Sample Questions for Midterm 2 (CS 421 Fall 2017)

Some of these questions may be reused for the exam.

1. Given a polymorphic type derivation for {} |- let id = fun x -> x in id id true : bool
2. Give a (most general) unifier for the following unification instance. Capital letters denote variables of unification. Show your work by listing the operation performed in each step of the unification and the result of that step.

   $$\{X = f(g(x),W); \; h(y) = Y; \; f(Z,x) = f(Y,W)\}$$

3. For each of the following descriptions, give a regular expression over the alphabet {a,b,c}, and a regular grammar that generates the language described.
   a. The set of all strings over {**a**, **b**, **c**}, where each string has at most one **a**

   b. The set of all strings over {**a**, **b**, **c**}, where, in each string, every **b** is immediately followed by at least one **c**.

   c. The set of all strings over {**a**, **b**, **c**}, where every string has length a multiple of four.

4. Consider the following grammar:

   $$\langle S\rangle ::= \langle A\rangle \mid \langle A\rangle \langle S\rangle$$
   $$\langle A\rangle ::= \langle Id\rangle \mid (\langle B\rangle$$
   $$\langle B\rangle ::= \langle Id\rangle] \mid \langle Id\rangle\langle B\rangle \mid (\langle B\rangle$$
   $$\langle Id\rangle ::= 0 \mid 1$$

   For each of the following strings, give a parse tree for the following expression as an <S>, if one exists, or write "No parse" otherwise:

   a. (0 1 (1] ((1 0] 1
   b. 0 (1 0 (1]
   c. (0 (1 0 1] 0]

5. Demonstrate that the following grammar is ambiguous (Capitals are non-terminals, lowercase are terminals):

   $$S ::= A \; a \; B \mid B \; a \; A$$
   $$A ::= b \mid c$$
   $$B ::= a \mid b$$

6. Write an unambiguous grammar generating the set of all strings over the alphabet {0, 1, +, -}, where + and – are infixed operators which both associate to the left and such that + binds more tightly than -.

7. Write an ocamlyacc parser for the language

   $$\langle S\rangle = \langle V\rangle \mid fun \; \langle V\rangle -> \langle S\rangle \mid \langle S\rangle\langle S\rangle$$

where <V> is any identifier, application (<S><S>) associates to the left and has higher precedence than fun <V> -> <S> . You should include in your header one or more datatypes for the abstract syntax trees of the language. You declarations should give the constructors for the datatype **token** of the tokens input into the parser.