## Main course topics

The following list is extracted from the main topics covered in the lecture notes and on this term's assignments and test.

- Context-Free Grammars:
  - BNF and extended-BNF notation
  - derivations; parse trees
  - ambiguity; precedence and associativity

  You should be able to:
  - write a CFG for a language, and justify that your CFG is correct;
  - determine whether or not a string belongs to the language defined by a CFG;
  - draw the parse tree(s) for a string generated by a CFG;
  - understand the concept of ambiguity;
  - use precedence and associativity to disambiguate a CFG.
- Functional Programming with ML:
  - type checking vs. type inference
  - basic data types (`unit`, `bool`, `int`, `real`, `string`, lists, tuples, records, functions)
  - currying
  - patterns
  - local declarations (`let`, `local`)
  - mutual recursion with `and`
  - scope
  - polymorphism
  - user-defined types (`type`, `datatype`), including recursive types
  - exceptions, handling exceptions
  - `option`s
  - references (`ref`, `!`, `:=`)

  You should be able to understand and write ML code using any feature above, using good functional programming style.
- Logic Programming with Prolog:
  - terms and predicates
  - facts, rules, and queries
  - logical interpretation of Prolog clauses
  - unification, most general unifiers
  - goal-oriented execution, backtracking, search trees
  - lists, including basic predicates (`member`, `append`, `length`)
  - arithmetic (`is`, operators and comparisons)
  - negation as failure (`\+`), guards
  - cut (`!`)
  - functors as data structures
  - input/output (`read`, `write`, `nl`)
  - self-modifying code (`dynamic`, `asserta`, `assertz`, `retract`)

  You should be able to understand and write Prolog code using any feature above, using good logic programming style.

## Problems from past exams

On the next page, I list pretty much every question that you should be able to answer. You should look at most of the other questions as well, but it is okay to ignore questions about regular expressions and Scheme—though some of the Scheme questions might make good ML questions...

- Question 2 from August 2003 [CFG's].
- Question 5 from August 2003 [Prolog].
- Question 6 from August 2003 [Prolog].
- Question 5 from August 2004 [Prolog].
- Question 1 from August 2005 [short answers].
- Question 2 from August 2005 [CFG's].
- Question 3 from August 2005 [CFG's].
- Question 7 from August 2005 [ML].
- Question 0 from December 2005 [CFG's].
- Question 3 from December 2005 [ML].
- Question 4 from December 2005 [ML].
- Question 5 from December 2005 [Prolog].
- Question 6 from December 2005 [Prolog].
- Question 7 from December 2005 [Prolog].
- Question 9 from December 2005 [Prolog].
- Question 8 from April/May 2006 [Prolog].
- Question 10 from April/May 2006 [CFG's].
- Question 3 from August 2006 [CFG's].
- Question 6 from August 2006 [ML].
- Question 7 from August 2006 [ML].
- Question 8 from August 2006 [Prolog].
- Question 9 from August 2006 [Prolog].
- Question 10 from August 2006 [Prolog].
- Question 1 from December 2006 [CFG's].
- Question 2 from December 2006 [CFG's].
- Question 5 from December 2006 [ML].
- Question 7 from December 2006 [Prolog].
- Question 8 from December 2006 [Prolog].
- Question 5 from April/May 2007 [ML].
- Question 6 from April/May 2007 [ML].
- Question 7 from April/May 2007 [Prolog].
- Question 8 from April/May 2007 [Prolog].
- Question 9 from April/May 2007 [Prolog].
- Question 2 from August 2007 [CFG's].
- Question 3 from August 2007 [CFG's].
- Question 2 from December 2007 [CFG's].
- Question 3 from December 2007 [ML].
- Question 5 from December 2007 [Prolog].
- Question 6 from December 2007 [ML].
- Question 7 from December 2007 [ML/Prolog].
- Question 6 from April/May 2008 [ML].
- Question 7 from April/May 2008 [ML].
- Question 8 from April/May 2008 [Prolog].
- Question 9 from April/May 2008 [Prolog].
- Question 2 from August 2008 [CFG's].
- Question 6 from August 2008 [ML].
- Question 7 from August 2008 [ML].
- Question 8 from August 2008 [Prolog].
- Question 9 from August 2008 [Prolog].