Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science
6.826 Principles of Computer Systems

PROBLEM SET 3

Issued: D a y 5 Due: D a y 7

The following problems explore proof techniques for showing that one module implements another module.

There are two problems in this problem set; please turn in each problem on a separate sheet of paper. Also give the amount of time you spend on each problem.

Problem 1. Turtle Robot

A turtle robot can move freely in a two-dimensional plane. The following module specifies robot's behavior.

```
MODULE Turtle EXPORT Move, Position

TYPE Coord = [x: Int, y: Int]
     Path = SEQ Coord

VAR p: Path := {}

APROC Move(dx: Int, dy: Int) =
  << p := p + { Coord(x:=dx, y:=dy) } >>

FUNC sumdx() = + : (p * (\ coord | coord.x))
FUNC sumdy() = + : (p * (\ coord | coord.y))
FUNC Position() -> Cord = Coord(x:=sumdx(), y:=sumdy())
```

The robot is controlled by an embedded processor that has limited memory.

a) Write an implementation `TurtleImpl` that implements the `Turtle` specification and avoids storing the entire path of the robot.
b) Prove that your `TurtleImpl` implements `Turtle`.

Problem 2. Lossy Memory

Consider a memory that sometimes silently refuses to perform a write to a location. The specification of this memory is given by the following module, which is a modification of the `Memory` module on page 3 of the Handout 5.

```
MODULE LMemory [A, V] EXPORT Read, Write =

TYPE M = A -> V
VAR m := Init()

APROC Init() -> M = << VAR m' | (ALL a | m'! a) => RET m' >>

FUNC Read(a) -> V = << RET m(a) >>
APROC Write(a, v) = << m(a) := v [] SKIP >>
END Memory
```

a) Write a write-back cache implementation `LWBCache` of `LMemory`. Assume that a write to the cache always succeeds (provided that the cache is not full), so do not lose data when writing to the cache. Assume further that writing the data back to the underlying memory may result in writes being silently skipped.

b) Prove that your `LWBCache` implements `LMemory` using the appropriate proof technique.