximal"!)
- **Negation as failure:** For example, in Prolog *not* Q is true if Q cannot be proven
- But not a good method for dealing with ignorance!

Nonmonotonic reasoning

- So far, two examples for default reasoning:
  1. Overriding values from super-category in semantic networks
  2. Overriding negative facts by adding positive literals
- More general phenomenon: set of conclusions does not always grow monotonically as new information arrives
- Unfortunately, in FOL this is the case!
  If $KB \models \alpha$ then $KB \wedge \beta \models \alpha \ldots$
- How can we deal with such **nonmonotonicity**?
- Two methods: **circumscription**/**default logic**

Circumscription

- Idea: more precise version of CWA
- Assume certain predicates (the ones to be circumscribed) to be false unless opposite is known to be true
- If *Abnormal* is assumed to be circumscribed, we are allowed to infer $\neg Abnormal(x)$ unless $Abnormal(x)$ is explicitly known
- Prefer those models for a formula that minimise "abnormal" objects (**model preference**)
- A sentence is entailed if true in all preferred models (rather than all models)

Circumscription

- Example: remember Nixon Diamond?

$$Republican(Nixon) \wedge Quaker(Nixon)$$
$$Republican(x) \wedge \neg Abnormal_1(x) \Rightarrow \neg Pacifist(x)$$
$$Quaker(x) \wedge \neg Abnormal_2(x) \Rightarrow Pacifist(x)$$

- Two preferred models: either $Abnormal_1(Nixon)$ and $Pacifist(Nixon)$ or $Abnormal_2(Nixon)$ and $\neg Pacifist(Nixon)$
- Both equally "abnormal" ➡ no conclusion drawn
- Additional preference ordering between different types of "abnormal" properties (**prioritised circumscription**) can be introduced

Default Logic

- Use explicit **default rules** $P : J_1, \ldots J_n / C$ to express that
  - under prerequisite P
  - infer conclusion C
  - unless any justification J_i can be proven false
- Back to our example:

$$Republican(Nixon) \wedge Quaker(Nixon)$$
$$Republican(x) : \neg Pacifist(x) / \neg Pacifist(x)$$
$$Quaker(x) : Pacifist(x) / Pacifist(x)$$

- Consider **extension**, i.e. maximal set of conclusions that can be drawn and the justifications are consistent with these conclusions
- As with circumscription, we get two extensions in the example

Discussion

These techniques solve part of the default reasoning problem, but ...

- What are suitable default rules?
- Are they context dependent?
- What are the implications of having a wrong default rule?
- Connection to probability theory?

Summary

- Discussed category reasoning systems and default reasoning
- Semantic networks: simple, tractable graphical models of ontological knowledge (but limited)
- Description logics: simplified "logic for categories", foundation for Semantic Web reasoning systems
- Default reasoning: dealing with closed worlds and nonmonotonic reaosoning
- Next time: In-depth example of a particular KR & R method
  - **Model-based reasoning**