CPSC 510: Fall Final Midway Exam

Robin Cockett

December 9, 1999

This exam is worth 12% of the course and 40% of the examination credit. This is a take home exam which is due on the first class in 2000. You must complete all except for the last question of the exam BY HAND (that is in your own hand writing).

1. (15 points) For each of these grammars you are expected to calculate the vital statistics (that is nullable, endable, first, and follow properties) and to give the item automata (Note: the grammars are written in CFG format with nonterminals in lower case, the productions for a nonterminal terminated by a ".", and the first nonterminal being the start symbol).

   (a) (5 points) Show that the following grammar is neither LL(1) nor LR(1):

   ```
   a -> A a A
      | .
   ```

   Is this grammar ambiguous?

   (b) (5 points) Show that the following grammar is LR(1) but not LALR(1):

   ```
   s -> A a D | B b D | A b E | B a E.
   a -> C.
   b -> C.
   ```

   Is the grammar LL(1)? Is this grammar ambiguous?

   (c) (5 points) Show that the following grammar is LL(1) but not SLR(1):

```
s -> a A a B | b B b A.
a -> .
b -> .
```

Is this grammar LR(1)? Is this grammar ambiguous?

2. (15 points) Consider the attribute grammar overleaf:

   (a) Draw the plumbing diagrams for this attribute system,

   (b) Show that this system is an attribute system. In particular, identify the inherited and synthesized attributes, identify the dependencies at the interfaces, and show that the system is a strongly non-circular system.

   (c) Generate mutually recursive pseudo-code to calculate the value of $S.err$.

3. (10 points) This question need not be answered by hand (i.e. you permitted to answer this question with the aid of a computer):

   Propose an extension to the language `m+` which includes string variables, and arrays of the three basic types (booleans, integers, and strings). Explain any additional instructions you would expect to have to add to the existing `am+` stack machine in order to implement this language.

   You are expected to propose a grammar for the extended language (the proposed language must be a superset of the existing language) and discuss its semantics. Pay careful attention to the following issues:

   (a) The grammar should be implementable in YACC!

   (b) Can the size of a local array in a declaration be determined by a variable/argument? If so what are the implementation issues?

   (c) Should arrays be passed by value or by reference. Discuss the implementation issues.

   (d) What basic string handling primitives should the language have?

   Please note you are not expected to solve these implementation issues in detail (hint: many of these issues are discussed in the book!)