Aims

This is an introduction to first order logic for mathematicians and computer scientists. There are three components to this course: proofs, propositional logic and predicate logic. Proofs are the basis of mathematics — how do we know what we say is true? — and also of computer science — how do I know this program will do what I think it will do? Propositional logic and predicate logic are the two ingredients of first-order logic. Propositional logic deals with proofs that can be analysed in terms of the words and, or, not, implies whereas predicate logic extends this to encompass the use of the words there exists and for all.

Syllabus

**Introduction:** An overview of the sorts of questions we shall be dealing with and, in particular, why mathematics is such an important ingredient in computer science. *(1 lecture)*

**Propositional logic:** Definition of the connectives by means of truth tables; truth tables of compound propositions; graphs for propositional formulae; order of precedence rules and brackets; contradictions, satisfiable formulae, tautologies; valid arguments; equivalence relations and logical equivalence; disjunctive and conjunctive normal forms and adequate sets of connectives; truth-trees; P = NP? *(12 lectures)*

**Boolean algebras:** Definition of Boolean algebras; the Boolean algebra of sets; the Boolean algebra of propositions; proving results about Boolean algebras; an introduction to circuit design. *(6 lectures)*

**First-order logic:** relations; names and predicates; quantification; syntax; semantics; Truth-trees. *(12 lectures)*

Teaching and Assessment

**Contact Hours:** 3 lectures and 1 tutorial per week

**Assessment:** up to 30% by class tests or other continuous assessment at least 70% by end of course 2-hour exam

**Resit Type:** exam

Content: 7th September 2015
By the end of the course, students should be able to:

- Construct truth tables of compound propositions.
- Determine whether a proposition is a contradiction, satisfiable or a tautology.
- Convert between different forms.
- Convert an argument into symbolic form and determine whether it is valid.
- Solve problems in propositional logic using truth-trees.
- Solve problems using Boolean algebras.
- Design simple circuits.
- Be able to interpret first order formulae.
- Solve problems in first order logic using truth-trees.