CS2LAP: Logic and Prolog 2000/2001
Revision Guide

1 Logic

On the logic side, this course has provided a review of the syntax and semantics of classical propositional and first-order predicate logic; it has presented deduction methods for these logics, namely natural deduction and the goal-directed system; and it has introduced the basic concepts of first-order model theory.

This part of the course is based on the textbook *Elementary Logics* by Dov Gabbay (Chapters 1, 3, 4, 7, and 8). A concise summary of the deduction methods introduced is also given in the *Proof Methods Summary* by Ulle Endriss. The most important definitions regarding syntax and semantics of first-order logic are summarised in the (handwritten) notes by Antonio de Freitas. We have covered the following issues (all of which are relevant to the exam):

1. Propositional logic and truth tables; tautologies, consistent formulas, and contradictions; propositional equivalences; normal forms (Chapter 1).
2. Natural deduction for propositional logic (Chapter 3).
3. Goal-directed deduction for propositional logic, including ready for computation form (Chapter 4).
4. Syntax and semantics of first-order logic; quantifier equivalences; models; theorems (valid formulas), consistent/satisfiable, and inconsistent/unsatisfiable formulas (Chapter 7, in particular Section 7.4 and exercises).
5. Goal-directed deduction for first-order logic, including unification and Skolemization (Chapter 8).

You are also expected to be familiar with the terminology used in the course when speaking about logic (e.g. soundness, completeness, ...).

Do as many natural deduction and goal-directed proofs as possible during your revision. In the exam, it is absolutely crucial that you write down the name of the rule (and, for natural deduction, the numbers of the formulas) you used for every single proof step. You will be given the set of natural deduction rules you are allowed to use (the same as in the *Proof Methods Summary*), but not the goal-directed rules or the translation rules for ready for computation form.

Practice questions on model theory using previous exam papers, coursework 5, and exercises from the textbook.

2 Prolog

In the Prolog part of the course the basic programming concepts of Prolog have been introduced, in particular recursion, lists, arithmetic evaluation, operators, cuts, and negation as failure. The underlying goal execution process (including matching and backtracking) has been discussed and a brief introduction to the logic foundations of Prolog has been given.

This part is based on the lecture notes *An Introduction to Prolog Programming* by Ulle Endriss. All material covered in the notes is relevant to the exam (please note that this excludes the lecture and the labs on the automated theorem prover, which are not directly relevant for the exam). Here's a brief overview:

1. Prolog syntax and terminology, matching, goal execution.
2. Lists: notation, head/tail-pattern, built-in predicates for list manipulation.
3. Arithmetic expressions: arithmetic evaluation using the `is`-operator, built-in arithmetic functions/operators and relations.
4. Operators: precedence and associativity, defining operators using `op/3`.
5. Backtracking: how it works, benefits and problems; cuts: how they work, benefits and problems; negation as failure: closed world assumption, the `\+`-operator; disjunction using `;` (semicolon).
6. Logic foundations: translation of Prolog clauses into first-order sentences, Horn formulas, goal execution as a deductive process in resolution and the goal-directed system (the latter point we have not discussed in detail).

You are expected to know the built-in predicates introduced in the notes (like e.g. `member/2`, `append/3`, `select/3`, `true/0`, `fail/0`, etc.), as well as the basic arithmetic functions (like `+` or `mod`) and relations (like `=:=` or `>=`).

During your revision you should do as many exercises as possible (again): from previous exams, the coursework, and the notes. It is important that you are confident about the Prolog syntax. Therefore, you should compile all your programs to find out about possible errors.

The concept of recursive programming is explained in some detail in Appendix A of the lecture notes.

Web site. All course material apart from the textbook and the handwritten notes on first-order logic is available from the (globally viewable) course web site:

`http://www.dcs.kcl.ac.uk/teaching/units/cs2lap/`