Massachusetts Institute of Technology
6.852: Distributed Algorithms
Prof. Nancy Lynch

Handout 13
November 5, 2009

Problem Set 5, Part a

Due: Thursday, November 19, 2009

Reading:

Herlihy-Shavit book, Chapter 7;
Mellor-Crummey and Scott paper;
Magnussen, Ladin, and Hagersten paper (optional); Chapter 11 (skim)
Chapter 12.

Reading for next week:

Chapter 13

Problems:

1. Prove that Anderson's array algorithm implements a queue lock by giving a simulation relation from that algorithm to the QueueME automaton in the book. This means you need to write automaton code for Anderson's algorithm. You should use Tempo to typecheck your code. You may find it easier to do the proof assuming an infinite array first.
2. Give an execution of Mellor-Crummey and Scott's MCS algorithm in which a process must wait (i.e., spin at the *waitfor* statement) in its exit region. Why does this not violate the progress condition for exiting the critical section? Assuming an upper bound ℓ on the local step time of any process (including access to shared memory), can you bound the time a process may have to wait in its exit region?
3. Exercise 12.2.
4. Exercise 12.5.

How to declare shared variables in tempo?

Short answer: Tempo has no built-in support for shared variables.

Declare them as regular variables and use tempo comments to indicate they are shared. e.g.:

```
    myvar1: Nat := 0;
    myvar2: Nat := 0;
%   ********* START OF SHARED VARIABLES ********* %
    myshared1: Nat := 0;
    myshared2: Nat := 0;
%   ********* END OF SHARED VARIABLES   ********* %
```

6.852J / 18.437J Distributed Algorithms

Fall 2009