More on Predicate Calculus

Logic - recap

- So far, we have seen that:
- Logic is a “language” which can be used to describe:
  - Statements about the real world
  - The simplest pieces of data in an automatic processing system such as a computer

Logic - recap

- The key characteristic of a statement, or a piece of data, in logic is its truth value. There are two possible truth values:
  - True
  - False

Logic - recap

- Simple statements, or simple pieces of data, can be joined together to give more complicated structures using logical connectives.
- The most important of these are “and”, “or”, “implies”, “equivalent to” and “not”, for which we use the symbols $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$, and $\neg$ respectively.

Logic - recap

- Each of these logical connectives is defined by a truth table.
- It is often possible to convert an English sentence into a logical formula – there are simple techniques for doing this.

Logic - recap

- Questions:
  - “when is this formula true?”
  - “are these two formulae essentially the same?”
- One useful technique for answering questions about logical formula (such as above) is to **draw up a large truth table** describing the formula/formulae in question.

What is low-level logic used for?

- To specify the way in which individual bits of information are processed in order to perform the arithmetic, and all the other forms of symbolic processing, that goes on in the circuits of a computer.

What is high-level logic used for?

- Used as an unambiguous language,
- i.e. as a medium for knowledge representation.

What is high-level logic used for?

- Used as a rigorous form of reasoning.
- Deductions which follow the rules of logic are guaranteed to be correct.
- This is not necessarily true of other representation schemes.
- So logic can act as a **"gold standard"** against which other forms or representation and reasoning can be assessed.

What is high-level logic used for?

- Logic is available for establishing whether a complicated argument, e.g. a legal case, makes sense or not.

What is high-level logic used for?

- Logic is a tool for mathematical research.
- The mathematician has available a set of propositions which have been established as true -
  - these are referred to as "the axioms".
- He/she has a proposition, or set of propositions, that he/she wants to prove to be true -
  - referred to as "the theorem".
- Both sets can be written as *statements in logic:*
  - the problem is to find a set of *sound logical manipulations* to convert one into the other.

More on processing logical formulae

- The “build a truth table” technique described earlier always works,
  - at least for the simple form of logic we’ve been using so far.
- However, sometimes it isn’t necessary. Instead, to solve a problem in logic, we can use:
  - Rules of inference
  - Logical equivalencies

Rules of inference

- These are well-established ways of proving an argument in logic.

The most useful are:

- **modus ponens:** $P \Rightarrow Q. \quad P. \quad \therefore Q.$
- **modus tolens:** $P \Rightarrow Q. \quad \neg Q. \; \therefore \neg P.$
- **disjunctive syllogism:** $P \vee Q. \quad \neg P. \; \therefore Q.$
- **hypothetical syllogism:** $P \Rightarrow Q. \quad Q \Rightarrow R.$
  $\therefore P \Rightarrow R.$
- **contraposition:** $P \Rightarrow Q. \; \therefore \neg Q \Rightarrow \neg P.$
- In English, the first of these would read “If you know that P implies Q, and you know that P is true, then you know that Q is true.”
- These can be used to establish the truth of arguments involving complicated formulae.

Logical equivalences:

- these are well-established ways of turning a logical statement containing one set of logical connectives into a statement containing a different set.

Logical equivalences

- Here are some of the more important ones.
  - *Idempotency*

    P is logically equivalent to $(P \wedge P)$
  - *Double negation*

    P is logically equivalent to $\neg\neg P$
  - *Commutative laws*

    $(P \wedge Q)$ is logically equivalent to $(Q \wedge P)$

    $(P \vee Q)$ is logically equivalent to $(Q \vee P)$

Logical equivalencies

- *Associative laws*

$(P \wedge (Q \wedge R)$ is logically equivalent to $((P \wedge Q) \wedge R)$

$(P \vee (Q \vee R))$ is logically equivalent to $((P \vee Q) \vee R)$

- *Distributive laws*

$(P \wedge (Q \vee R)$ is logically equivalent to $((P \wedge Q) \vee (P \wedge R))$

$(P \vee (Q \wedge R))$ is logically equivalent to $((P \vee Q) \wedge (P \vee R))$

Logical equivalences

- *De Morgan's laws*

$\neg(P \wedge Q)$ is logically equivalent to $(\neg P \vee \neg Q)$

$\neg(P \vee Q)$ is logically equivalent to $(\neg P \wedge \neg Q)$

- *Contraposition*

$(P \Rightarrow Q)$ is logically equivalent to $(\neg Q \Rightarrow \neg P)$

Logical equivalences

- *Redefining material conditional as a disjunction or conjunction*

$(P \Rightarrow Q)$ is logically equivalent to $(\neg P \vee Q)$

$(P \Rightarrow Q)$ is logically equivalent to $\neg(P \wedge \neg Q)$

- *Exportation*

$(P \Rightarrow (Q \Rightarrow R))$ is logically equivalent to

$((P \wedge Q) \Rightarrow R))$

Logical equivalencies

- **It's not necessary** to remember the elaborate names, or the exact details of these equivalences.
- It is important to appreciate that, by using one of these equivalences, a logical formula can be **turned into a formula** which looks quite different, but which actually *has exactly the same truth value*.

Tautologies and contradictions

- A logical statement which is always true (no matter what values you give to the parts that make it up) is called a ***tautology*.**
- A statement which is always false is called a ***contradiction*.**
- All other logical statements are known as **contingent statements.**

Tautologies and contradictions

- *example:* Consider the statement
  $(\neg P \Rightarrow ((P \vee Q) \Leftrightarrow Q))$.

P could be true and Q could be true.
P could be false and Q could be false.
P could be true and Q could be false.
P could be false and Q could be true.

- In every one of these cases, the whole statement would be true.
  - So the statement is ***a tautology.***

Propositional calculus

- The simple form of logic which we have been using so far, in which the symbols stand for ideas which can be expressed as whole sentences, is known as **Propositional Calculus**.

Propositional calculus

- Propositional calculus is ***decidable***:
  - it's always possible to decide whether an argument in propositional calculus is valid or not.
  - You can either use the logical equivalence rules, and the rules of inference, described above, or you can build a big truth table.

Propositional calculus

- Propositional calculus:
  - is the simplest form of logic,
  - and also the weakest,

  in terms of its power to express ideas.
- Other forms of logic have been invented to ***express more complex ideas***.

Predicate calculus

- **Another form of logic:** first order predicate calculus.

- There are many fairly simple, obvious arguments that propositional calculus cannot handle at all.

e.g. *All lawyers are scoundrels. Sir James Thames is a lawyer. Therefore Sir James Thames is a scoundrel.*

Predicate calculus

- 1st order predicate calculus has some **extra features**: the statements joined by the logical connectives can be predicates. e.g.

  a(b) a(b, c)

  lawyer(Sir_James_Thames)

  owes(Jim, Sid, 500pounds)

  These can *convey ideas* such as "b is an a", “b has the quality a”, or "b & c are connected by the property a".

  A predicate can have any number of places, i.e. items in brackets.

Predicate calculus

- The items inside the brackets of a predicate can be constants (names of things or relationships) or variables.
  - If they're variables, they don't refer to specific things; *specific items* can be substituted for them.
  - When this happens, all variables with the same name will refer to the same thing.

Predicate calculus - the universal quantifier

- 1st order predicate calculus has a symbol ∀ called the **universal quantifier**.
- ∀X means "for every x".

Predicate calculus - the universal quantifier

- Thus, if lawyer(x) means "x is a lawyer" and scoundrel(x) means "x is a scoundrel",

  $$\forall x \; (\text{lawyer}(x) \Rightarrow \text{scoundrel}(x))$$

  means "for every x, if that x is a lawyer, then that x is a scoundrel".

  Which is the predicate calculus way of saying "All lawyers are scoundrels."

Predicate calculus - the existential quantifier

- 1st order predicate calculus has a symbol ∃ called ***the existential quantifier*.**
- ∃x means "there is an x such that".

Predicate calculus - the existential quantifier

- If lawyer(x) means "x is a lawyer" and scoundrel(x) means "x is a scoundrel",

  $$\exists x \, (\text{lawyer}(x) \wedge \text{scoundrel}(x))$$

  means "There is an x such that x is a lawyer, and x is a scoundrel".

  Which means that there is at least one lawyer who is a scoundrel, and is the predicate calculus way of saying "Some lawyers are scoundrels."

Predicate calculus

- Predicate calculus uses the same logical connectives as propositional calculus.

Logic and ambiguity

- One of the problems with converting statements in English into statements in some other knowledge representation is that English statements are <u>often ambiguous</u>. e.g.

  "Someone is mugged in Soho every day."

  [Could mean: "a particular person goes to Soho every day, and gets mugged"; or *"A mugging takes place in Soho every day (not necessarily with the same victim)".*]

Logic and ambiguity

- Statements in predicate calculus aren't ambiguous, so translating English into predicate calculus forces the ambiguity to be ironed out. The previous statement translates into:

  $\exists x(\ \text{person}(x) \wedge \forall y(\ \text{day}(y) \Rightarrow (\text{mugged\_on}(x, y) \wedge \text{mugged\_in}(x, \text{Soho}))))$

or $\forall x(\ \text{day}(x) \Rightarrow \exists y(\ \text{person}(y) \wedge \text{mugged\_on}(y, x) \wedge \text{mugged\_in}(y, \text{Soho}))))$

Predicate calculus

- Establishing whether an argument is valid or not, in First Order Predicate Calculus:
  - this is **harder** than in propositional calculus.
- You **can't use truth tables**.

Predicate calculus

- There are some rules of inference (i.e. ways of proving an argument) which correspond to those found in propositional calculus;

  e.g. universal instantiation:

$\forall x(\ p(x) \Rightarrow c(x)\ )$ p(a). $\therefore$c(a).

and the rules already established for propositional calculus apply here too.

Predicate calculus

- There are techniques for proving an argument in FOPC - sequences of steps which will always show that an argument is true, if it is true.
- If it isn't true, the sequence of steps is liable to go on for ever, without producing a result.
- For this reason, FOPC is said to be ***semi-decidable***.

Problems and Assignments

- Read Chapter 2 in the Luger/Stubblefield book.
- Read chapter 3.3 in the Luger/Stubblefield book.
- Read chapter 5.4 in the Luger/Stubblefield book.
- Create a robot planner using predicate calculus, next program it in Prolog.

Sources

- John Platts