Knowledge Representation and Reasoning

Chapter 12

Some material adopted from notes by Andreas Geyer-Schulz and Chuck Dyer
Overview

• Approaches to knowledge representation
  • Deductive/logical methods
    – Forward-chaining production rule systems
    – Semantic networks
    – Frame-based systems
    – Description logics
  • Abductive/uncertain methods
    – What’s abduction?
    – Why do we need uncertainty?
    – Bayesian reasoning
    – Other methods: Default reasoning, rule-based methods, Dempster-Shafer theory, fuzzy reasoning
Semantic Networks

• Simple representation scheme: a graph of labeled nodes and labeled, directed arcs to encode knowledge
  – often used for static, taxonomic, concept dictionaries

• Typically used with a special set of accessing procedures that perform “reasoning”
  – e.g., inheritance of values and relationships

• Semantic networks popular in 60s & 70s, less used in ‘80s &’90s, back since‘00s as RDF
  – less expressive than other formalisms: both a feature & bug!

• The **graphical depiction** associated with a semantic network is a significant reason for their popularity
Nodes and Arcs

Arcs define binary relationships that hold between objects denoted by the nodes

\[
\begin{align*}
\text{mother}(\text{john}, \text{sue}) & \\
\text{age}(\text{john}, 5) & \\
\text{wife}(\text{sue}, \text{max}) & \\
\text{age}(\text{max}, 34) & \\
\ldots & 
\end{align*}
\]
What do these nodes and edges mean?
Semantic Networks

• ISA (is-a) or AKO (a-kind-of) relations often used to link instances to classes and classes to super-classes
• Some links (e.g. hasPart) are inherited along ISA paths
• Meaning of a semantic net can be relatively informal or very formal
  – often defined by implementation
Reification

- Non-binary relationships can be represented by “turning the relationship into an object”
- Logicians and philosophers call this reification – reify v : consider an abstract concept to be real
- We might want to represent the generic give event as a relation involving three things: a giver, a recipient and an object, give(john, mary, book32)
Individuals and Classes

Many semantic networks distinguish
– nodes representing individuals & those representing classes
– E.g., subclass from instance_of relation

Formalization must deal with nodes like Bird
– OWL uses punning
Inference by Inheritance

• One kind of reasoning done in semantic nets is inheritance along subclass & instance links
  – It’s like inheritance in object-oriented languages

• Semantic networks differ in details of
  – Inheriting along subclass or instance links, e.g.
    • Only inherit values on instance links
  – Inheriting multiple different values, e.g.
    • All possible values are inherited, or
    • Only the “closest” value or values are inherited
From Semantic Nets to Frames

- Semantic networks evolved into frame representation languages in the 70s and 80s
- Frames like a OO classes with more meta-data
  - Cf. AI’s focus on knowledge over data
- A frame has a set of slots
- Slots represents relations to other frame or literal values (e.g., number or string)
- A slot has one or more facets
- A facet represents some aspect of the relation
Facets

• A slot in a frame can hold more than a value
• Other facets might include:
  – **Value**: current fillers
  – **Default**: default fillers
  – **Cardinality**: minimum and maximum number of fillers
  – **Type**: type restriction on fillers, e.g. another frame
  – **Procedures**: if-needed, if-added, if-removed
  – **Salience**: measure on the slot’s importance
  – **Constraints**: attached constraints or axioms
• In some systems, the slots themselves are instances of frames.
(a) A frame-based knowledge base

(b) Translation into first-order logic

```
Rel(Alive, Animals, T)
Rel(Flies, Animals, F)

Birds ⊆ Animals
Mammals ⊆ Animals

Rel(Flies, Birds, T)
Rel(Legs, Birds, 2)
Rel(Legs, Mammals, 4)

Penguins ⊆ Birds
Cats ⊆ Mammals
Bats ⊆ Mammals

Rel(Flies, Penguins, F)
Rel(Legs, Bats, 2)
Rel(Flies, Bats, T)

Opus ∈ Penguins
Bill ∈ Cats
Pat ∈ Bats

Name(Opus, "Opus")
Name(Bill, "Bill")
Name(Pat, "Pat")

Friend(Opus, Bill)
Friend(Bill, Opus)
Friend(Bill, Opus)
```

(a) A frame-based knowledge base

(b) Translation into first-order logic
Description Logics

• Description logics are a family of frame-like KR systems with a formal semantics
  – E.g., KL-ONE, OWL
• Additional kind of inference is automatic classification of Frames and objects
  – Automatically finding right place in a hierarchy
• Many current systems limit languages to support decidably complete reasoning
• The Semantic Web language OWL based on description logic
Beyond Deduction

• Logical deduction is not the only kind of reasoning that’s useful
Abduction

Abduction: reasoning that derives an explanatory hypothesis from a given set of facts

- Inference result is a hypothesis that, if true, could explain the occurrence of the given facts
- Inherently unsound and uncertain

Example: Medical diagnosis

- Facts: symptoms, test results, other observed findings
- KB: causal associations between diseases & symptoms
- Reasoning: diseases whose presence would causally explain the occurrence of the given manifestations
## Deduction, Abduction, Induction

<table>
<thead>
<tr>
<th>Deduction: major premise:</th>
<th>All balls in the box are black</th>
</tr>
</thead>
<tbody>
<tr>
<td>minor premise:</td>
<td>These balls are from the box</td>
</tr>
<tr>
<td>conclusion:</td>
<td>These balls are black</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Abduction: rule:</th>
<th>All balls in the box are black</th>
</tr>
</thead>
<tbody>
<tr>
<td>observation:</td>
<td>These balls are black</td>
</tr>
<tr>
<td>explanation:</td>
<td>These balls are from the box</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Induction: case:</th>
<th>These balls are from the box</th>
</tr>
</thead>
<tbody>
<tr>
<td>observation:</td>
<td>These balls are black</td>
</tr>
<tr>
<td>hypothesized rule:</td>
<td>All ball in the box are black</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Deduction: from causes to effects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abduction: from effects to causes</td>
</tr>
<tr>
<td>Induction: from specific cases to general rules</td>
</tr>
</tbody>
</table>
Non-monotonic reasoning

• Abduction is *non-monotonic* reasoning
• Monotonic: your knowledge can only increase
  – Propositions don’t change their truth value
  – You never unknow things
• In abduction: plausibility of hypotheses can increase/decrease as new facts are collected
• In contrast, deductive inference is *monotonic*: it never change a sentence’s truth value, once known
• In abductive and inductive reasoning, hypotheses may be discarded and new ones formed when new observations are made
Default logic

• Default reasoning is another kind of non-monotonic reasoning

• We know many facts which are mostly true, typically true, or good default assumptions
  – E.g., birds can fly, dogs have four legs, etc.
Default logic

• Default reasoning is another kind of non-monotonic reasoning

• We know many facts which are mostly true, typically true, or good default assumptions
  – E.g., birds can fly, dogs have four legs, etc.

• Sometimes these facts are wrong however
  – Ostriches are birds, but can not fly
  – A dead bird can not fly
  – Uruguay President José Mujica has a 3-legged dog
Negation as Failure

• Prolog introduced the notion of *negation as failure*, which is widely used in logic programming languages and many KR systems

• Proving $P$ in classical logic can have three outcomes: true, false, unknown (+ still thinking)

• Sometimes being unable to prove something can be used as evidence that it is not true

• This is typically the case in a database context
  – Is John registered for CMSC 671?
  – If there’s no record for John in the registrar’s database, he’s not registered
Default Logic

• There are several models for default reasoning
  – All have advantages and disadvantages, supporters and detractors
• Implementations often use negatation as failure
canfly(x) :- bird(x), \+ cannotfly(X).
cannotfly(X) :- ostritch(X); dead(X).
• Autoepistemic reasoning (reasoning about what you know) is useful also
  – Does President Obama have a wooden leg?
Sources of Uncertainty

- Uncertain inputs -- missing and/or noisy data
- Uncertain knowledge
  - Multiple causes lead to multiple effects
  - Incomplete enumeration of conditions or effects
  - Incomplete knowledge of causality in the domain
  - Probabilistic/stochastic effects
- Uncertain outputs
  - Abduction and induction are inherently uncertain
  - Default reasoning, even deductive, is uncertain
  - Incomplete deductive inference may be uncertain
  - Probabilistic reasoning only gives probabilistic results (summarizes uncertainty from various sources)
Decision making with uncertainty

**Rational** behavior:

• For each possible action, identify the possible outcomes
• Compute the **probability** of each outcome
• Compute the **utility** of each outcome
• Compute the probability-weighted **(expected) utility** over possible outcomes for each action
• Select action with the highest expected utility (principle of **Maximum Expected Utility**)

- 23