Lecture 19: Logic for KR
Overview

• Last time
  – Expert Systems and Ontologies

• Today
  – Logic as a knowledge representation scheme
  – Propositional Logic
    • Syntax
    • Semantics
    • Proof theory
      – Natural deduction

• Learning outcomes covered today:

Distinguish the characteristics, and advantages and disadvantages, of the major knowledge representation paradigms that have been used in AI, such as production rules, semantic networks, propositional logic and first-order logic;

Solve simple knowledge-based problems using the AI representations studied;
Introduction

- We have considered a number of forms of knowledge representation
- Despite some of their advantages, all suffer from the lack of a well-defined semantics
- Logic is a method of KR which does have a well-defined semantics
- Rules and structured objects were meant to correspond to the way people store knowledge
  - Thinking humanly
- Logic provides the paradigm for thinking rationally
- Current AI is more concerned with describing rationality than pragmatic replication of human behaviour
Knowledge-Based Agents

• Knowledge base = set of sentences in a formal language

• **Declarative** approach to building an agent
  – 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
Knowledge-Based Agents

• 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
Logic in General

• **Logics**: formal languages for representing information such that conclusions can be drawn

• Common logics: *propositional* or *first-order predicate* logic

• However there are many other logics, e.g. modal logics, temporal logics, description logics, ...

• A *logic* usually has a well-defined *syntax*, *semantics* and *proof theory*

• The *syntax* of a logic defines the syntactically acceptable objects of the logic, or *well-formed formulae*

• The *semantics* of a logic associate each formula with a *meaning*

• The *proof theory* is concerned with manipulating formulae according to certain rules
Propositional Logic

• The syntax of propositional logic is constructed from propositions and connectives.
• A proposition is a statement that is either true or false but not both.
• Propositions may be combined with other propositions to form compound propositions. These in turn may be combined into further propositions.

• The connectives that may be used are
  - $\top$ true
  - $\perp$ false
  - $\land$ and conjunction ($\&$ or $.$)
  - $\lor$ or disjunction ($|$ or $+$)
  - $\neg$ not negation ($\sim$)
  - $\Rightarrow$ if . . . then implication ($\rightarrow$)
  - $\Leftrightarrow$ if and only if equivalence ($\leftrightarrow$)

• Some books use different notations, as indicated by the alternative symbols given in parentheses.
Well-Formed Formulae

• The set of sentences or well-formed propositional formulae (WFF) is defined as:
  – Any propositional symbol is in WFF.
  – The nullary connectives, true and false are in WFF.
  – If A and B are in WFF then so is ¬A, A∨B, A∧B, A→B and A↔B.
  – If A is in WFF then so is (A).

So, e.g. ((A∨B) ∧ (P∨B)) ⇒ ¬Q
Propositional Logic Semantics

• Propositions can be true or false. Formally:
  Let \( I : PROP \rightarrow \{T, F\} \) be an interpretation which assigns a truth value to each atomic proposition.

  E.g. \( P \quad Q \quad R \)
  \[
  \begin{array}{ccc}
  \text{true} & \text{true} & \text{false} \\
  \end{array}
  \]

• Rules for evaluating truth with respect to an interpretation \( I \) are determined by truth tables.

• If a compound proposition is true for ALL values of the propositions it contains, it is a tautology, and is logically true.

• If a compound proposition is false for ALL values of the propositions it contains, it is a contradiction, and is logically false.
Truth Tables

- We can summarise the operation of the connectives using truth tables.
- Rows in the table give all possible setting of the propositions to true (T) or false (F)

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>¬p</th>
<th>p∧q</th>
<th>p∨q</th>
<th>p⇒q</th>
<th>p⇔q</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>
Exercise

• Construct the truth table for the following formula and state whether the formula is a tautology, a contradiction or neither (a contingency):

\[(p \Rightarrow q) \lor (q \Rightarrow p)\]
<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>((p \Rightarrow q))</th>
<th>((q \Rightarrow p))</th>
<th>((p \Rightarrow q) \lor (q \Rightarrow p))</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>
Back to Knowledge Representation

- We are interested in a computer-suitable language to
  - represent explicit knowledge
  - reason

<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*
  - Clear syntax and *semantics*
  - Adequate (in many aspects)
  - Natural
Entailment

• *Entailment* means that one thing follows from another:
  \[ \text{KB} \models \alpha \]

• 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 Rangers won” entails “Either the Giants won or the Rangers won”

• E.g., \( x + y = 4 \) entails \( 4 = x + y \)

• Entailment is a relationship between sentences (i.e., *syntax*) that is based on *semantics*.
Propositional Logic Example (I)

\[ \text{alarm} \_ \text{beeps} \land \text{hot} \land (\text{hot} \land \text{smoky} \rightarrow \text{fire}) \land (\text{alarm} \_ \text{beeps} \rightarrow \text{smoky}) \land (\text{fire} \rightarrow \text{switch} \_ \text{on} \_ \text{sprinklers}) \]

\[ \models \text{switch} \_ \text{on} \_ \text{sprinklers} \]
Propositional Logic Example (II)

\[
\begin{align*}
(hot \land \text{smoky} \Rightarrow \text{fire}) \land (\text{alarm\_beeps} \Rightarrow \text{smoky}) \land (\text{fire} \Rightarrow \text{switch\_on\_sprinklers})
\end{align*}
\]

\[
\text{alarm\_beeps} \land \text{hot} \Rightarrow \text{switch\_on\_sprinklers}
\]
Propositional Logic Example (III)

\[(\text{hot} \land \text{smoky} \Rightarrow \text{fire}) \land (\text{alarm_beeps} \Rightarrow \text{smoky}) \land (\text{fire} \Rightarrow \text{switch_on_sprinklers})\]

\[\vdash \neg\text{switch_on_sprinklers} \Rightarrow \neg\text{fire}\]
Propositional Logic Example (IV)

\[(\text{hot} \land \text{smoky} \Rightarrow \text{fire}) \land (\text{alarm_beeps} \Rightarrow \text{smoky}) \land (\text{fire} \Rightarrow \text{switch_on_sprinklers})\]

\[\neg \text{switch_on_sprinklers} \land \text{hot} \Rightarrow \neg \text{smoky}\]
Propositional Logic for KR

• Describe what we know about a particular domain by a propositional formula, KB

• Formulate a hypothesis, \( \alpha \)

• We want to know whether KB implies \( \alpha \)

• Given a knowledge base KB and a property \( \alpha \), check if \( \text{KB} \vdash \alpha \)
  – Use truth tables
  – Prove \( \alpha \) from KB
  – Relate with \textit{validity} and \textit{satisfiability}
  – Davis-Putnam algorithm
Entailment Test

• How do we know that KB ╞ α?
• Models
• Inference
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 $m$

• Each line on a truth table that evaluates to true is a model for the formula.

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

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

  e.g. $KB = \text{Giants won and Rangers won}$
  $\alpha = \text{Giants won}$
Example

\[(\text{hot} \land \text{smoky} \Rightarrow \text{fire}) \land (\text{alarm\_beeps} \Rightarrow \text{smoky}) \land (\text{fire} \Rightarrow \text{switch\_on\_sprinklers}) \Rightarrow \neg \text{switch\_on\_sprinklers} \Rightarrow \neg \text{fire}\]

Abbreviations:

\textbf{Hot}, \textbf{Smoky}, \textbf{Fire}, \textbf{Alarm\_beeps}, \textbf{Switch\_on\_sprinklers}

\[((\text{H}\land\text{S}\Rightarrow\text{F})\land(\text{A}\Rightarrow\text{S})\land(\text{F}\Rightarrow\text{W})) \Rightarrow (\neg\text{W}\Rightarrow\neg\text{F})\]
**Truth Table**

...gives a truth table for all possible interpretations.

<table>
<thead>
<tr>
<th>H</th>
<th>S</th>
<th>F</th>
<th>A</th>
<th>W</th>
<th>((H \land S \Rightarrow F) \land (A \Rightarrow S) \land (F \Rightarrow W))</th>
<th>(\neg W \Rightarrow \neg F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Inference

• KB ⊢_i α
  Reads: sentence α can be derived from KB by procedure _i

  • Soundness: _i is sound if
    whenever KB ⊢_i α, it is also true that KB ⊨ α

  • Completeness: _i is complete if
    whenever KB ⊨ α, it is also true that KB ⊢_i α

  • That is, the procedure will answer any question whose answer follows from what is known by the KB.
Inference Example and Proof Rules

\[
\begin{array}{c}
\text{fire} \quad \text{fire} \Rightarrow \text{switch\_on\_sprinklers} \\
\hline
\text{switch\_on\_sprinklers}
\end{array}
\]

- Stating that B follows (or is provable) from \(A_1, \ldots, A_n\) can be written

\[
A_1, \ldots, A_n \\
\hline
B
\]

25
Some Proof Rules

• *Modus ponens* is a well known proof rule: \[ A \Rightarrow B, \ A \quad \Rightarrow \quad B \]
  where A and B are any WFF.

• Another common proof rule, is *∧*-elimination: \[ A \quad \land \quad B \quad \land \quad B \quad \land \quad B \]
  \[ A \quad B \quad A \quad B \]
  Reads: *if A and B hold (or are provable or true) then A (resp. B) must also hold.*

• Another proof rule, is *∨*-introduction is: \[ A \quad A \quad A \quad A \quad A \quad A \]
  \[ A \quad v \quad B \quad v \quad A \quad v \quad A \]
  Reads: *if A holds (or is provable or true) then A ∨ B must also hold.*
Natural Deduction Example

• From \( r \land s \) and \( s \Rightarrow p \) can we prove \( p \), i.e. show \( r \land s, s \Rightarrow p \vdash p \)?

1. \( r \land s \) [Given]
2. \( s \Rightarrow p \) [Given]
3. \( s \) [1 \( \land \)-elimination] \( r \land s \)
   \[ \frac{r \land s}{s} \]
4. \( p \) [2,3 modus ponens] \( s \Rightarrow p, s \)
   \[ \frac{s \Rightarrow p, s}{p} \]
Proof Theory

• Reasoning about statements of the logic without considering interpretations is known as proof theory.

• **Proof rules** (or inference rules) show us, given true statements how to generate further true statements.

• **Axioms** describe ‘universal truths’ of the logic.
  – Example \( \vdash p \lor \neg p \) is an axiom of propositional logic.

• We use the symbol \( \vdash \) to denote *is provable* or *is true*.

• We write \( A_1, \ldots, A_n \vdash B \) to show that \( B \) is provable from \( A_1, \ldots, A_n \) (given some set of inference rules).
Proofs

• Let $A_1, \ldots, A_m, B$ be well-formed formulae.

• There is a proof of $B$ from $A_1, \ldots, A_m$ iff there exists some sequence of formulae $C_1, \ldots, C_n$ such that $C_n = B$, and each formula $C_k$, for $1 \leq k < n$ is either an axiom or one of the formulae $A_1, \ldots, A_m$, or else is the conclusion of a rule whose premises appeared earlier in the sequence.
Example

• From $p \Rightarrow q, (\neg r \lor q) \Rightarrow (s \lor p), q$ can we prove $s \lor q$?
  1. $p \Rightarrow q$ [Given]
  2. $(\neg r \lor q) \Rightarrow (s \lor p)$ [Given]
  3. $q$ [Given]
  4. $s \lor q$ [3, $\lor$-introduction]

• Think how much work we would have had to do to construct a truth table to show

$$(((p \Rightarrow q) \land ((\neg r \lor q) \Rightarrow (s \lor p)) \land q) \models (s \lor q))$$
Exercise

• Show \( r \) from \( p \Rightarrow (q \Rightarrow r) \) and \( p \land q \) using the rules we have seen so far. That is, prove

\[ p \Rightarrow (q \Rightarrow r), \ p \land q \vdash r \]
Soundness and Completeness

• Let $A_1, \ldots, A_n, B$ be well-formed formulae and let
  $A_1, \ldots, A_n \vdash B$
  denote that $B$ is derivable from $A_1, \ldots, A_n$.

• Informally, soundness involves ensuring our proof system gives the *correct* answers.
  – **Theorem** (Soundness): If $A_1, \ldots, A_n \vdash B$ then
    $A_1 \land \ldots \land A_n \models B$

• Informally, completeness involves ensuring that *all* formulae that should be able to be proved can be.
  – **Theorem** (Completeness): If $A_1 \land \ldots \land A_n \models B$ then
    $A_1, \ldots, A_n \vdash B$
More on Soundness and Completeness

• Example: An unsound (bad) inference rule is: \[ A, B \quad C \]

Using this rule, from any \( p \) and \( q \) we could derive \( r \) yet \( p \land q \models r \) does not hold.

• The set of rules modus ponens and \( \land \)-elimination is incomplete: without \( \lor \)-introduction we cannot do the proof on slide 28, yet

\[
((p \Rightarrow q) \land (\neg r \lor q) \Rightarrow (s \lor p)) \land q \models (s \lor q)
\]
Summary

• We have had a brief recap of the syntax and semantics of propositional logic
• We have discussed proof (inference) rules and axioms, but have not seen the full set (- see books covering Natural Deduction in logic)
• We have seen some example proofs
• Note, at any step in the proof there may be many rules which could be applied; may need to apply search techniques, heuristics or strategies to find a proof
• Getting computers to perform proof is an area of AI itself known as automated reasoning

• Next time
  – We will look at how we can automate deduction