Programming 2009 — international version

Basic exam

June 30, 2009

Duration of the exam: 180 minutes. Grades:

points	grade
0– 8	2.0
9–10	3.0
11–12	3.5
13–14	4.0
15–16	4.5
17	5.0

Problem 1 (1 p.). Define in Prolog a predicate `p(?L)` satisfied when the remainder of the division of the length of the list `L` by 4 is 3. You are not allowed to use arithmetic or auxiliary predicates.

Problem 2 (1 p.). Define in Prolog a predicate `revapp/3` that acts similarly to the standard predicate `append/3` with the difference that the first list is reversed before it is appended to the second one. We have for example:

```
?- revapp([1,2,3],[4,5],X).
X = [3, 2, 1, 4, 5].
```

The predicate should work correctly at least in the mode `revap(+L1,+L2,?L3)`. You are not allowed to use arithmetic or auxiliary predicates.

Problem 3 (1 p.). We represent in Prolog binary representations of nonnegative integers using lists of digits `0` and `1` from the least to the most significant. Define a predicate `succ/2` that computes the successor of a number in this representation. We have for example:

```
?- succ([0,1,0,1],L).
L = [1, 1, 0, 1].

?- succ([1,1,0,1],L).
L = [0, 0, 1, 1].

?- succ([1,1,1,1],L).
L = [0, 0, 0, 0, 1].
```

Problem 4 (1 p.). We represent in Prolog binary trees with labelled internal nodes using structures built from the atom `leaf/0` and the functor `node/3`, whose the first and third argument is, respectively, the left and right subtree, and the second — the label of a node. Define a predicate `r(+T,?N)` satisfied when the label `a/0` occurs `N` times in any path from the root to the leaf in a tree `T`.

Problem 5 (1 p.). What is the answer of the Prolog machine to the query

```
?- \+ append(X,Y,Z).
```

Problem 6 (2 p.). Write the set of productions of a context-free grammar over the terminal alphabet $\{a, b\}$ and nonterminal alphabet $\{S\}$ that generates the language

$$\{w \in \{a, b\}^* : |w|_a \leq |w|_b\},$$

where $|w|_x$ for $x \in \{a, b\}$ stands for the number of occurrences of a symbol x in a word w.

Problem 7 (1 p.). Write a single sentence containing the definition of the notion of an unambiguous context-free grammar.

Problem 8 (1 p.). Write the rules of the big-step operational semantics (natural semantics) for the instruction

$$\texttt{repeat}\ c\ \texttt{until}\ b$$

Execution of this statement consists in repetitive execution of the statement c. The loop is completed when the condition b is satisfied after the execution of the statement c.

Problem 9 (1 p.). Arithmetic expressions of the "While" language are described by the following abstract syntax:

$$\begin{array}{rcl} n & ::= & \text{integer literals} \\ x & ::= & \text{identifiers} \\ a & ::= & n \mid x \mid a_1 \oplus a_2 \\ \oplus & ::= & + \mid - \mid \times \mid \texttt{div} \mid \texttt{mod} \end{array}$$

Give the rules of the small-step operational semantics (structural operational semantics) for arithmetic expressions of the "While" language.

Problem 10 (2 p.). We add the postincrementation and postdecrementation operators to the set of arithmetic expressions of the "While" language:

$$
\begin{aligned}
n &::= \text{integer literals} \\
x &::= \text{identifiers} \\
a &::= n \mid x \mid a_1 \oplus a_2 \mid x\texttt{--} \mid x\texttt{++} \\
\oplus &::= + \mid - \mid \times \mid \texttt{div} \mid \texttt{mod}
\end{aligned}
$$

Define the domain of denotations of the arithmetic expressions above.

Define the denotational semantics of those arithmetic expressions.

Problem 11 (1 p.). Give the weakest precondition for the program

```
while N <> 0 do R := 2 * R; N := N - 1; done
```

with the postcondition $\texttt{R} = 2^i$.

Problem 12 (3 p.). Decorate the program below with assertions to get the proof of partial correctness of this program with respect to the given specification.

$\{\texttt{N} = i\}$

{ .... }

```
R := 1;
```

{ .... }

{ .... }

```
while N <> 0 do
```

{ .... }

{ .... }

```
    R := 2 * R;
```

{ .... }

```
    N := N - 1;
```

{ .... }

{ .... }

```
done
```

{ .... }

$\{\texttt{R} = 2^i\}$

Problem 13 (1 p.). Consider a signature consisting of symbols $a/0$, $b/0$, $:/2$ and $+\!\!+/2$. Let

$$E_1 \quad = \quad \{a +\!\!+ x = a : x,\ b +\!\!+ x = b : x,\ (a : x) +\!\!+ y = a : (x +\!\!+ y),\ (b : x) +\!\!+ y = b : (x +\!\!+ y)\}$$

We assume the initial algebra semantics. Prove that $(x +\!\!+ y) +\!\!+ z = x +\!\!+ (y +\!\!+ z)$.