CSE 473: Logic in AI

Hanna Hajishirzi

(With slides from Luke Zettlemoyer, Dan Weld, Mausam, Stuart Russell, Dieter Fox, Henry Kautz...)
Knowledge Representation

• Represent knowledge in a manner that facilitates inference (i.e. drawing conclusions) from knowledge.

• Typically based on
  – Logic
  – Probability
  – Logic and Probability
Propositional Logic: Syntax

• **Atoms**
  - P, Q, R, ...

• **Literals**
  - P, ¬P

• **Sentences**
  - Any literal is a sentence
  - If S is a sentence
    • Then (S \land S) is a sentence
    • Then (S \lor S) is a sentence

• **Conveniences**
  P \rightarrow Q \text{ same as } \neg P \lor Q
A Knowledge Base

If the unicorn is mythical, then it is immortal, but if it is not mythical, it is a reptile. If the unicorn is either immortal or a reptile, then it is horned.

\[ (\neg R \lor H) \quad (\neg I \lor H) \]

\[ (M \lor R) \quad (\neg M \lor I) \]

M = mythical
I = immortal
R = reptile
H = horned
Wumpus World

- **Performance measure**
  - Gold: +1000, death: -1000
  - -1 per step, -10 for using the arrow

- **Environment**
  - Squares adjacent to wumpus are smelly
  - Squares adjacent to pit are breezy
  - Glitter iff gold is in the same square
  - Shooting kills wumpus if you are facing it
  - Shooting uses up the only arrow
  - Grabbing picks up gold if in same square
  - Releasing drops the gold in same square

- **Sensors**: Stench, Breeze, Glitter, Bump, Scream
- **Actuators**: Left turn, Right turn, Forward, Grab, Release, Shoot
Let $P_{i,j}$ be true if there is a pit in $[i, j]$. Let $B_{i,j}$ be true if there is a breeze in $[i, j]$.

**KB:**

- $\neg P_{1,1}$
- $\neg B_{1,1}$

"Pits cause breezes in adjacent squares"

- $B_{1,1} \Leftrightarrow (P_{1,2} \lor P_{2,1})$
- $B_{2,1} \Leftrightarrow (P_{1,1} \lor P_{2,2} \lor P_{3,1})
A Simple Knowledge Based Agent

```plaintext
function KB-AGENT( percept ) returns an action
    static: KB, a knowledge base
            t, a counter, initially 0, indicating time

        Tell( KB, Make-Percept-Sentence( percept, t ) )
        action ← Ask( KB, Make-Action-Query( t ) )
        Tell( KB, Make-Action-Sentence( action, t ) )
        t ← t + 1
    return action
```

The agent must be able to:
- Represent states, actions, etc.
- Incorporate new percepts
- Update internal representations of the world
- Deduce hidden properties of the world
- Deduce appropriate actions
### Entailment in Wumpus World

**Knowledge Base (KB):**

\[
\begin{align*}
&\neg P_{1,1}, \neg W_{1,1}, \neg B_{1,1}, \neg G_{1,1}, \\
&\neg P_{1,1}, \neg W_{1,1}, B_{1,1}, \neg G_{1,1}, \\
&\ldots \\
&B_{1,1} \iff (P_{1,2} \lor P_{2,1}) \\
&\ldots
\end{align*}
\]

Situation after detecting nothing in [1,1], moving right, breeze in [2,1]

Consider possible models for ?s assuming only pits

3 Boolean choices \( \Rightarrow \) 8 possible models
Wumpus Models

Possible assignments for the three locations which we have evidence about:

\[
\text{KB} = \{ \neg P_{1,1}, \neg W_{1,1}, \neg B_{1,1}, \neg G_{1,1}, \\
\neg P_{2,1}, \neg W_{2,1}, B_{1,1}, \neg G_{1,1}, \\
\ldots \\
B_{1,1} \iff (P_{1,2} \lor P_{2,1}) \\
\ldots \}
\]

KB is satisfiable!
Wumpus Models

Models that are consistent with our KB:

\[ KB = \{ \neg P_{1,1}, \neg W_{1,1}, \neg B_{1,1}, \neg G_{1,1}, \neg P_{1,1}, \neg W_{1,1}, B_{1,1}, \neg G_{1,1}, \ldots \} \]

\[ B_{1,1} \iff (P_{1,2} \lor P_{2,1}) \]

\[ \ldots \}

\[ KB = \text{wumpus-world rules + observations} \]
Wumpus Models

This KB does entail that [1,2] is safe:

\[
KB = \{ \neg P_{1,1}, \neg W_{1,1}, \neg B_{1,1}, \neg G_{1,1}, \neg P_{1,2}, \neg W_{1,2}, B_{1,2}, \neg G_{1,2}, \ldots \}
\]

\[
B_{1,1} \iff (P_{1,2} \lor P_{2,1})
\]

\[
\alpha_1 = \neg P_{1,2} \land \neg W_{1,2}
\]

\[
KB = \text{wumpus-world rules + observations}
\]

\[
\alpha_1 = \text{“[1,2] is safe”, } KB \models \alpha_1, \text{ proved by model checking}
\]
Wumpus Models

This KB does not entail that [2,2] is safe:

\[
\text{KB} = \{ \neg p_{1,1}, \neg w_{1,1}, \neg b_{1,1}, \neg g_{1,1}, \\
\neg p_{1,2}, \neg w_{1,2}, b_{1,2}, \neg g_{1,2}, \\
\ldots \\
\} \\
\]

\[
\alpha_2 = \neg p_{2,2} \land \neg w_{2,2} \\
\]

\[
KB = \text{wumpus-world rules} + \text{observations} \\
\alpha_2 = "[2,2] \text{ is safe}" \implies KB \not\models \alpha_2 \\
\]
Summary: Models

• Logicians often think in terms of *models*, which are formally structured worlds with respect to which truth can be evaluated
  – In propositional case, each model = truth assignment
  – Set of models can be enumerated in a truth table

• We say $m$ is a model of a sentence $\alpha$ if $\alpha$ is true in $m$

• $M(\alpha)$ is the set of all models of $\alpha$

• Entailment: $KB \models \alpha$ iff $M(KB) \subseteq M(\alpha)$
  – E.g. $KB = (P \lor Q) \land (\neg P \lor R)$
    $\alpha = (P \lor R)$

• How to check?
  – One way is to enumerate all elements in the truth table – slow 😞
Pros and Cons of Propositional Logic

- Propositional logic is *declarative*: pieces of syntax correspond to facts
- Propositional logic allows *partial/disjunctive/negated* information (unlike most data structures and databases)
- Propositional logic is *compositional*:
  - meaning of $B_{1,1} \land P_{1,2}$ derived from meanings of $B_{1,1}$ and $P_{1,2}$
- Propositional logic has very limited expressive power (unlike natural language)
  - E.g., cannot say “pits cause breezes in adjacent squares” except by writing one sentence for each square
Why First Order Logic

Propositional logic: Deals with facts and propositions (can be true or false):

- $P_{1,1}$ -- “there is a pit in (1,1)”
- George_Monkey -- “George is a monkey”
- George_Curious -- “George is curious”
- Jack_Monkey – “Jack is a monkey”
- 473student1_curious – “student 1 is a curious”
- 473student2_curious – “student 2 is a curious”
- $(George\_Monkey \land \neg 473student1\_Monkey) \lor \ldots$
FOL Definitions

**Constants:** Name a specific object.
   
   George, Monkey2, Larry, Hanna...

**Variables:** Refer to an object without naming it.
   
   X, Y, ...

**Relations (predicates):** Properties of or relationships between objects.
   
   Curious(.), PokesInTheEyes(.,.), SmarterThan(.,.)...

**Functions:** Mapping from objects to objects.
   
   banana-of(.), grade-of(.), child-of(.,.)
Syntax of First Order Logic

Constants  
KingJohn, 2, UCB, ...

Predicates  
Brother, >, ...

Functions  
Sqrt, LeftLegOf, ...

Variables  
x, y, a, b, ...

Connectives  
∧, ∨, ¬, ⇒, ⇔

Equality  
=

Quantifiers  
∀, ∃

Atomic sentence  
= predicate(term₁, ..., termₙ)
or term₁ = term₂

Term  
= function(term₁, ..., termₙ)
or constant or variable

Atomic Sentences:
E.g., Brother(KingJohn, RichardTheLionheart)
> (Length(LefLegOf(Richard)), Length(LefLegOf(KingJohn)))

Complex Sentences:
E.g. Sibling(KingJohn, Richard) ⇒ Sibling(Richard, KingJohn)
>(1, 2) ∨ ≤(1, 2)
>(1, 2) ∧ ¬>(1, 2)
Wumpus World

• **Performance measure**
  – Gold: +1000, death: -1000
  – -1 per step, -10 for using the arrow

• **Environment**
  – Squares adjacent to wumpus are smelly
  – Squares adjacent to pit are breezy
  – Glitter iff gold is in the same square
  – Shooting kills wumpus if you are facing it
  – Shooting uses up the only arrow
  – Grabbing picks up gold if in same square
  – Releasing drops the gold in same square

• **Sensors:** Stench, Breeze, Glitter, Bump, Scream
• **Actuators:** Left turn, Right turn, Forward, Grab, Release, Shoot
Wumpus World

Properties of locations:
\[ \forall x, t \quad \text{At}(\text{Agent}, x, t) \land \text{Smelt}(t) \Rightarrow \text{Smelly}(x) \]
\[ \forall x, t \quad \text{At}(\text{Agent}, x, t) \land \text{Breeze}(t) \Rightarrow \text{Breezy}(x) \]

Diagnostic rule—infer cause from effect
\[ \forall y \quad \text{Breezy}(y) \Rightarrow \exists x \quad \text{Pit}(x) \land \text{Adjacent}(x, y) \]

Causal rule—infer effect from cause
\[ \forall x, y \quad \text{Pit}(x) \land \text{Adjacent}(x, y) \Rightarrow \text{Breezy}(y) \]

Neither of these is complete—e.g., the causal rule doesn’t say whether squares far away from pits can be breezy

Definition for the *Breezy* predicate:
\[ \forall y \quad \text{Breezy}(y) \iff \exists x \quad \text{Pit}(x) \land \text{Adjacent}(x, y) \]
First Order Models

Sentences are true with respect to a model and an interpretation.

Model contains $\geq 1$ objects (domain elements) and relations among them.

Interpretation specifies referents for:
- constant symbols $\rightarrow$ objects
- predicate symbols $\rightarrow$ relations
- function symbols $\rightarrow$ functional relations

An atomic sentence $\text{predicate}(\text{term}_1, \ldots, \text{term}_n)$ is true iff the objects referred to by $\text{term}_1, \ldots, \text{term}_n$ are in the relation referred to by $\text{predicate}$.
Example: A World of Kings and Legs

• Syntactic elements:

  **Constants:**
  - Richard, John, RsLeftLeg, ...

  **Functions:**
  - leftleg(.),
  - onheadof(.), ...

  **Relations:**
  - On(.,.)
  - IsKing(.),
  - IsPerson(.), ...

---

Diagram: Two figures labeled 'R' and 'J' representing Richard and John, with relationships such as brother, left leg, on head, crown, and person.
Semantics

• Logical connectives: and, or, not, ⇒, ⇔

• Quantifiers:
  – ∀ For all (Universal quantifier)
  – ∃ There exists (Existential quantifier)

• Examples
  – George is a monkey and he is curious
    \[ \text{Monkey}(George) \land \text{Curious}(George) \]
  – All monkeys are curious
    \[ \forall m: \text{Monkey}(m) \Rightarrow \text{Curious}(m) \]
  – There is a curious monkey
    \[ \exists m: \text{Monkey}(m) \land \text{Curious}(m) \]
Quantifier / Connective Interaction

$\forall x: \ M(x) \land C(x)$  \hspace{2cm} $M(x) = \text{“}x\text{ is a monkey”}$

“Everything is a curious monkey”

$\forall x: \ M(x) \implies C(x)$  \hspace{1cm} $C(x) = \text{“}x\text{ is curious”}$

“All monkeys are curious”

$\exists x: \ M(x) \land C(x)$

“There exists a curious monkey”

$\exists x: \ M(x) \implies C(x)$

“There exists an object that is either a curious monkey, or not a monkey at all”
Fun With Sentences

• Brothers are siblings.
  \[ \forall x, y \; \text{Brother}(x, y) \Rightarrow \text{Sibling}(x, y). \]
• “Sibling” is symmetric.
  \[ \forall x, y \; \text{Sibling}(x, y) \Leftrightarrow \text{Sibling}(y, x). \]
• One’s mother is one’s female parent.
  \[ \forall x, y \; \text{Mother}(x, y) \Leftrightarrow (\text{Female}(x) \land \text{Parent}(x, y)). \]
• A first cousin is a child of a parent’s sibling.
  \[ \forall x, y \; \text{FirstCousin}(x, y) \Leftrightarrow \exists p, ps \; \text{Parent}(p, x) \land \text{Sibling}(ps, p) \land \text{Parent}(ps, y). \]
# Propositional Logic vs. First Order Logic

<table>
<thead>
<tr>
<th><strong>Ontology</strong></th>
<th>Facts (P, Q,…)</th>
<th>Objects, Properties, Relations</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Syntax</strong></td>
<td>Atomic sentences Connectives</td>
<td>Variables &amp; quantification Sentences have structure: terms father-of(mother-of(X)))</td>
</tr>
<tr>
<td><strong>Semantics</strong></td>
<td>Truth Tables</td>
<td>Interpretations &amp; Models (Much more complicated)</td>
</tr>
<tr>
<td><strong>Inference Algorithm</strong></td>
<td>DPLL, WalkSAT Fast in practice</td>
<td>Unification Forward, Backward chaining Prolog, theorem proving</td>
</tr>
<tr>
<td><strong>Complexity</strong></td>
<td>NP-Complete</td>
<td>Semi-decidable May run forever if $\text{KB} \not\models \alpha$</td>
</tr>
</tbody>
</table>