Lecture 9: Knowledge-Based Agents and Propositional Logic
Reading: AIAMA 7.1-7.4

Today’s Schedule:
- Introduce knowledge-based agents
- Wumpus World
- Propositional logic
- Introduce Theorem Proving
Knowledge-Based Agents

- Real World
  - Percepts
  - Actions

- Agent
  - State Space Model
Knowledge Base Agents

- A knowledge base (KB) is a set of sentences. These sentences are expressed in a formal knowledge representation language.
- To add sentences to the KB you TELL it a sentence.
- To query the KB you ASK it a sentence.
- TELL and ASK may use logical inference internally.

Example: Lets say the KB is initially empty and we
TELL(RAINING)
TELL(IF RAINING THEN PAVEMENT_WET)
What do you expect ASK(PAVEMENT_WET) to return?
- So what representation language should we use?
The goal of Logic Theorist was to take the five axioms in Whitehead and Russel’s *Principia Mathematica* and prove all the theorem’s inside. The axioms

1. \((p \lor p) \implies p\)
2. \(p \implies (q \lor p)\)
3. \((p \lor q) \implies (q \lor p)\)
4. \([p \lor (q \lor r)] \implies [q \lor (p \lor r)]\)
5. \((p \implies q) \implies [r \lor p) \implies (r \lor q)]\)

Two Rules

1. Substitution: a variable may be substituted by an expression
2. Replacement: \(p \implies q\) can be replaced by \(\neg p \lor q\)
Example Proof

Given the axioms

1. \((p \lor p) \implies p\)
2. \(p \implies (q \lor p)\)
3. \((p \lor q) \implies (q \lor p)\)
4. \([p \lor (q \lor r)] \implies [q \lor (p \lor r)]\)
5. \((p \implies q) \implies [r \lor p] \implies (r \lor q)\)

Prove \((p \implies \neg p) \implies \neg p\)
Review of Logic Terminology and Concepts

- syntax vs semantics
- truth and possible worlds (models)
- satisfaction
- entailment
- logical inference
- model checking
- sound (truth preserving) inference
- complete inference
- grounding
The definition of entailment is that $\alpha \models \beta$ if and only if $M(\alpha) \subseteq M(\beta)$. Some questions:

- Why are the models written as sets?
- Is it possible to write down a truth table for entailment?
Warmup #1

Given the current state of the WW below, what is the percept expected and what would it be for all possible (single) moves from this state?
Example Reasoning in the Wumpus World

```
4  Stench  Breeze  PIT
3  Breeze  Stench  PIT  Breeze
2  Stench  Breeze
1  Breeze  PIT  Breeze
```

START

```
1  2  3  4
```

---

**Note:**
- **Stench** indicates a pit that emits a stench.
- **Breeze** indicates a pit that is not emitting a stench.
- **PIT** indicates a pit that is emitting a stench.

---

**Reasoning:**
- At position 1, the player finds a **PIT** and a **Breeze**.
- At position 2, the player finds another **PIT** and a **Breeze**.
- At position 3, the player finds a **PIT** and a **Breeze**.
- At position 4, the player finds a **Stench** and a **Breeze**.

---

**Conclusion:**
- The player should avoid positions 1, 2, and 3 as they are pits.
- The player should continue to explore since position 4 has a stench.

---

**Additional Notes:**
- The Wumpus World is a classic example of a logical reasoning problem in artificial intelligence, where the player must deduce the location of the pits and the wumpus (a monster) using sensory information such as stench and breeze.
Propositional Logic

The simplest knowledge representation language is the propositional logic (PL) (digital logic)

- The KB is made up of conjunctions of atomic or complex sentences
- Atomic symbols are True, False, strings
- The syntax of PL is defined by those atoms and 5 operators: \( \neg, \land, \lor, \rightarrow, \iff \)
- Semantics are established by an assignment of True or False to every symbol, often using truth tables for dependent symbols
Syntax of PL

\[
\begin{align*}
    \text{Sentence} & \rightarrow \text{AtomicSentence} | \text{ComplexSentence} \\
    \text{AtomicSentence} & \rightarrow \text{True} | \text{False} | P | Q | R | \ldots \\
    \text{ComplexSentence} & \rightarrow (\text{Sentence}) | [\text{Sentence}] \\
                        & | \neg \text{Sentence} \\
                        & | \text{Sentence} \land \text{Sentence} \\
                        & | \text{Sentence} \lor \text{Sentence} \\
                        & | \text{Sentence} \Rightarrow \text{Sentence} \\
                        & | \text{Sentence} \Leftrightarrow \text{Sentence}
\end{align*}
\]

**Operator Precedence**: $\neg, \land, \lor, \Rightarrow, \Leftrightarrow$

**Figure 7.7** A BNF (Backus–Naur Form) grammar of sentences in propositional logic, along with operator precedences, from highest to lowest.
Example of PL Semantics

Consider a world made up of blocks on a table and a robot to rearrange them

How might we define the syntax in PL? How would we establish the semantics? What do we do about time?
Syntax is a relatively easy concept compared to Semantics. Why?
Exercise

Consider the logic required to represent the game rocks-paper-scissors.

How might we define the syntax in PL?
How would we establish the semantics?
Consider a simple KB

\[ KB \triangleq (A \land B) \implies C \]

with semantics specified by a truth table following the rules of conjunction and implication.

Suppose the query is \( \alpha \triangleq C \), can we infer the query?

Suppose we TELL(\( \neg A \)), and query again?

Suppose instead we TELL(\( A \)), and query again?

Suppose instead we TELL(\( A \)), TELL(\( B \)) and query again?
Inference by model checking

\textbf{function} \textsc{TT-Entails?}(KB, \alpha) \textbf{returns} true or false
\begin{itemize}
  \item \textbf{inputs:} KB, the knowledge base, a sentence in propositional logic \\
       \alpha, the query, a sentence in propositional logic
\end{itemize}
\begin{align*}
  \textit{symbols} & \leftarrow \text{a list of the proposition symbols in } \text{KB and } \alpha \\
  \textbf{return} \, \textsc{TT-Check-All}(KB, \alpha, \text{symbols}, \{\})
\end{align*}

\textbf{function} \textsc{TT-Check-All}(KB, \alpha, \text{symbols}, \text{model}) \textbf{returns} true or false
\begin{itemize}
  \item \textbf{if} \textsc{Empty?}(\text{symbols}) \textbf{then}
    \begin{itemize}
      \item \textbf{if} \textsc{PL-True?}(KB, \text{model}) \textbf{then} \textbf{return} \textsc{PL-True?}(\alpha, \text{model})
      \item \textbf{else} \textbf{return} true \text{ // when } \text{KB} \text{ is false, always return true}
    \end{itemize}
  \item \textbf{else} \textbf{do}
    \begin{itemize}
      \item \textit{P} \leftarrow \textsc{First}(\text{symbols})
      \item \textit{rest} \leftarrow \textsc{Rest}(\text{symbols})
      \item \textbf{return} \textsc{TT-Check-All}(KB, \alpha, \text{rest}, \text{model} \cup \{P = \text{true}\}) \text{ and } \textsc{TT-Check-All}(KB, \alpha, \text{rest}, \text{model} \cup \{P = \text{false}\})
    \end{itemize}
\end{itemize}

\textbf{Figure 7.8} A truth-table enumeration algorithm for deciding propositional entailment. (TT stands for truth table.) \textsc{PL-True?} returns \textit{true} if a sentence holds within a model. The variable \textit{model} represents a partial model—an assignment to some of the symbols. The keyword “and” is used here as a logical operation on its two arguments, returning \textit{true} or \textit{false}. 
Consider a KB

\[ KB \triangleq (A \implies B) \land (B \implies A) \]

with semantics specified by a truth table following the rules of conjunction, implication, and equivalence.

Using model checking can you infer the query \[ \alpha \triangleq A \iff B \]
Next Actions

- Reading on Theorem Proving, AIAMA 7.5
- Take warmup before noon on Thursday 2/19.

Announcements:

- Problem Set 2 released - Due Monday 3/23 by 8am
- Quiz 1 will be on Tuesday 2/24