Marcel Zauder 16-124-836
Pascal Gerig 16-104-721

3.1 Several Questions

a) *What does a safety property do in FSP?*
A safety property in FSP is an expression that defines which actions are allowed. Actions that were not specified within a safety property are considered to lead to an ERROR.

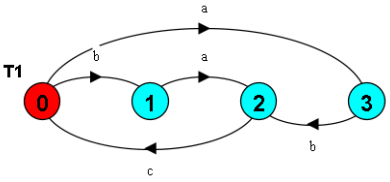
b) *Is the busy-wait mutex protocol fair? Justify your answer.*
Yes, it is, because assuming that P1 wants to enter the critical section, but can't because P2 is currently in the CS. Therefore *enter1* is TRUE, *enter2* is TRUE, and *turn* is P2. P1 is therefore currently in the busy wait loop. Eventually P2 will terminate its CS, assuming that none of the processes crashes, otherwise no progrss could be made if P2 crashes while in the CS, and set *enter2* to false. Therefore P1 will leave the busy loop and enter the CS. If P2 wants to enter the CS again it will wait until P1 has finished its CS.

c) *Can you ensure safety for shared data in concurrent programs without using any kind of locks?*

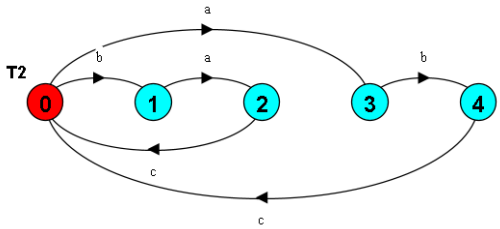
d) *The Java language designers decided to implement concurrency based on monitors. What is the main reason behind this decision?*

3.2 FSP Processes

As shown below with the LTSA tool both process definitions are not equal, because in T1 *c* is a shared action, whereas in T2 it is a seperate action for both traces.



R = (a->c->R).
S = (b->c->S).
||T1 = (R || S).



T2 = (a->b->c->T2 | b->a->c->T2).

3.3 FM Radio

3.4 Stack Implementation