Logic in Computer Science

Thierry Coquand

Nov. 9, 2007

Mathematical Logic

Traditionally mathematical logic was developped by philosophers and mathematicians interested in foundations

Logic plays a special role in computer science: it has been called "the calculus of computer science"

Logic plays a similar role in computer science to that played by calculus in the physical sciences and traditional engineering disciplines. (M. Vardi, 2007)

Logic and Computer Science

"It is reasonable to hope that the relationship between computation and mathematical logic will be as fruitful in the next century as that between analysis and physics in the last."

(J. MacCarthy, 1961)

Three systems

propositional logic

temporal logic

predicate logic

History

The greeks (Aristotle) discovered the *formal* nature of logical reasonings

All men are mortal

All greeks are men. Hence all greeks are mortal

We don't need to understand what are "men", "mortal", "greeks" to recognise the validity of this inference

History

We can use *symbols*

All A are B

All B are C. Hence all A are C

History

This is like in algebra (symbols were introduced much later there)

We can do the reasoning mechanically, without understanding the meaning of the symbols

Leibniz had the idea of reducing *reasoning* (in various domains, for instance laws) to *computation*: "*The only way to rectify our reasonings is to make them as tangible as those of the Mathematicians, so that we can find our error at a glance, and when there are disputes among persons we can simply say: let us calculate.*"

Propositional Logic

Propositional logic (Boole) provides precisely such a symbolic notation

$$A \rightarrow B, \ A \land B, \ A \lor B, \ \neg A, \ A \leftrightarrow B$$

Used extensively to automate reasoning in artifical intelligence

A diplomatic problem

As chief of staff, you are to sent out invitations to the embassy ball. You have the following constraints

(1) The ambassador instructs you to invite Peru or exclude Qatar

(2) The vice-ambassador wants you to invite Qatar or Romania or both

(3) A recent diplomatic incident means that you cannot invite both Romania and Peru

Who do you invite??

A diplomatic problem

Symbolic representation

$P \vee \neg\, Q$

$Q \vee R$

$\neg\, (R \wedge P)$

Solution (truth table): $P \wedge Q \wedge \neg R$ or $\neg P \wedge \neg Q \wedge R$

Computation of the Disjunctive Normal Form

Another example

B: battery is on

L: lamp is on

a: schwitch is on

A model of a circuit can be M: $L \leftrightarrow (a \wedge B)$

Question: do we have $M \rightarrow (\neg a \rightarrow \neg L)$??

The system and specification are abstracted by a logical formula

Use in computer science

Represent real systems: 100 000 variables, 100000 assertions (called clauses)

If we try to write the truth table: 2^{100000} cases, infeasible!

Can we solve this problem quickly?? Is there a way to solve this problem which is polynomial and not exponential (in the number of variables and clauses)

This is known as the P = NP problem

Fundamental question in mathematics and computer science (this is one of the problem of the Clay mathematics institute with 1 million dollar prize)

Use in computer science

Stålmark (swedish logician) thought he had a solution

Not quite there, but found a very efficient algorithm: he founded a company in 1989, which has now as customers Airbus, Swedish National Rail Administration, Norwegian National Rail Administration, RATP (Paris Metro), ...

Use in designing circuit

One well-known application of Boolean logic/propositional logic is for the design of circuit

Shannon Master Thesis 1937

Each digital gate is represented by a logical connective

Port XOR can be represented naturally with 5 gates but also with 4 gates

Reduction of half-adder: from 6 to 4 gates

by using logical equivalence

Towards temporal logic

The electrical values (0 or 1) in a circuit vary with time

We can have feedback

Work of McCulloch (neurologist) and Pitts (mathematician/logician): model of activities of neurons in the brain

At the origin of neural networks, finite automata

Towards temporal logic

Independently, already Aristotle asked about the logical status of statements like

The sun is rising

which has a truth value which depends on time

More recently, in philosophy, Prior (around 1950): how to have a calculus (temporal logic) for such propositions

Has the time a branching or linear structure? Free will?

Temporal logic

modal logic: $X\ A$, $F\ A$, $G\ A$

F, G, X are *modalities* for *discrete*, *linear* time

$G\ A \to F\ A$

"What will always be, will be"

$G\ A \leftrightarrow A \wedge XG\ A$

$A \to F\ A$

Temporal logic

In the 70s it was realised that it is exactly what is needed to represent concurent systems in computer science

Surprising fact: it is possible to write a program which decides if a temporal formula is valid or not!

This is not at all obvious: for instance are

$$G\ (A \lor B) \to \quad GF\ A \lor GF\ B$$

$$G\ (A \to X\ A) \land A \to G\ A$$

valid?

Temporal logic

Safety (nothing bad happens)

$G\ (ack_1 \wedge ack_2)$ "mutual exclusion"

Liveness (something good happens)

$G\ (req \rightarrow F\ ack)$ "if req then eventually ack"

Fairness

$GF\ req \rightarrow GF\ ack$ "if infinitely often req then infinitely often ack"

Predicate logic

Formalism for specifying properties of mathematical structure such as *graphs, partial order, rings,...*

Graph: a set A with a relation E

A relational structure is essentially a *relational database*

Predicate logic

"node x has at least two distinct neighbors"

$$\exists y \; \exists z \;\; y \neq z \wedge E(x, y) \wedge E(x, z)$$

"each node has at least two distinct neighbors"

$$\forall x \; \exists y \; \exists z \;\; y \neq z \wedge E(x, y) \wedge E(x, z)$$

Predicate logic

One can writes an algorithm to decide the truth of a propositional formula or a temporal formula

There is *no* algorithm to decide if a predicate logic formula is valid or not

This is exactly in order to analyse this problem that the notion of *algorithm* and *program* was first formulated (around 1930)

Predicate logic

One cannot find automatically if a formula is valid or not but it is possible to write a program to check if a given *proof* of a formula is valid or not

Interactive theorem proving

Useful for checking large complex proofs: four color theorems Kepler conjecture. Build mathematical proofs by analogy with modern software (modular way).

One joint INRIA-Microsoft project is working on this

Logic and Computer Science

Some applications of logic:

architecture (logic gates)

software engineering (specification and verification)

programming languages (semantics, logic programming)

databases (relational algebra)

artificial intelligence (automatic theorem proving)

theory of computation (general notion of complexity)

Logic and Computer Science

Logic in computer science is an *applied science*, combining foundational research with applications

Essential to solve the software/hardware correctness problem

Logic and Computer Science

Some relevant courses (master level)

Logic in Computer Science

Software engineering using formal methods

Hardware description and verification