Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science

6.826 – Principles of Computer Systems

Handout 23 October 4, 1993

Problem Set #4

Due date: October 13, 1993

The first three problems test your understanding of non-atomic semantics of Spec and concurrency.

Problem 1.

Consider the following Spec program:

```
VAR i: Int := 0

PROC P1 () =
     DO i//3#0 => i := i+1 OD

PROC P2 () =
     DO i//3#1 => i := i+1 OD

PROC P3 () =
     DO i//3#2 => i := i+1 OD
```

What is (are) the possible outcome(s) (that is, the final values of the state variables and any exceptions) of running `P1`, `P2` and `P3` concurrently? Give reasons.

Problem 2.

Consider the following set of Spec programs. Briefly explain what is (are) the possible outcome(s) (that is, the final values of the state variables and any exceptions) and the possible undesirable behavior(s) of running `P1` and `P2` concurrently in each case.

(A)

```
VAR c1: Bool := true
    c2: Bool := true

PROC P1() =
  DO ~c2 => SKIP OD;
  c1 := false;
  x := x + 1;
  c1 := true
```

```
PROC P2() =
  DO ~c1 => SKIP OD;
  c2 := false;
  x := x + 1;
  c2 := true
```

(B)

```
VAR c: Int
    x: Int := 0

PROC T1 () =
  c := 1;
  DO c=1 => SKIP OD
  x := x+1;
  c := 1
```

```
PROC T2 () =
  c := 2;
  DO c=2 => SKIP OD;
  x := x+1;
  c := 2
```

(C)

```
VAR c1: Bool := true
    c2: Bool := true
    turn: Int := 1

PROC P1() =
  c1 := false;
  DO ~c2 =>
       turn=2 => c1 := true;
                 DO turn=2 => SKIP OD;
                 c1 := false
       [*] SKIP
  OD
  x := x + 1;
  turn := 2;
  c1 := true
```

```
PROC P2() =
  c2 := false;
  DO ~c1 =>
      turn=1 => c2 := true;
                DO turn=1 => SKIP OD;
                c2 := false
      [*] SKIP
  OD
  x := x + 1;
  turn := 1;
  c2 := true
```

Problem 3.

Consider the following Spec programs:

```
PROC P () -> Int =
     VAR i: Int, j: Int |
          j< 10 => BEGIN i*j>0 => RET 1 END
          [*]
          j>-10 => BEGIN
                         i*j<0 => RET 2
                         [*]
                         BEGIN i>= 0 => RET 3 [*] RET 4 END
                   END
```

(A) What is (are) the possible outcome(s) (that is, the final values of the state variables and any exceptions) of running `P`. Can `P` deadlock?

(B) How does your answer to part A change if the **first** `[*]` in `P` is replaced by `[]`?

(C) How does your answer to part A change if `P` is made an `APROC` and atomicity brackets added to its body?

Problem 4.

Lampson's "Hints for Computer System Design" paper (Handout 6) divides the hints into three categories: functionality, speed, and fault-tolerance. Pick any two hints from two different categories, and describe how the designers of the BSD Log-Structured File System (Handout 10) applied the hints or failed to apply them. Answers should be a few paragraphs long; it is not enough to say "... is a good feature" or "... is undesirable."