INST0072 Lecture 1: Module Introduction, Propositional Logic

1. About This Module

- Module home page on Moodle - lots of information here.
- Mostly about *classical logic*, and the *logic programming* language Prolog (using SWI Prolog for practical exercises).
- Problem-based learning. The exercises are essential! Attempting additional exercises from the books will also help.
- Books: the main texts are by Ertel (for logic) and Bratko (for Prolog). Suggested reading for each week on Moodle.
- Pre-recorded lectures will be available on Moodle each week, with an accompanying set of notes.
- Ideally you should watch the lecture videos each week before the Tuesday support class.
- Assessment: One logic component (25%) and two Prolog programming components (25% and 50%).

2. What is Knowledge Representation?

- Wikipedia: *"Knowledge representation is a field of AI that focuses on designing computer representations that capture information about the world that can be used to solve complex problems ... Knowledge representation goes hand in hand with automated reasoning ..."*.
- Knowledge-based systems separate knowledge (facts and rules about the world) from computation. So the system behaviour can be extended just by adding new knowledge.
- Computation often reflects aspects of human reasoning (e.g. logical deduction or inference). So the system behaviour can be justified and explained in human terms.
- Levesque and Lakemeyer: *"The hallmark of a knowledge-based system is that by design it has the ability to be told facts about its world and adjust its behaviour correspondingly"*.

3. A Simple Prolog Example

This example is (more or less) from Levesque and Lakemeyer, Chapter 1.

- Procedural version:

```
printColour(snow) :- !, write('It is white.').
printColour(grass) :- !, write('It is green.').
printColour(sky) :- !, write('It is blue.').
printColour(vegetation) :- !, write('It is green.').
printColour(X) :- write('Beats me.').
```

- Knowledge-based version:

```
printColour(X) :- colour(X,Y), !,
    write('It is '), write(Y), write('.').
printColour(X) :- write('Beats me.').

colour(snow, white).
colour(sky, blue).
colour(X, Y) :- madeof(X, Z), colour(Z, Y).
madeof(grass, vegetation).
colour(vegetation, green).
```

4. Classical Logic

- A logic is a symbolic calculus representing (aspects of) rational human thought and reasoning. Logic has a very long history, starting with ancient Greeks such as Aristotle.
- *Classical logic* has two main forms: *propositional logic* (or *calculus*), and *predicate logic* (or *calculus*).
- *Propositions* are statements or assertions that can either be *false* or *true*. In propositional logic, *formulas* are made by stringing propositions together with *logical connectives*:

$$(\mathit{it\_is\_raining} \land \mathit{i\_am\_outside}) \rightarrow \mathit{i\_am\_wet}$$

- In predicate logic, propositions become *predicates* with (zero or more) *arguments*, which can sometimes be *universally* or *existentially quantified variables*:

$$[\forall x(\mathit{man}(x) \rightarrow \mathit{mortal}(x)) \land \mathit{man}(\mathit{Socrates})] \rightarrow \mathit{mortal}(\mathit{Socrates})$$

5. Commonly Used Propositional Logic Symbols and Connectives, and Formulas

Symbol	**Meaning**
$\top$, *true*, *True*, *t*	true
$\bot$, *false*, *False*, *f*	false
$\neg$ ...	not ...
... $\wedge$ ...	... and ...
... $\vee$ ...	... or ...
... $\rightarrow, \Rightarrow, \supset, \Longrightarrow$ ...	if ... then ... (... implies ...)
... $\leftarrow, \Leftarrow, \subset, \Longleftarrow$ ...	... if ... (... implied by ...)
... $\leftrightarrow, \Leftrightarrow, \equiv, \Longleftrightarrow$ ...	... if and only if ...

The set of words and/or single characters used for propositions is called the *signature* of the logic, and can include any symbols except the above:

$$[(\mathit{is\_lecturer} \vee \mathit{is\_student}) \wedge \mathit{is\_at\_UCL}] \leftrightarrow \mathit{is\_very\_clever}$$

6. Limitations of Propositional Logic?

Using the propositions *electric_shock* and *touch_wire*, how if at all can the following sentences be represented as propositional logic formulas?

- “If you touch the wire then you’ll get an electric shock.”
- “If you don’t touch the wire then you won’t get an electric shock.”
- “If you touch the wire then you may get an electric shock.”
- “If you touch the wire then you’ll probably get an electric shock.”
- “Don’t touch the wire!”

7. Interpretations and the Meanings of the Logical Operators

- An *interpretation* of a propositional logic is an assignment of a truth value to each proposition in its signature. In other words it is a *mapping* from the signature to the set $\{f, t\}$, e.g:

$$I : \{electric\_shock, touch\_wire\} \mapsto \{f, t\}$$
$$I(electric\_shock) = f,\ I(touch\_wire) = t$$

- The meaning (or *semantics*) of each logical operator is shown in the following table. Each row in the first two columns is a different interpretation of the logic with signature $\{A, B\}$:

A	B	$\neg A$	$A \wedge B$	$A \vee B$	$A \rightarrow B$	$A \leftarrow B$	$A \leftrightarrow B$
t	t	f	t	t	t	t	t
t	f	f	f	t	f	t	f
f	t	t	f	t	t	f	f
f	f	t	f	f	t	t	t

8. Formulas, Truth Tables and Models

We can use the rules in the table on the previous slide to form a *truth table* for any formula, as in this example:

$$((red \vee big) \wedge \neg red) \rightarrow big$$

We construct the table as follows:

red	big	$red \vee big$	$\neg red$	$(red \vee big) \wedge \neg red$	$((red \vee big) \wedge \neg red) \rightarrow big$
t	t	t	f	f	t
t	f	t	f	f	t
f	t	t	t	t	t
f	f	f	t	f	t

Interpretations that result in a value of *true* for the formula are called *models of the formula*. So the formula above has four models.

9. Truth Tables for Large Formulas

An alternative method for working out truth tables for large formulas is to place truth values under each proposition and connective, working outwards from the most nested connectives to the single least nested connective, as in the following example:

$$((red \lor big) \land (\neg red \lor soft)) \to (big \lor soft)$$

In this case the "→" is the least nested connective. We construct the table as follows:

red	big	$soft$	$((red$	$\lor$	$big)$	$\land$	$(\neg$	red	$\lor$	$soft))$	$\to$	$(big$	$\lor$	$soft)$
t	t	t	$\cancel{t}$	$\cancel{t}$	$\cancel{t}$	$\cancel{t}$	$\cancel{f}$	$\cancel{t}$	$\cancel{t}$	$\cancel{t}$	t	$\cancel{t}$	$\cancel{t}$	$\cancel{t}$
t	t	f	$\cancel{t}$	$\cancel{t}$	$\cancel{t}$	$\cancel{f}$	$\cancel{f}$	$\cancel{t}$	$\cancel{f}$	$\cancel{f}$	t	$\cancel{t}$	$\cancel{t}$	$\cancel{f}$
t	f	t	$\cancel{t}$	$\cancel{t}$	$\cancel{f}$	$\cancel{t}$	$\cancel{f}$	$\cancel{t}$	$\cancel{t}$	$\cancel{t}$	t	$\cancel{f}$	$\cancel{t}$	$\cancel{t}$
t	f	f	$\cancel{t}$	$\cancel{t}$	$\cancel{f}$	$\cancel{f}$	$\cancel{f}$	$\cancel{t}$	$\cancel{f}$	$\cancel{f}$	t	$\cancel{f}$	$\cancel{f}$	$\cancel{f}$
f	t	t	$\cancel{f}$	$\cancel{t}$	$\cancel{t}$	$\cancel{t}$	$\cancel{t}$	$\cancel{f}$	$\cancel{t}$	$\cancel{t}$	t	$\cancel{t}$	$\cancel{t}$	$\cancel{t}$
f	t	f	$\cancel{f}$	$\cancel{t}$	$\cancel{t}$	$\cancel{t}$	$\cancel{t}$	$\cancel{f}$	$\cancel{t}$	$\cancel{f}$	t	$\cancel{t}$	$\cancel{t}$	$\cancel{f}$
f	f	t	$\cancel{f}$	$\cancel{f}$	$\cancel{f}$	$\cancel{f}$	$\cancel{t}$	$\cancel{f}$	$\cancel{t}$	$\cancel{t}$	t	$\cancel{f}$	$\cancel{t}$	$\cancel{t}$
f	f	f	$\cancel{f}$	$\cancel{f}$	$\cancel{f}$	$\cancel{f}$	$\cancel{t}$	$\cancel{f}$	$\cancel{t}$	$\cancel{f}$	t	$\cancel{f}$	$\cancel{f}$	$\cancel{f}$

10. Another Truth Table Example

- What is the truth table of the formula

  $$(touch\_wire \to electric\_shock) \leftrightarrow (touch\_wire \land \neg electric\_shock)$$

  ?

11. Some Logic Terminology

- **Semantic Equivalence**: Two formulas are ***semantically equivalent*** if they have the same truth table.
- **Satisfaction**: An interpretation ***satisfies*** a formula if it assigns a value of true (t) to the formula, in which case the interpretation is a ***model*** of the formula. A formula is ***satisfiable*** or ***consistent*** if it has at least one model. Otherwise it is ***unsatisfiable*** or ***inconsistent*** or ***a contradiction***.
- **Validity**: A formula is ***(logically) valid*** or ***a tautology*** if it is satified by every interpretation, i.e. if every interpretation is a model of the formula. For any formula A, the expression $\models A$ means that A is logically valid.
- **Entailment**: For any two formulas A and B, the expression $A \models B$ (read as "A ***entails*** B") means that every model of A is also a model of B. In this case B ***is entailed by*** or is a ***semantic consequence*** of A.

12. Some Standard Equivalences

For any formulas A, B and C:

Formula	**Equivalent to**	
$\neg A \lor B$	$A \rightarrow B$	(implication)
$A \rightarrow B$	$\neg B \rightarrow \neg A$	(contrapositive)
$(A \rightarrow B) \land (B \rightarrow A)$	$A \leftrightarrow B$	(equivalence)
$\neg(A \land B)$	$\neg A \lor \neg B$	(De Morgan's laws)
$\neg(A \lor B)$	$\neg A \land \neg B$	(De Morgan's laws)
$A \lor (B \land C)$	$(A \lor B) \land (A \lor C)$	($\lor$ distribution)
$A \land (B \lor C)$	$(A \land B) \lor (A \land C)$	($\land$ distribution)
$A \lor \neg A$	*any tautology*	(tautology)
$A \land \neg A$	*any contradiction*	(contradiction)
$A \land (B \lor \neg B)$	A	(tautology elimination)
$A \lor (B \land \neg B)$	A	(contradiction elimination)
$A \land (B \land C)$	$(A \land B) \land C$	($\land$ associativity)
$A \lor (B \lor C)$	$(A \lor B) \lor C$	($\lor$ associativity)
$A \land B$	$B \land A$	($\land$ reordering)
$A \lor B$	$B \lor A$	($\lor$ reordering)
$A \leftrightarrow B$	$B \leftrightarrow A$	($\leftrightarrow$ reordering)
$\neg\neg A$	A	($\neg$ cancellation)
$A \land A$	A	($\land$ repetition)
$A \lor A$	A	($\lor$ repetition)

13. Some Examples

- Which of the terms "*satisfiable*", "*consistent*", "*unsatisfiable*", "*inconsistent*", "*contradiction*", "*valid*" and "*tautology*" are true of each of the following formulas, and how many models does each have?
  - $(a \rightarrow b) \leftrightarrow \neg(a \wedge \neg b)$
  - $(\neg a \rightarrow \neg b) \wedge \neg(\neg b \vee a)$
  - $(a \rightarrow b) \rightarrow a$
- Which two of these formulas are semantically equivalent?
  (1) $\neg(a \vee \neg b)$
  (2) $\neg a \wedge \neg b$
  (3) $\neg a \wedge b$

14. Two Theorems

The Deduction Theorem
For any two formulas A and B

$$A \models B \quad \text{if and only if} \quad \models A \rightarrow B$$

Proof is by considering all the possible models for $A \rightarrow B$. (See Ertel, page 27.)

Proof by Contradiction
For any two formulas A and B

$$A \models B \quad \text{if and only if} \quad A \wedge \neg B \text{ is a contradiction.}$$

Proof: $A \wedge \neg B$ is a contradiction if and only if $\neg(A \wedge \neg B)$ is a tautology. $\neg(A \wedge \neg B)$ is equivalent to $\neg A \vee \neg\neg B$ and so also to $A \rightarrow B$, so the theorem is a consequence of the Deduction Theorem.

15. Sets of Formulas, Knowledge Bases, Theories and Axioms

- A logic *knowledge base* is a (possibly large) finite set KB of formulas. A set of formulas is also sometimes called a *theory* or *axiomatisation*, and the formulas within it are sometimes called *axioms*.

- In propositional logic, the theory $KB = \{F_1, F_2, \ldots, F_n\}$ is defined as semantically equivalent to the single (possibly long) formula $F_1 \wedge F_2 \wedge \ldots \wedge F_n$. So

$$KB \models F \qquad \text{means} \qquad (F_1 \wedge F_2 \wedge \ldots \wedge F_n) \models F$$

- Using the above, the *Proof by Contradiction* theorem can be re-expressed in terms of sets of formulas:

Proof by Contradiction (set-theoretic version)
For any knowledge base or theory KB and formula F

$$KB \models F \quad \text{if and only if} \quad KB \cup \{\neg F\} \text{ is unsatisfiable.}$$

16. Inference Rules and Soundness

- An *inference rule* has the form

$$\frac{P_1,\ \ldots,\ P_n}{C}$$

$P_1, \ldots, P_n$ are called the *premises* of the rule, and C is called the *concluslon*. $P_1, \ldots, P_n$ and C are all formula templates or patterns (sometimes called *formula* or *axiom schemas*).

- Examples:

$$\frac{A,\ A \rightarrow B}{B}\ \textit{(modus ponens)} \qquad \frac{A \rightarrow B,\ \neg B}{\neg A}\ \textit{(modus tollens)}$$

- An inference rule is *sound* if the conclusion is always a semantic consequence of the premises, i.e. if $F_1 \wedge \ldots \wedge F_n \models F$ for any formulas $F_1, \ldots, F_n$ and F that collectively match the formula templates $P_1, \ldots, P_n$ and C.

- A set of inference rules is sometimes called a *calculus*.

17. Derivations

- Given a set of inference rules (i.e. a calculus) and a theory KB, a *derivation of a formula* F is a process in which KB is made increasingly large by repeatedly adding a conclusion C of an inference rule whose premises $P_1, \dots, P_n$ are already in the (growing) theory, until F itself is in the theory.
- $KB \vdash F$ means it is possible to derive F from KB in this way.
- Here is an example when $KB = \{a,\ a \to \neg b,\ c \to b\}$:

$$\{a,\ a \to \neg b,\ c \to b\} \ \vdash\ \neg c$$

derivation uses modus ponens followed by modus tollens:

$$\{a,\ a \to \neg b,\ c \to b\} \Rightarrow \frac{a,\ a \to \neg b}{\neg b} \Rightarrow \{a,\ a \to \neg b,\ c \to b,\ \neg b\}$$
$$\Rightarrow \frac{c \to b,\ \neg b}{\neg c} \Rightarrow \{a,\ a \to \neg b,\ c \to b,\ \neg b,\ \neg c\}$$

18. Writing Out Derivations

Example:

$$KB = \{raining,\ raining \to \neg warm,\ sunny \to warm\}$$

$$KB \ \vdash\ \neg sunny$$

Derivation:

(1) $raining$	by assumption (from KB)
(2) $raining \to \neg warm$	by assumption (from KB)
(3) $\neg warm$	by (1), (2), modus ponens
(4) $sunny \to warm$	by assumption (from KB)
(5) $\neg sunny$	by (3), (4), modus tollens