COMP219: Artificial Intelligence

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
  ⊤ **true**
  ⊥ **false**
  ∧ and conjunction (& or .)
  ∨ or disjunction (| or +)
  ¬ not negation (∼)
  ⇒ if . . . then implication (→)
  ⇔ if and only if equivalence (↔)

- 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 $\neg A$, $A \lor B$, $A \land B$, $A \Rightarrow B$ and $A \Leftrightarrow B$.
  - If A is in WFF then so is (A).

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

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	Q	R
true	*true*	*false*

- 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)

p	q	$\neg p$	$p \wedge q$	$p \vee q$	$p \Rightarrow q$	$p \Leftrightarrow q$
T	T	F	T	T	T	T
T	F	F	F	T	F	F
F	T	T	F	T	T	F
F	F	T	F	F	T	T

Exercise

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

$(p \Rightarrow q) \vee (q \Rightarrow p)$

Exercise

p	q	$(p \Rightarrow q)$	$(q \Rightarrow p)$	$(p \Rightarrow q) \lor (q \Rightarrow p)$
T	T	T	T	T
T	F	F	T	T
F	T	T	F	T
F	F	T	T	T

Back to Knowledge Representation

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

Inference engine	← Domain-independent algorithms
Knowledge base	← Domain-specific content

- 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:

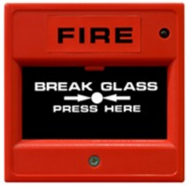
$$KB \models \alpha$$

- Knowledge base KB entails sentence α if and only if α 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*.



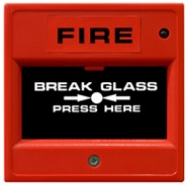
Propositional Logic Example (I)

$$\left.\begin{array}{l} \texttt{alarm\_beeps} \wedge \texttt{hot} \\ \wedge(\texttt{hot} \wedge \texttt{smoky} \Rightarrow \texttt{fire}) \\ \wedge(\texttt{alarm\_beeps} \Rightarrow \texttt{smoky}) \\ \wedge(\texttt{fire} \Rightarrow \texttt{switch\_on\_sprinklers}) \end{array}\right\} \models \texttt{switch\_on\_sprinklers}$$

Propositional Logic Example (II)

$$\left.\begin{array}{l}(\texttt{hot} \land \texttt{smoky} \Rightarrow \texttt{fire})\\ \land(\texttt{alarm\_beeps} \Rightarrow \texttt{smoky})\\ \land(\texttt{fire} \Rightarrow \texttt{switch\_on\_sprinklers})\end{array}\right\} \models$$

$$\texttt{alarm\_beeps} \land \texttt{hot} \Rightarrow \texttt{switch\_on\_sprinklers}$$



Propositional Logic Example (III)

$$\left.\begin{array}{l}(\mathtt{hot} \wedge \mathtt{smoky} \Rightarrow \mathtt{fire}) \\ \wedge(\mathtt{alarm\_beeps} \Rightarrow \mathtt{smoky}) \\ \wedge(\mathtt{fire} \Rightarrow \mathtt{switch\_on\_sprinklers})\end{array}\right\} \models$$

$$\neg\mathtt{switch\_on\_sprinklers} \Rightarrow \neg\mathtt{fire}$$

Propositional Logic Example (IV)

$$\left.\begin{array}{l} (\mathtt{hot} \wedge \mathtt{smoky} \Rightarrow \mathtt{fire}) \\ \wedge(\mathtt{alarm\_beeps} \Rightarrow \mathtt{smoky}) \\ \wedge(\mathtt{fire} \Rightarrow \mathtt{switch\_on\_sprinklers}) \end{array}\right\} \models$$

$$\neg \mathtt{switch\_on\_sprinklers} \wedge \mathtt{hot} \Rightarrow \neg \mathtt{smoky}$$

Propositional Logic for KR

- Describe what we know about a particular domain by a propositional formula, KB
- Formulate a hypothesis, α
- We want to know whether KB implies α

- Given a knowledge base KB and a property α, check if KB $\models$ α
  - Use truth tables
  - Prove α from KB
  - Relate with *validity* and *satisfiability*
  - Davis-Putnam algorithm

Entailment Test

- How do we know that KB $\models \alpha$?
- 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 α if α 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 α
- Then KB $\models \alpha$ if and only if $M(KB) \subseteq M(\alpha)$

  e.g. KB = Giants won and Rangers won

  α = Giants won

Example

$$\left.\begin{array}{l}(\texttt{hot} \wedge \texttt{smoky} \Rightarrow \texttt{fire})\\ \wedge(\texttt{alarm\_beeps} \Rightarrow \texttt{smoky})\\ \wedge(\texttt{fire} \Rightarrow \texttt{switch\_on\_sprinklers})\end{array}\right\} ?\models$$

$$\neg\texttt{switch\_on\_sprinklers} \Rightarrow \neg\texttt{fire}$$

Abbreviations:

Hot, **S**moky, **F**ire, **A**larm_beeps, s**W**itch_on_sprinklers

$$((H \wedge S \Rightarrow F) \wedge (A \Rightarrow S) \wedge (F \Rightarrow W)) ?\models (\neg W \Rightarrow \neg F)$$

Truth Table

…gives a truth table for all possible interpretations.

H	S	F	A	W	$((H \land S \Rightarrow F) \land (A \Rightarrow S) \land (F \Rightarrow W))$	$\neg W \Rightarrow \neg F$
T	T	T	T	T	T	T
T	T	T	T	F	F	F
T	T	T	F	T	T	T
T	T	T	F	F	F	F
T	T	F	T	T	F	T
T	T	F	T	F	F	T
…	…	…	…	…	…	…

Inference

- KB $\vdash_i$ α

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

- *Soundness: $_i$ is sound if*

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

- *Completeness: $_i$ is complete if*

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

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

Inference Example and Proof Rules

$$\frac{\texttt{fire} \qquad \texttt{fire} \Rightarrow \texttt{switch\_on\_sprinklers}}{\texttt{switch\_on\_sprinklers}}$$

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

$$\frac{A_1, \ . \ . \ ., A_n}{B}$$

Some Proof Rules

- *Modus ponens* is a well known proof rule: $$\frac{A \Rightarrow B,\quad A}{B}$$

  where A and B are any WFF.

- Another common proof rule, is $\wedge$-elimination: $$\frac{A \wedge B}{A} \qquad \frac{A \wedge B}{B}$$

  Reads: *if A and B hold (or are provable or true) then A (resp. B) must also hold.*

- Another proof rule, is $\vee$-introduction is: $$\frac{A}{A \vee B} \qquad \frac{A}{B \vee A}$$

  Reads: *if A holds (or is provable or true) then A $\vee$ B must also hold.*

Natural Deduction Example

- From $r \wedge s$ and $s \Rightarrow p$ can we prove p, i.e. show $r \wedge s,\ s \Rightarrow p \vdash p$?

1. $r \wedge s$ [Given]
2. $s \Rightarrow p$ [Given]
3. s [1 $\wedge$**-elimination**] $\dfrac{r \wedge s}{s}$
4. p [2,3 modus ponens] $\dfrac{s \Rightarrow p,\ \ s}{p}$

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 $\vdash p \vee \neg p$ 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 $p \Rightarrow q, (\neg r \vee q) \Rightarrow (s \vee p), q$ can we prove $s \vee q$?

    1. $p \Rightarrow q$ [Given]
    2. $(\neg r \vee q) \Rightarrow (s \vee p)$ [Given]
    3. q [Given]
    4. $s \vee q$ [3, $\vee$-introduction]

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

$$((p \Rightarrow q) \wedge ((\neg r \vee q) \Rightarrow (s \vee p)) \wedge q) \models (s \vee q)$$

Exercise

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

$$p \Rightarrow (q \Rightarrow r),\ p \wedge q \vdash r$$

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
    $A_1 \wedge \ldots \wedge A_n \models B$

- Informally, completeness involves ensuring that *all* formulae that should be able to be proved can be.
  - **Theorem**(Completeness): If $A_1 \wedge \ldots \wedge A_n \models B$ then
    $A_1, \ldots, A_n \vdash B$

More on Soundness and Completeness

- Example: An unsound (bad) inference rule is: $\dfrac{A, \quad B}{C}$

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

- The set of rules modus ponens and ∧-elimination is incomplete: without ∨-introduction we cannot do the proof on slide 28, yet

$$((p \Rightarrow q) \wedge ((\neg r \vee q) \Rightarrow (s \vee p)) \wedge q) \models (s \vee q)$$

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