

Midterm

CSC 7101, Spring 2005

3 March 2005

Read the whole exam first (there are 6 pages) and plan your time. You have 80 minutes to complete all the questions. There are a total of 100 points. The exam is open notes but closed neighbors. Good luck!

Name:

1. (25 pts)

Given the following grammar with start symbol $\langle A \rangle$:

$$\begin{aligned}\langle A \rangle & ::= \langle B \rangle \langle C \rangle \\ & \quad \{ \langle B \rangle.i := \langle C \rangle.s1; \} \\ & \quad \{ \langle C \rangle.i := \langle B \rangle.s + 2; \} \\ & \quad \{ \text{print}(\langle C \rangle.s2); \} \\ \langle B \rangle & ::= \mathbf{b} \\ & \quad \{ \langle B \rangle.s = 2 * \langle B \rangle.i; \} \\ \langle C \rangle_1 & ::= \langle D \rangle \langle C \rangle_2 \\ & \quad \{ \langle C \rangle_1.s1 := \langle D \rangle.s + \langle C \rangle_2.s1 + 2; \} \\ & \quad \{ \langle C \rangle_1.s2 := \langle C \rangle_2.s2 + 2 * \langle D \rangle.s; \} \\ & \quad \{ \langle C \rangle_2.i := \langle C \rangle_1.i + 1; \} \\ \langle C \rangle & ::= \langle D \rangle \\ & \quad \{ \langle C \rangle.s1 := \langle D \rangle.s + 1; \} \\ & \quad \{ \langle C \rangle.s2 := \langle D \rangle.s + \langle C \rangle.i; \} \\ \langle D \rangle & ::= \mathbf{d} \\ & \quad \{ \langle D \rangle.s := 1 \}\end{aligned}$$

State the values of all the attributes in the order in which they are evaluated for the input `bdd`, and state the value printed by the print statement.

Hints: You might want to draw the parse tree and the dependencies between the attributes first. The evaluation corresponds to two traversals of the parse tree, starting with $\langle D \rangle.s$.

2. (25 pts)

Consider the following grammar:

$$\langle S \rangle ::= \langle E \rangle$$

$$\langle E \rangle ::= 0 \mid 1 \mid 0 \langle E \rangle \mid 1 \langle E \rangle$$

This grammar accepts an arbitrary sequence of zeros and ones as input. Add appropriate attributes and conditions to the grammar to restrict the accepted language so that any ones in the input must come in multiples of three. For example, the sequence '011100111111' is legal, while '011010' is not. Do not change the context-free grammar, and do not construct the whole string, pass it up to the root, and do the computation there.

3. (25 pts)

Develop a proof rule for our axiomatic proof system for the **while-do-until** loop (PL/I and Algol had similar and even more complicated loop constructs):

while B_1 **do** S **until** B_2

For each loop iteration, we first test B_1 . If it is false, the loop terminates. If it is true, we proceed to execute the loop body S and test the second loop condition B_2 . If B_2 is true, the loop terminates. If it is false, we proceed to the next iteration.

Hint: express this statement as combinations of statements for which we already have proof rules; and build the new proof rule out of the existing ones. The resulting proof rule should have a Hoare triple with a single **while-do-until** statement at the bottom of the rule.

4. (25 pts)

The following Egyptian Multiplication algorithm takes two integers x and y as input and computes the product $x * y$ as the value of s .

Provide a sketch of the **partial** correctness proof of the Egyptian Multiplication algorithm. The loop invariant has been provided.

```
{ true }
a := x;
b := y;
s := 0;
{ x * y = a * b + s }
while b != 0 do
  if odd(b) then
    s := s + a;
  a := 2 * a;
  b := b div 2;
end
{ s = x * y }
```

where `div` is regular integer division, and `odd()` returns true if its argument is an odd integer and false otherwise.

Clearly separate the proof into the three parts before, in, and after the while loop, and indicate the implications resulting from the rule of consequence. You don't need to prove these implications in detail. It's good enough to show the structure of the proof.

(Space for answering question 4.)