Universiteit Leiden

Leiden University
Leiden Institute for Advanced Computer Science
Fall 2007

Concepts of Programming Languages

F. Arbab

Final Exam

Thursday 20 December 2007

Student name:

Student number:

This exam consists of 10 pages and 9 questions, for a total of 100 points.

Instructions:

- Write your name and student number on this hand-out.
- Verify that your copy of this hand-out is complete and legible.
- This is a closed-book, closed-notes, individual exam.
- You can work on this exam only within the allocated time-slot.
- Do not unstaple or tear off pages of this hand-out.
- You may write your answers on this hand-out.
- You must return all pages of this hand-out to the proctor at the end of the exam, regardless of whether or not you have written anything on it.

1. **8 points**

   (a) **(5 points)** Write a grammar for the language consisting of strings that have $n > 0$ copies of the letter `a`, followed by a single letter `b`, followed by $2 \times n$ copies of the letter `c`. For instance, the following examples illustrate some of the strings that do and do not belong to this language:

   - reject `a`
   - reject `b`
   - reject `abc`
   - reject `aabcc`
   - reject `aabbcccc`
   - accept `abcc`
   - accept `aabcccc`
   - accept `aaabcccccc`

   (b) **(3 points)** Draw the parse tree for the sentence `aabcccc` as derived from your grammar.

2. **10 points**
Consider the following grammar:

$\langle S \rangle \rightarrow \langle A \rangle$
$\langle A \rangle \rightarrow \langle A \rangle + \langle A \rangle \mid \langle id \rangle$
$\langle id \rangle \rightarrow a \mid b \mid c$

(a) **(5 points)** Prove that this grammar is ambiguous.

(b) **(5 points)** Write an unambiguous grammar for the same language as the above ambiguous grammar.

3. **10 points**
   Prove that the following program is correct:

```
{n > 0}
count = n;
sum = 0;
while count <> 0 do
  sum = sum + count;
  count = count - 1;
end
{sum = 1 + 2 + 3 + ...  + n}
```

4. **10 points**
Consider the following program written in some programming language with Ada-like syntax.

```
procedure Main is
  x :  integer := 10;
  procedure Sub3; -- This header declaration allows Sub1 to call Sub3
  procedure Sub1 is
    x :  integer := 20;
    procedure Sub2 is
      begin -- of Sub2
        print(x);
        Sub3;
      end -- of Sub2
    begin -- of Sub1
      print(x);
      Sub2;
    end -- of Sub1
  procedure Sub3 is
    procedure Sub4 is
      begin -- of Sub4
        print(x);
      end -- of Sub4
    begin -- of Sub3
      print(x);
      Sub4;
    end -- of Sub3
  begin -- of Main
    print(x);
    Sub1;
  end -- of Main
```

Assuming that the procedure call `print(x)` prints the value of its integer argument, what sequence of value will the run of this program produce if:

(a) **(5 points)** This programming language exclusively uses static scoping.

(b) **(5 points)** This programming language exclusively uses dynamic scoping.

5. **10 points** (5×2)
   Consider the following storage allocation schemes for string variables. Define what precisely each of these schemes means in terms of when storage is allocated/deallocated, and how it affects the possible mutations of the contents of string variables.

   (a) static

   (b) fixed stack-dynamic

   (c) stack-dynamic

   (d) fixed heap-dynamic

   (e) heap-dynamic

6. **10 points** (2×10)
Consider the following C function:

```
int fun(int *k)
{
  *k -= 4;
  return 5 * (*k) + 4;
}
```

Suppose `fun()` is used in a program as follows:

```
void main()
{
  int i = 20, j = 30, sum1, sum2;
  sum1 = (i / 2) + fun(&i);
  sum2 = fun(&j) + (j / 2);
}
```

What are the values of `sum1` and `sum2` if the operands in the expression are evaluated from:

(a) left to right?

(b) right to left?

7. **12 points**
Consider the following program written in C syntax:

```
void swap(int a, int b)
{
  int temp;
  temp = a;
  a = b;
  b = temp;
}

void main()
{
  int value = 1, list[5] = {1, 3, 5, 7, 9};
  swap(value, list[1]);
  swap(list[1], list[2]);
  swap(value, list[value]);
}
```

Array indices start with 0. Show the value of the variable `value` and the entire contents of the array `list` after each of the three `swap()` calls, assuming that in this language parameters are passed:

(a) **(4 points)** by value.

(b) **(4 points)** by reference.

(c) **(4 points)** by value-result.

8. **10 points**
Consider two tasks A and B that use the shared variable `Balance`, as below:

```
Balance := 6;

task A;
  Balance := 2 * Balance;
end A;

task B;
  Balance := Balance - 1;
end B;
```

Assume that each of these arithmetic operations is done in three atomic steps: (1) fetching the current value, (2) performing the operation, and (3) storing the new value back. In the absence of any synchronization, what sequences of events are possible, and what values result in each case?

9. **20 points**

Write a Scheme function `(map f l)` that takes two parameters, `f` and `l` and returns a list. The parameter `f` is a function with a single parameter, and `l` is a list. The returned list is the result of the application of `f` to the respective elements of `l`. For instance, if `double` is a function defined as `(DEFINE (double x) (* 2 x))` then `(map double '(1 3 5 7 9))` must return the list `(2 6 10 14 18)`.