Introductory lecture

CS242 Formal Specification and Verification

University of Warwick

Spring term 2010

Motivation

History of logic:

- antiquity and middle ages
- formal logic
- beginnings of computer science

Applications of logic in computer science:

- model checking
- deductive verification
- artificial intelligence

Aristotle 1

The Greek philosopher Aristotle developed a study of logic in a group of works collectively known as *The Organon*, written around 350 BC.
The object of Aristotle's logic is to discover the laws that govern valid reasoning, as observed in deductions.

> *A deduction is speech in which, certain things having been supposed, something different from those supposed results of necessity because of their being so.*
> *Prior Analytics I.2, 24b18–20*

Aristotle 2

A *syllogism* is, in Aristotle's theory, one of the laws that govern valid reasoning. Taken together, syllogisms constitute a *theory of inference*.

> *If all Xs are Ys, and if Z is an X, then Z is a Y.*

is an example of syllogism, that subsumes the deduction

> *All men are mortal. Socrates is a man. Therefore, Socrates is mortal.*

In Aristotle's theory, reasoning means applying syllogisms to a set of *premises* until the desired *conclusion* is reached.

Leibniz 1

The mathematician and philosopher Leibniz wrote in 1667 what is referred to today as Leibniz's dream:

> *If we could find characters or signs appropriate for expressing all our thoughts as definitely and as exactly as arithmetic expresses numbers or geometric analysis expresses lines, we could in all subjects in so far as they are amenable to reasoning accomplish what is done in Arithmetic and Geometry.*
> *For all inquiries which depend on reasoning would be performed by the transposition of characters and by a kind of calculus, which would immediately facilitate the discovery of beautiful results.*

Leibniz 2

And if someone would doubt my results, I should say to him: "Let us calculate, Sir", and thus by taking to pen and ink, we should soon settle the question.
Now the characters which express all our thoughts will constitute a new language. . . this language will be. . . very easy to learn. It will be quickly accepted by everybody on account of its great utility and its surprising facility, and it will serve wonderfully in communication among various peoples.
There will be no equivocations or amphibolies, and everything which will be said intelligibly in the language will be said with propriety.

Leibniz 3

Using modern concepts:

- reasoning is done in a formal language
- inference rules can be applied calculationally
- validity of a proof can be checked algorithmically
- sentences have well-defined meanings

Frege 1

Leibniz wanted to have logic as a branch of mathematics.
The mathematician and philosopher Frege went further and attempted to embed mathematics into logic.
In the *Begriffsschrift*, written in 1879, Frege invented *predicate logic*, a formal language for mathematics.

$$\forall x \forall y (x < y \rightarrow \exists z (x < z \ \wedge \ z < y))$$

Most mathematical statements can be written in predicate logic.

Frege 2

Frege also formalised the notion of *proof*: counterpart to the Aristotelian theory of inference.
A proof derives a *conclusion* from *axioms*, by applying *inference rules* (counterpart to the Aristotelian syllogisms) to axioms and formulas previously derived.
A proof can be represented as a *labelled tree*.

Some further history

- Russell's Paradox (1901)
- Russell & Whitehead's *Principia Mathematica* (1910–1913)
- Gödel's Incompleteness Theorems (1931)
- Continuum Hypothesis: consistent with ZFC by Gödel (1938), independent of ZFC by Cohen (1963)
- Theory of computability: Turing Machine (1934), Church's Thesis, von Neumann architecture (1945)

CS245 Automata and Formal Languages
CS246 Further Automata and Formal Languages

Formal methods

Logical and mathematical methods to specify, design, construct and maintain *reliable* hardware and software systems.

Model checking. Determining algorithmically whether a model $\mathcal{M}$ of a system satisfies a correctness condition ϕ, i.e. whether $\mathcal{M} \models \phi$. Tools called *model checkers* implement such algorithms.

Deductive verification. Correctness of a system is expressed by an assertion, and we attempt to prove the assertion using axioms and inference rules. Tools called *theorem provers* assist in constructing proofs.

CS132 Computer Organisation and Architecture
CS332 Programming Language Design and Semantics
CS400 Advanced Specification Methods

Artificial intelligence

Logics of knowledge. In a multi-agent system, each agent may have different knowledge about the environment, and also about the knowledge of other agents. Logics of knowledge can be used to facilitate reasoning within agents, and also analysis of the multi-agent system.

CS328 Artificial Intelligence

The wise-men puzzle

There are three wise men. It's common knowledge — known by everyone and known to be known by everyone, etc. — that there are three red hats and two white hats. The king puts a hat on each of the wise men in such a way that they are not able to see their own hat, and asks each one in turn whether they know the colour of the hat on their head. Suppose the first man says he does not know; then the second says he does not know either. It follows that the third man must be able to say that he knows the colour of his hat. Why is this? What colour has the third man's hat?

This module

Part 1: Weeks 2–4 and Weeks 8-10 (Dr Meurig Beynon):

- Propositional logic
- Predicate logic
- Verification by model checking
- Program verification

Part 2: Weeks 5–7 (Dr Jane Sinclair):

- Alloy tool

15 CATS
Class test (**TBC - sometime during weeks 8-10**): 10%
Assignment: 30%
Exam: 60%

Part 1

Book. M R A Huth & M D Ryan, *Logic in Computer Science: Modelling and reasoning about systems*, 2nd edition, Cambridge University Press, 2004.

Web page. `http://www.dcs.warwick.ac.uk/~wmb/fsv/`

Web tutor. `http://www.cs.bham.ac.uk/research/projects/lics/tutor/`

Forum. Departments > Computer Science > UGyear2 > CS242

Seminars. Thursdays 11-12pm (surnames A–M)
Thursdays 12-1 pm 8-10 (surnames N–Z)
Venue will be: S0.13 wks 3-4 / CS1.01 wks 8-10
Ebrahim Ardeshir, eardeshir@warwick.ac.uk
Tutor TBA
from week 3

Some references

- Stanford Encyclopedia of Philosophy:
  `http://www.seop.leeds.ac.uk/`
- Gödel's Incompleteness Theorems:
  `http://www.miskatonic.org/godel.html`
- von Neumann's architecture:
  `http://www.wikipedia.org/wiki/Von_Neumann_architecture`

Acknowledgement

Eric Martin, *Comp 2411: Logic and Logic Programming*, University of New South Wales, Australia, 2003.