Knowledge Representation and Reasoning
Chapters 12.1-12.6

Prof. Laura Zavala, laura.zavala@umbc.edu, ITE 373, 410-455-8775
Introduction

- Real knowledge representation and reasoning systems come in several major varieties.
- These differ in their intended use, expressivity, features,…
- Some major families are
  - Logic programming languages
  - Theorem provers
  - Rule-based or production systems
  - Semantic networks
  - Frame-based representation languages
  - Databases (deductive, relational, object-oriented, etc.)
  - Constraint reasoning systems
  - Description logics
  - Bayesian networks
  - Evidential reasoning
Ontologies

- Specification of a conceptualization
- Representations of concepts
- Explicit formal specifications of the terms in the domain and relations among them
- Usually represented as a type hierarchy

Diagram:

```
Vehicle
  +--- Car
  +--- Truck
     +--- 2-wheel
     +--- 4-wheel
```
A more general ontology
Different levels

Upper Ontology:
- Thing
  - Vehicle
    - Landcraft
    - Sea Vessel
    - Aircraft
      - Airplane
      - Helicopter
        - Drone
        - Airliner
        - Fighter
          - Global Hawk
          - Predator
          - Boeing 747
          - Boeing 777
          - F/A-18C
          - F/A-22

Lower Ontology:

Upper Ontologies

- Highest-level categories: typically these might include:
  - Measurements
  - Objects and their properties (including fluent, or changing, properties)
  - Events and temporal relationships
  - Continuous processes
  - Mental events, processes; “beliefs, desires, and intentions”

- Also useful:
  - Subtype relationships
  - PartOf relationships
  - Composite objects
Upper ontology

- General purpose
- Applicable in any special-purpose domain
  - Extended with more specific concepts
- Bridge independent domains
- Attempts have been made to define a universal general-purpose ontology
- Several incompatible upper ontologies that attempt to represent all knowledge exist
  - CYC
An upper ontology: CYC
Why do we need an ontology

- To share common understanding of the structure of information among people or software agents
- To enable reuse of domain knowledge
- To make domain assumptions explicit
- To separate domain knowledge from the operational knowledge
  - We can merge, extend, and change
- To analyze domain knowledge
Ontological engineering

- How do you create an ontology for a particular application?
- How do you maintain an ontology for changing needs?
- How do you merge ontologies from different fields?
- How do you map across ontologies from different fields?
Reasoning systems for categories

- Categories are the primary building blocks of large-scale knowledge representation schemes.
- Semantic networks
  - Graphical aids
  - Infer properties of objects based on category membership
- Description Logics
  - Constructing and combining categories
  - Subset and superset relationships
Semantic Networks

- Simple representation scheme that uses a graph of labeled nodes and labeled, directed arcs to encode knowledge.
  - Usually used to represent 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 were very popular in the ‘60s and ‘70s but are less frequently used today.
  - Often much less expressive than other KR formalisms
- The **graphical depiction** associated with a semantic network is a significant reason for their popularity.
**Semantic Networks: example (1)**

- SN allow representation of individual objects, categories of objects, and relations among objects.

![Semantic Network Diagram](image)

A semantic network with four objects (John, Mary, 1, and 2) and four categories. Relations are denoted by labeled links.
Nodes and Arcs

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

\[
\text{mother(john,sue)} \\
\text{age(john,5)} \\
\text{wife(sue,max)} \\
\text{age(max,34)} \\
\ldots
\]
The ISA (is-a) or AKO (a-kind-of) relation is often used to link instances to classes, classes to superclasses.

Some links (e.g. hasPart) are inherited along ISA paths.

The semantics of a semantic net can be relatively informal or very formal:
- often defined at the implementation level
Only binary relations

- In FOL we can assert
  - \(\text{Fly}(\text{Shankar}, \text{NewYork}, \text{NewDelhi}, \text{Yesterday})\)

- In semantic networks the links between nodes represent only binary relations
Semantic Networks: example (2)

- A fragment of a semantic network showing the representation of the logical assertion
  \(\text{Fly}(\text{Shankar}, \text{New York}, \text{New Delhi}, \text{Yesterday})\).
Non-binary relationships can be represented by “turning the relationship into an object”

This is an example of what logicians call “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 and those representing classes
  - the “subclass” relation from the “instance-of” relation
## Link Types

<table>
<thead>
<tr>
<th>Link Type</th>
<th>Semantics</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A \subseteq B$</td>
<td>$A \subseteq B$</td>
<td>$Cats \subseteq Mammals$</td>
</tr>
<tr>
<td>$A \in B$</td>
<td>$A \in B$</td>
<td>$Bill \in Cats$</td>
</tr>
<tr>
<td>$R(A, B)$</td>
<td>$R(A, B)$</td>
<td>$Bill \leftarrow AB \rightarrow 12$</td>
</tr>
<tr>
<td>$\forall x ; x \in A \Rightarrow R(x, B)$</td>
<td>$\forall x ; x \in A \Rightarrow R(x, B)$</td>
<td>$Birds \leftarrow Leg \rightarrow 2$</td>
</tr>
<tr>
<td>$\forall x ; \exists y ; x \in A \Rightarrow y \in B \land R(x, y)$</td>
<td>$\forall x ; \exists y ; x \in A \Rightarrow y \in B \land R(x, y)$</td>
<td>$Birds \leftarrow Parent \rightarrow Birds$</td>
</tr>
</tbody>
</table>
Inference by Inheritance

- One of the main kinds of reasoning done in a semantic net is the inheritance of values along the subclass and instance links.
- Semantic networks differ in how they handle the case of inheriting multiple different values.
  - All possible values are inherited, *or*
  - Only the “lowest” value or values are inherited
Conflicting Inherited Values
Multiple Inheritance

- A node can have any number of superclasses that contain it, enabling a node to inherit properties from multiple “parent” nodes and their ancestors in the network.

- These rules are often used to determine inheritance in such “tangled” networks where multiple inheritance is allowed:
  - If $X<A<B$ and both $A$ and $B$ have property $P$, then $X$ inherits $A$’s property.
  - If $X<A$ and $X<B$ but neither $A<B$ nor $B>Z$, and $A$ and $B$ have property $P$ with different and inconsistent values, then $X$ does not inherit property $P$ at all.
Semantic Networks vs FOL

- Reification makes it possible to represent every ground, function-free atomic sentence of FOL in semantic networks.
- Some kinds of universally quantified sentences
- We still do not have:
  - Negation, disjunction, nested function symbols, and existential quantification.
- Semantic networks main advantages
  - Simplicity, transparency, and decidability of the inference procedure
Defaults and Overriding

- John has one leg

- A person is assumed to have two legs unless that default is overridden.
- In a strictly logical KB this would be a contradiction.
- Or we could have an exception:
  - $\forall x \ x \in \text{Persons} \land x \neq \text{John} \Rightarrow \text{Legs}(x,2)$
This was the classic example circa 1980.
From Semantic Nets to Frames

- Semantic networks morphed into Frame Representation Languages in the ‘70s and ‘80s.
- A frame is a lot like the notion of an object in OOP, but has more meta-data.
- A **frame** has a set of **slots**.
- A **slot** represents a relation to another frame (or value).
- A slot has one or more **facets**.
- A **facet** represents some aspect of the relation.
Facets

- A slot in a frame holds more than a value.
- Other facets might include:
  - current fillers (e.g., values)
  - default fillers
  - minimum and maximum number of fillers
  - type restriction on fillers (usually expressed as another frame object)
  - attached procedures (if-needed, if-added, if-removed)
  - salience measure
  - attached constraints or axioms
- In some systems, the slots themselves are instances of frames.
A frame-based knowledge base

Translation into first-order logic
Description Logics

- Describe definitions and properties of categories
- Two main inference tasks
  - **subsumption** (whether categories belong within other categories)
  - **classification** (checking whether an object belongs to a category)
    - finding the right place in a hierarchy of objects for a new description
- Current systems take care to keep the languages simple, so that all inference can be done in polynomial time (in the number of objects)
  - ensuring tractability of inference
- **CLASSIC** language is a typical description logic
Description Logics (2)

- More expressive than propositional logic
- More efficient decision problems than first order predicate logic
- DL are of particular importance in providing a logical formalism for Ontologies and the Semantic Web
Semantic Web

- A group of methods and technologies to allow machines to understand the meaning - or "semantics" - of information on the World Wide Web
  - Resource Description Framework (RDF)
  - Ontologies
    - Web Ontology Language (OWL)
  - Rule Engines or Systems (Forward Chaining and Backward Chaining)
  - SPARQL is a protocol and query language for semantic web data sources
Non-monotonic Reasoning

- In normal monotonic logic, adding more sentences to a KB only entails more conclusions.
  - if KB |- P then KB U {S} |- P

- Inheritance with exceptions is not monotonic (it is nonmonotonic)
  - Bird(Opus)
  - Fly(Opus)? Yes

  - Penguin(Opus)
  - Fly(Opus)? no
Non-monotonic Reasoning

- Nonmonotonic logics attempt to formalize such reasoning by allow **default rules** of the form:
  - If P and concluding Q is consistent, then conclude Q
  - If Bird(X) then if consistent Fly(x)
Abduction is a reasoning process that tries to form plausible explanations for abnormal observations.

- Abduction is distinctly different from deduction and induction.
- Abduction is inherently uncertain.

Uncertainty is an important issue in abductive reasoning.

Some major formalisms for representing and reasoning about uncertainty:

- Mycin’s certainty factors (an early representative)
- Probability theory (esp. Bayesian belief networks)
- Dempster-Shafer theory
- Fuzzy logic
- Truth maintenance systems
- Nonmonotonic reasoning
Abduction

- **Definition** (Encyclopedia Britannica): reasoning that derives an explanatory hypothesis from a given set of facts
  - The inference result is a hypothesis that, if true, could explain the occurrence of the given facts
- **Examples**
  - Dendral, an expert system to construct 3D structure of chemical compounds
    - Fact: mass spectrometer data of the compound and its chemical formula
    - KB: chemistry, esp. strength of different types of bounds
    - Reasoning: form a hypothetical 3D structure that satisfies the chemical formula, and that would most likely produce the given mass spectrum
Abduction examples (cont.)

- Medical diagnosis
  - Facts: symptoms, lab test results, and other observed findings (called manifestations)
  - KB: causal associations between diseases and manifestations
  - Reasoning: one or more diseases whose presence would causally explain the occurrence of the given manifestations
- Many other reasoning processes (e.g., word sense disambiguation in natural language process, image understanding, criminal investigation) can also been seen as abductive reasoning
Comparing Abduction, Deduction, and Induction

**Deduction:**
- Major premise: All balls in the box are black
- Minor premise: These balls are from the box
- Conclusion: These balls are black

**Abduction:**
- Rule: All balls in the box are black
- Observation: These balls are black
- Explanation: These balls are from the box

**Induction:**
- Case: These balls are from the box
- Observation: These balls are black
- Hypothesized rule: All balls in the box are black

**Deduction** reasons from causes to effects
**Abduction** reasons from effects to causes
**Induction** reasons from specific cases to general rules
Characteristics of Abductive Reasoning

- “Conclusions” are hypotheses, not theorems (may be false even if rules and facts are true)
  - E.g., misdiagnosis in medicine

- There may be multiple plausible hypotheses
  - Given rules A => B and C => B, and fact B, both A and C are plausible hypotheses
  - Abduction is inherently uncertain
  - Hypotheses can be ranked by their plausibility (if it can be determined)
Characteristics of Abductive Reasoning (cont.)

- Reasoning is often a hypothesize-and-test cycle
  - **Hypothesize**: Postulate possible hypotheses, any of which would explain the given facts (or at least most of the important facts)
  - **Test**: Test the plausibility of all or some of these hypotheses
  - One way to test a hypothesis $H$ is to ask whether something that is currently unknown—but can be predicted from $H$—is actually true
    - If we also know $A \Rightarrow D$ and $C \Rightarrow E$, then ask if $D$ and $E$ are true
    - If $D$ is true and $E$ is false, then hypothesis $A$ becomes more plausible (**support** for $A$ is increased; **support** for $C$ is decreased)
Characteristics of Abductive Reasoning (cont.)

- Reasoning is **non-monotonic**
  - That is, the 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, some hypotheses may be discarded, and new ones formed, when new observations are made
Sources of Uncertainty

- **Uncertain inputs**
  - Missing data
  - 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 in deductive fashion, 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 the action with the highest expected utility (principle of **Maximum Expected Utility**)

---

**Note:**
- The concept of expected utility is a fundamental principle in decision theory, where the expected utility of an action is the sum of the utilities of all possible outcomes, each multiplied by the probability of that outcome.
- This approach assumes that individuals have well-defined preferences over outcomes and that these preferences can be captured by a utility function.
- The principle of Maximum Expected Utility suggests that the best action is the one that maximizes expected utility.
Bayesian Reasoning

- Probability theory
- Bayesian inference
  - Use probability theory and information about independence
  - Reason diagnostically (from evidence (effects) to conclusions (causes)) or causally (from causes to effects)
- Bayesian networks
  - Compact representation of probability distribution over a set of propositional random variables
  - Take advantage of independence relationships
Other Uncertainty Representations

- Default reasoning
  - Nonmonotonic logic: Allow the retraction of default beliefs if they prove to be false

- Rule-based methods
  - Certainty factors (Mycin): propagate simple models of belief through causal or diagnostic rules

- Evidential reasoning
  - Dempster-Shafer theory: $\text{Bel}(P)$ is a measure of the evidence for $P$; $\text{Bel}(\neg P)$ is a measure of the evidence against $P$; together they define a belief interval (lower and upper bounds on confidence)

- Fuzzy reasoning
  - Fuzzy sets: How well does an object satisfy a vague property?
  - Fuzzy logic: “How true” is a logical statement?
Uncertainty Tradeoffs

- **Bayesian networks:** Nice theoretical properties combined with efficient reasoning make BNs very popular; limited expressiveness, knowledge engineering challenges may limit uses
- **Nonmonotonic logic:** Represent commonsense reasoning, but can be computationally very expensive
- **Certainty factors:** Not semantically well founded
- **Dempster-Shafer theory:** Has nice formal properties, but can be computationally expensive, and intervals tend to grow towards [0,1] (not a very useful conclusion)
- **Fuzzy reasoning:** Semantics are unclear (fuzzy!), but has proved very useful for commercial applications