Artificial Intelligence

First-order logic
Previously

• Propositional logic
  – Simplest language
  – Its world only consists of facts (and “explicit rules”)
• Too puny a language to represent knowledge of complex environments with many objects in a concise way
  – Difficult to represent even the Wumpus world

\[ B_{1,1} \Rightarrow P_{1,2} \lor P_{2,1} \]

Would like to say, “squares adjacent to pits are breezy” (not enumerate for all possible squares)
First-Order Logic

• Also called first-order predicate calculus
  – FOL, FOPC

• Makes stronger commitments
  – World consists of objects
    • Things with identities
    • e.g., people, houses, colors, …
  – Objects have properties/relations that distinguish them from other objects
    • e.g., Properties: red, round, square, …
    • e.g., Relations: brother of, bigger than, inside, …
  – Have functional relations
    • Return the object with a certain relation to given “input” object
    • The “inverse” of a (binary) relation
    • e.g., father of, best friend
Examples of Facts as Objects and Properties or Relations

- “Squares neighboring the Wumpus are smelly”
  - Objects
    - Wumpus, squares
  - Property
    - Smelly
  - Relation
    - Neighboring
Syntax of FOL: Basic Elements

• Constant symbols for specific objects
  \textit{KingJohn}, 2, \textit{OSU}, …

• Predicate (boolean) properties (unary) / relations (binary+)
  \textit{Brother()}, \textit{Married()}, \text{>, …}

• Functions (return objects)
  \textit{Sqrt}(), \textit{LeftLegOf}(), \textit{FatherOf}(), …

• Variables
  \(x, y, a, b, \ldots\)

• Connectives
  \(\wedge \vee \neg \Rightarrow \Leftrightarrow\)

• Equality
  \(=\)

• Quantifiers
  \(\forall \exists\)
Atomic Sentences

• Collection of terms and relation(s) together to state facts

• Atomic sentence
  – \( \text{predicate}(\text{term}_1, \ldots, \text{term}_n) \)
  – Or \( \text{term}_1 = \text{term}_n \)

• Examples

  \( \text{Brother}(\text{Richard}, \text{John}) \)

  \( \text{Married}(\text{FatherOf}(\text{Richard}), \text{MotherOf}(\text{John})) \)
Complex Sentences

• Made from atomic sentences using logical connectives
  \( \neg S, \ S_1 \land S_2, \ S_1 \lor S_2, \ S_1 \Rightarrow S_2, \ S_1 \Leftrightarrow S_2 \)

Examples:

\( \text{Older}(\text{John}, 30) \Rightarrow \neg \text{Younger}(\text{John}, 30) \)
\( > (1,2) \lor \leq (1,2) \)
Quantifiers

• Currently have logic that allows objects
• Now want to express properties of entire collections of objects
  – Rather than enumerate the objects by name
• Two standard quantifiers
  – Universal $\forall$
  – Existential $\exists$
Universal Qualification

• “For all …” (typically use implication \( \Rightarrow \))
  – Allows for “rules” to be constructed

• \( \forall \ <\text{variables}> \ <\text{sentence}> \)
  – Everyone at OSU is smart
    \( \forall x \ At(x, \ OSU) \Rightarrow Smart(x) \)

• \( \forall x \ P \) is equivalent to conjunction of all instantiations of \( P \)
  \( (At(John, \ OSU) \Rightarrow Smart(John)) \)
  \( \land (At(Bob, \ OSU) \Rightarrow Smart(Bob)) \)
  \( \land (At(Mary, \ OSU) \Rightarrow Smart(Mary)) \land \ldots \)
Existential Quantification

• “There exists …” (typically use conjunction ∧)
  – Makes a statement about some object (not all)

• \( \exists \ <\text{variables}> <\text{sentences}> \)
  – Someone at OSU is smart
    \( \exists x \ At(x, OSU) \land Smart(x) \)

• \( \exists x \ P \) is equivalent to disjunction of all instantiations of \( P \)
  
  \( (At(John, OSU) \land Smart(John)) \lor (At(Bob, OSU) \land Smart(Bob)) \lor (At(Mary, OSU) \land Smart(Mary)) \lor \ldots \)

• Uniqueness quantifier
  \( \exists! \ x \) says a unique object exists (i.e. there is exactly one)
Properties of Quantifiers

• Quantifier duality: Each can be expressed using the other

\[ \forall x \ Person(x) \Rightarrow Likes(x, \ IceCream) \]

“Everybody likes ice cream”

\[ \neg \exists x \ Person(x) \land \neg Likes(x, \ IceCream) \]

“Not exist anyone who does not like ice cream”

\[ \exists x \ Person(x) \land Likes(x, \ Broccoli) \]

“Someone likes broccoli”

\[ \neg \forall x \ Person(x) \Rightarrow \neg Likes(x, \ Broccoli) \]

“Not the case that everyone does not like broccoli"
Properties of Quantifiers

• **Important relations**

\[ \exists x \ P(x) = \neg \forall x \ \neg P(x) \]
\[ \forall x \ P(x) = \neg \exists x \ \neg P(x) \]

\[ P(x) \Rightarrow Q(x) \text{ is same as } \neg P(x) \lor Q(x) \]

\[ \neg (P(x) \land Q(x)) \text{ is same as } \neg P(x) \lor \neg Q(x) \]
Proof

\[ P(x) \implies Q(x) \]

is same as

\[ \neg P(x) \lor Q(x) \]

<table>
<thead>
<tr>
<th>( P )</th>
<th>( Q )</th>
<th>( P \implies Q )</th>
</tr>
</thead>
<tbody>
<tr>
<td>False</td>
<td>False</td>
<td>TRUE</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>TRUE</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>FALSE</td>
</tr>
<tr>
<td>True</td>
<td>True</td>
<td>TRUE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( P )</th>
<th>( Q )</th>
<th>( \neg P )</th>
<th>( \neg P \lor Q )</th>
</tr>
</thead>
<tbody>
<tr>
<td>False</td>
<td>False</td>
<td>True</td>
<td>TRUE</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>True</td>
<td>TRUE</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>False</td>
<td>FALSE</td>
</tr>
<tr>
<td>True</td>
<td>True</td>
<td>False</td>
<td>TRUE</td>
</tr>
</tbody>
</table>
Proof

\[ \neg(P \land Q) \]

<table>
<thead>
<tr>
<th>( P )</th>
<th>( Q )</th>
<th>( P \land Q )</th>
<th>( \neg(P \land Q) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>False</td>
<td>False</td>
<td>False</td>
<td>TRUE</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>False</td>
<td>TRUE</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>False</td>
<td>TRUE</td>
</tr>
<tr>
<td>True</td>
<td>True</td>
<td>True</td>
<td>FALSE</td>
</tr>
</tbody>
</table>

\(-\neg(P(x) \land Q(x))\)

is same as

\(-P(x) \lor \neg Q(x)\)

\[
\begin{array}{|c|c|c|c|}
\hline
P & Q & \neg P & \neg Q & \neg P \lor \neg Q \\
\hline
False & False & True & True & TRUE \\
False & True & True & False & TRUE \\
True & False & False & True & TRUE \\
True & True & False & False & FALSE \\
\hline
\end{array}
\]
Proof

\[
\begin{array}{|c|c|c|c|}
\hline
P & Q & P \lor Q & \neg(P \lor Q) \\
\hline
\text{False} & \text{False} & \text{False} & \text{TRUE} \\
\text{False} & \text{True} & \text{True} & \text{FALSE} \\
\text{True} & \text{False} & \text{True} & \text{FALSE} \\
\text{True} & \text{True} & \text{True} & \text{FALSE} \\
\hline
\end{array}
\]

\[\neg(P(x) \lor Q(x))\]

is same as

\[\neg P(x) \land \neg Q(x)\]
Conversion Example

1. $\forall x \ Person(x) \Rightarrow Likes(x, IceCream)$

[use: $\forall x \ P(x) = \neg \exists x \ \neg P(x)$]

2. $\neg \exists x \ \neg (Person(x) \Rightarrow Likes(x, IceCream))$

[use: $P(x) \Rightarrow Q(x)$ is same as $\neg P(x) \lor Q(x)$]

3. $\neg \exists x \ \neg (\neg Person(x) \lor Likes(x, IceCream))$

[distribute negatives]

4. $\neg \exists x \ Person(x) \land \neg Likes(x, IceCream)$
Nested Quantifiers

• $\forall x \forall y$ is same as $\forall y \forall x$ \hspace{1cm} ( $\forall x,y$ )

• $\exists x \exists y$ is same as $\exists y \exists x$ \hspace{1cm} ( $\exists x,y$ )

• $\exists x \forall y$ is not same as $\forall y \exists x$
  \hspace{1cm} $\exists y \text{Person}(y) \land (\forall x \text{Person}(x) \Rightarrow \text{Loves}(x,y))$
  \hspace{1cm} “There is someone who is loved by everyone”

  $\forall x \text{Person}(x) \Rightarrow \exists y \text{Person}(y) \land \text{Loves}(x,y)$
  \hspace{1cm} “Everybody loves somebody”
  \hspace{1cm} (not guaranteed to be the same person)
Equality

• Equality symbol (=)
  – Make statements to the effect that two terms refer to the same object

  “Henry is the Father of John”
  \[ \text{Father}(John) = \text{Henry} \]

  “Spot has at least two sisters”
  \[ \exists x,y \text{ Sister}(x, \text{Spot}) \land \text{Sister}(y, \text{Spot}) \land \neg(x=y) \]
More Sentences

• “Brothers are siblings”
  \[ \forall x, y \ Brothe(x, y) \implies Sibling(x, y) \]

• “One’s mother is one’s female parent”
  \[ \forall x, y \ Mother(x, y) \implies Female(x) \land Parent(x, y) \]
Kinds of Rules

• For “Squares are breezy near a pit”
  – **Diagnostic** rule
    • Lead from observed effects to hidden causes
      – “Infer cause from effect”
      \[ \forall y \ Breezy(y) \Rightarrow \exists x \ Pit(x) \land Adjacent(x,y) \]
  – **Causal** “model-based” rule
    • Hidden world properties causes certain percepts
      – “Infer effect from cause”
      \[ \forall x,y \ Pit(x) \land Adjacent(x,y) \Rightarrow Breezy(y) \]
General Knowledge and Dealing with Categories

• Can we get further using less knowledge or more general knowledge?
• Classic example in AI is about knowledge and generality in sentences about birds and flight
Birds and Flight

• Organize facts about birds as listing of facts
  (robins fly)       (gannets fly)       (western grebes fly)
  (crows fly)       (penguins don’t fly)  (ostriches don’t fly)
  (common loons fly) (fulmars fly)       (arctic loons fly)

• Approximately 8,600 species of birds in world
  – Big list
  – Small in comparison to world population of ~100 billion birds!
Birds and Flight

• Rather than extending table, simpler to represent most facts with single symbol structure representing that birds of all species fly

\[ \forall x, s \quad \text{species}(s, \text{bird}) \land \text{inst}(x, s) \Rightarrow \text{flys}(x) \]

• Reasoning about classes
Categorization

• Categorization is very basic cognitive mechanism
  – Treat different things as equivalent
  – Respond in terms of **class membership** rather than **individuality**
  – Fundamental to cognition and knowledge engineering
  – Reasoning by classes **reduces complexity**

• Overgeneralization
  – Treating **all** birds as equivalent about questions of **flying**
  – Need to handle exceptions
    • e.g., penguins (and ostriches) do not fly!
Category Exceptions

• How many circumstances determine whether individual birds can fly?

• Minsky, AAAI-85
  – “Cooked birds can’t fly”
    • How about a cooked bird served in airline meal?
  – “Stuffed birds, frozen birds, and drowned birds cannot fly”
  – “Birds wearing concrete overcoats, and wooden bird decoys do not fly”

• Amount of special case knowledge increases

• Whether a particular bird can fly is determined by:
  – Its identity, condition, situation
Qualification Problem

• Want to separate **typical statements** from **exceptions**

• Qualification problem (McCarthy)
  – Proliferation of number of rules
  – Want to organize knowledge in general statements about **usual cases**
    • Categories are used to exploit **regularities** of the world
  – Then qualify statements describing their **exceptions**
Qualification Problem

• Abnormalcy predicates
  – Lay out qualifications for abnormal conditions
  – Example: “birds fly unless something abnormal about them”
    
    \[(\text{bird } x) \text{ and (not (ab}_1 \text{ x)) } \rightarrow (\text{flies } x)\]
  – Classes of birds that are abnormal and conditions
    
    \[(\text{disabled-bird } x) \rightarrow (\text{ab}_1 x)\]
    \[(\text{fake-bird } x) \rightarrow (\text{ab}_1 x)\]
    \[(\text{wears x concrete-overshoes}) \rightarrow (\text{disabled-bird } x)\]
    \[(\text{dead } x) \rightarrow (\text{disabled-bird } x)\]
    \[(\text{drowned } x) \rightarrow (\text{dead } x)\]
    \[(\text{stuffed } x) \rightarrow (\text{dead } x)\]
    \[(\text{cooked } x) \rightarrow (\text{dead } x)\]
    
    \[(\text{wooden-image } x) \text{ and (bird } x) \rightarrow (\text{fake-bird } x)\]
Categories and Exceptions

• Benefits of separating knowledge of typical from exceptions
  – Reduces number of sentences
  – Focus on categories of information
    • Guides introduction of new kinds of knowledge into categories to answer questions

• Basic goal to provide framework for assumptions
  – Default statements believed in absence of contradictory information
  – “Unless you know otherwise for a particular bird, assume the bird can fly”

• However, people have much richer model of what’s going on for flying
Summary

• First-order logic
  – Increased expressive power over Propositional Logic
  – Objects and relations are semantic primitives
  – Syntax: constants, functions, predicates, equality, quantifiers
    • Two standard quantifiers
      – Universal $\forall$
      – Existential $\exists$

• Dealing with categories and exceptions
  – Qualification problem