Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science

6.826 – Principles of Computer Systems

Handout 7 September 13, 1993

Problem Set #1

Due date: September 20, 1993

The first few problems are simple finger exercises to get you used to Spec. The last problem reinforces the material covered in the first two lectures.

Problem 1.

What is the final value of `s` in the following program?

```
TYPE S = SEQ INT
TYPE I = INT
VAR i := 3, s, s′, s″ |
s   := S{2,5,11,17,3,5,7,13};
s′  := S{2,11,5};
s″  := S{5,2,17,3};
s   := ((s ++ i) + s′) - s″
```

Problem 2.

Describe the input-output relation denoted by the following procedure.

```
TYPE S = SEQ INT
APROC F (i: INT, s) -> INT =
   <<   VAR s′ | (s′ ++ i) <<= s => RET 0
     [] BEGIN i # 0 => i := 0 [*] SKIP END; RET 1>>
```

Problem 3.

Explain how the `F` routine above would behave differently if `BEGIN` and `[*] SKIP END` were left out?

Problem 4.

Describe the input-output relation denoted by the following procedure.

```
APROC G (i: INT, j: INT) -> INT =
   <<   i // 3 = 0 \/ j // 3 = 2 => RET 0
     [] i // 3 = 1 \/ j // 3 = 2 => RET 1 >>
```

Problem 5.
Can the `G` routine above be a `FUNC` instead of an `APROC`? Why or why not?

Problem 6.
For what initial values of `x` is the following loop guaranteed to terminate? (`x` has type `INT`.) Explain briefly how the other initial values of `x` may make the loop non-terminating.

```
DO
  x < 100  => x := x + 1
 []
  x//2 = 0 => x := (x-2)**2
OD
```

Problem 7.
Write an expression whose value is a function of type `(INT, INT -> INT)` that returns the remainder of its first argument being divided by its second argument.

Problem 8.
Let `modulus` denote the value of the expression from Problem 1. Write an expression using a function constructor whose value is the same as `modulus` except when the second argument is 0, it maps to -1.

Problem 9.
(A) Write a specification in Spec of a routine that finds a shortest path from node 0 to node $n-1$ of a graph whose nodes are labeled by integers in the range 0 to $n-1$. Assume that the input graph is represented by an adjacency list `al` of type `SEQ SEQ INT`, where `al[i]` is a sequence whose elements are the successors of the node labeled by `i`. Also, asumme that all edges are of unit length. The output of the routine should be of type `SEQ INT`.

(B) What type should your routine be? (FUNC, APROC, or PROC?)

(C) Describe in words the input-output relation denoted by the specification you wrote in part (A).

(D) Write an implementation in Spec of the module you specified in part (A).

(E) Describe in words the input-output relation denoted by the implementation you wrote in part (D).