Logic and Modelling

Todd Davies

November 7, 2014

Introduction

This is a unique course developed at the University of Manchester. It explains how implementations of logic can be used to solve a number a number of problems, such as solving hardest Sudoku puzzles in no time, analysing two-player games, or finding serious errors in computer systems.

Aims

This course intends to build an understanding of fundamentals of (mathematical) logic as well as some of the applications of logic in modern computer science, including hardware verification, finite domain constraint satisfaction and verification of concurrent systems.

Additional reading

Contents

1 Propositional logic

2 Evaluating formulae

3 Satisfiability

4 CNF

5 Definitional Clausal Form Translation

6 DPLL

7 Encoding problems in SAT

8 Randomly generated clause sets

9 Randomised algorithms for satisfiability checking

10 Signed Formulae

A signed formula is one where there is an expression, A and a boolean b, where $A = b$. If $A = b$ is *true*, in an interpretation I, then it is denoted by $I \models A = b$, and consequently, I is a model of the signed formula $A = b$.

This is also when $I(A) = b$.

If a signed formula has a model, then it is satisfiable.

10.1 Finding a model of a satisfiable formula

If we had a signed formula such as $A \Leftrightarrow B = 1$, the three interpretations that model it are:

A	B
0	0
0	1
1	1

10.2 Tabelau

A tableau is a tree with each node being a signed formula. The tableau for the signed formula $A = b$ would have the root node as $A = b$.

The notation for a set of branches is $B_1|...|B_n$, where each B_i is a branch.

10.2.1 Branch expansions

There are a number of rules that can be used to expand the branches of a tabelau.