Logical agents

Chapter 6

Outline

◇ Knowledge bases
◇ Wumpus world
◇ Logic in general
◇ Propositional (Boolean) logic
◇ Normal forms
◇ Inference rules
Knowledge bases

<table>
<thead>
<tr>
<th>Inference engine</th>
<th>domain-independent algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowledge base</td>
<td>domain-specific content</td>
</tr>
</tbody>
</table>

Knowledge base = set of sentences in a formal language

**Declarative** approach to building an agent (or other system):

Tell it what it needs to know

Then it can Ask itself what to do—answers should follow from the KB

Agents can be viewed at the knowledge level

i.e., what they know, regardless of how implemented

Or at the implementation level

i.e., data structures in KB and algorithms that manipulate them

---

**A simple knowledge-based agent**

```java
// A simple knowledge-based agent

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))

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
**Wumpus World PAGE description**

**Percepts** Breeze, Glitter, Smell  
**Actions** Left turn, Right turn, Forward, Grab, Release, Shoot  
**Goals** Get gold back to start without entering pit or wumpus square  

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

---

**Wumpus world characterization**

**Is the world deterministic??**

**Is the world fully accessible??**

**Is the world static??**

**Is the world discrete??**
Wumpus world characterization

Is the world deterministic?? Yes—outcomes exactly specified
Is the world fully accessible?? No—only local perception
Is the world static?? Yes—Wumpus and Pits do not move
Is the world discrete?? Yes

Exploring a wumpus world
Other tight spots

Breeze in (1,2) and (2,1)  
⇒ no safe actions  
Assuming pits uniformly distributed,  
(2,2) is most likely to have a pit

Smell in (1,1) ⇒ cannot move  
Can use a strategy of coercion:  
shoot straight ahead  
wumpus was there ⇒ dead ⇒ safe  
wumpus wasn’t there ⇒ safe
Logic in general

Logics are formal languages for representing information such that conclusions can be drawn.

Syntax defines the sentences in the language.

Semantics define the "meaning" of sentences; i.e., define truth of a sentence in a world.

E.g., the language of arithmetic

\[ x + 2 \geq y \] is a sentence; \( x2 + y > \) is not a sentence.

\[ x + 2 \geq y \] is true iff the number \( x + 2 \) is no less than the number \( y \).

\[ x + 2 \geq y \] is true in a world where \( x = 7, y = 1 \).

\[ x + 2 \geq y \] is false in a world where \( x = 0, y = 6 \).

Types of logic

Logics are characterized by what they commit to as "primitives".


Epistemological commitment: what states of knowledge?

<table>
<thead>
<tr>
<th>Language</th>
<th>Ontological Commitment</th>
<th>Epistem. Comm.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Propositional logic</td>
<td>facts</td>
<td>true/false/unknown</td>
</tr>
<tr>
<td>First-order logic</td>
<td>facts, objects, relations</td>
<td>true/false/unknown</td>
</tr>
<tr>
<td>Temporal logic</td>
<td>facts, objects, relations, times</td>
<td>true/false/unknown</td>
</tr>
<tr>
<td>Probability theory</td>
<td>facts</td>
<td>degree of belief 0...1</td>
</tr>
<tr>
<td>Fuzzy logic</td>
<td>degree of truth</td>
<td>degree of belief 0...1</td>
</tr>
</tbody>
</table>
**Entailment**

\[ KB \models \alpha \]

Knowledge base \( KB \) entails sentence \( \alpha \) if and only if \( \alpha \) is true in all worlds where \( KB \) is true.

E.g., the KB containing “the Giants won” and “the Reds won” entails “Either the Giants won or the Reds won.”

**Models**

Logicians typically think in terms of models, which are formally structured worlds with respect to which truth can be evaluated.

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 \( \alpha \).

Then \( KB \models \alpha \) if and only if \( M(KB) \subseteq M(\alpha) \).

E.g. \( KB = \) Giants won and Reds won
\[ \alpha = \text{Giants won} \]
Inference

$KB \vdash_i \alpha = \text{sentence } \alpha \text{ can be derived from } KB \text{ by procedure } i$

Soundness: $i$ is sound if

whenever $KB \vdash_i \alpha$, it is also true that $KB \models \alpha$

Completeness: $i$ is complete if

whenever $KB \models \alpha$, it is also true that $KB \vdash_i \alpha$

Preview: we will define a logic (first-order logic) which is expressive enough to say almost anything of interest, and for which there exists a sound and complete inference procedure.

That is, the procedure will answer any question whose answer follows from what is known by the $KB$.

Propositional logic: Syntax

Propositional logic is the simplest logic—illustrates basic ideas

The proposition symbols $P_1$, $P_2$ etc are sentences

If $S$ is a sentence, $\neg S$ is a sentence

If $S_1$ and $S_2$ is a sentence, $S_1 \land S_2$ is a sentence

If $S_1$ and $S_2$ is a sentence, $S_1 \lor S_2$ is a sentence

If $S_1$ and $S_2$ is a sentence, $S_1 \Rightarrow S_2$ is a sentence

If $S_1$ and $S_2$ is a sentence, $S_1 \Leftrightarrow S_2$ is a sentence
Propositional logic: Semantics

Each model specifies true/false for each proposition symbol.

E.g. $A \quad B \quad C$

| True | True | False |

Rules for evaluating truth with respect to a model $m$:

- $\neg S$ is true iff $S$ is false,
- $S_1 \land S_2$ is true iff $S_1$ is true and $S_2$ is true,
- $S_1 \lor S_2$ is true iff $S_1$ is true or $S_2$ is true,
- $S_1 \Rightarrow S_2$ is true iff $S_1$ is false or $S_2$ is true,
- $S_1 \Leftrightarrow S_2$ is true iff $S_1 \Rightarrow S_2$ is true and $S_2 \Rightarrow S_1$ is true.

Propositional inference: Enumeration method

Let $\alpha = A \lor B$ and $KB = (A \lor C) \land (B \lor \neg C)$

Is it the case that $KB \models \alpha$?

Check all possible models—$\alpha$ must be true wherever $KB$ is true.

<table>
<thead>
<tr>
<th>$A$</th>
<th>$B$</th>
<th>$C$</th>
<th>$A \lor C$</th>
<th>$B \lor \neg C$</th>
<th>$KB$</th>
<th>$\alpha$</th>
</tr>
</thead>
<tbody>
<tr>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>True</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>True</td>
<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>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
</tbody>
</table>
**Propositional inference: Solution**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>A ∨ C</th>
<th>B ∨ ¬C</th>
<th>KB</th>
<th>α</th>
</tr>
</thead>
<tbody>
<tr>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>True</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>True</td>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>False</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
</tbody>
</table>

**Normal forms**

Other approaches to inference use syntactic operations on sentences, often expressed in standardized forms.

**Conjunctive Normal Form (CNF—universal)**

- conjunction of disjunctions of literals
  
  E.g., \((A ∨ ¬B) ∧ (B ∨ ¬C ∨ ¬D)\)

**Disjunctive Normal Form (DNF—universal)**

- disjunction of conjunctions of literals
  
  E.g., \((A ∧ B) ∨ (A ∧ ¬C) ∨ (A ∧ ¬D) ∨ (¬B ∧ ¬C) ∨ (¬B ∧ ¬D)\)
Horn Form (restricted)

conjunction of Horn clauses (clauses with $\leq 1$ positive literal)

E.g., $(A \lor \neg B) \land (B \lor \neg C \lor \neg D)$

Often written as set of implications:

$B \Rightarrow A$ and $(C \land D) \Rightarrow B$

Validity and Satisfiability

A sentence is valid if it is true in all models

e.g., $A \lor \neg A$, $A \Rightarrow A$, $(A \land (A \Rightarrow B)) \Rightarrow B$

Validity is connected to inference via the Deduction Theorem:

$KB \models \alpha$ if and only if $(KB \Rightarrow \alpha)$ is valid

A sentence is satisfiable if it is true in some model

e.g., $A \lor B$, $C$

A sentence is unsatisfiable if it is true in no models

e.g., $A \land \neg A$

Satisfiability is connected to inference via the following:

$KB \models \alpha$ if and only if $(KB \land \neg \alpha)$ is unsatisfiable

i.e., prove $\alpha$ by *reductio ad absurdum*
Proof methods

Proof methods divide into (roughly) two kinds:

Model checking
- truth table enumeration (sound and complete for propositional)
- heuristic search in model space (sound but incomplete)
  e.g., the GSAT algorithm (Ex. 6.15)

Application of inference rules
- Legitimate (sound) generation of new sentences from old
- Proof = a sequence of inference rule applications
  Can use inference rules as operators in a standard search alg.

Inference rules for propositional logic

Resolution (for CNF): complete for propositional logic
\[ \alpha \lor \beta, \quad \neg \beta \lor \gamma \quad \Rightarrow \quad \alpha \lor \gamma \]

Modus Ponens (for Horn Form): complete for Horn KBs
\[ \alpha_1, \ldots, \alpha_n, \quad \alpha_1 \land \cdots \land \alpha_n \Rightarrow \beta \quad \Rightarrow \quad \beta \]

Can be used with forward chaining or backward chaining
Summary

Logical agents apply inference to a knowledge base to derive new information and make decisions.

Basic concepts of logic:
- syntax: formal structure of sentences
- semantics: truth of sentences wrt models
- entailment: necessary truth of one sentence given another
- inference: deriving sentences from other sentences
- soundness: derivations produce only entailed sentences
- completeness: derivations can produce all entailed sentences

Wumpus world requires the ability to represent partial and negated information, reason by cases, etc.

Propositional logic suffices for some of these tasks.

Truth table method is sound and complete for propositional logic.