Assignment 02 - Formal Methods

Raphaelle Akhras

March 2018

Discussion

Unlike the first assignment, this graph will include liveness and another feature which allows a process to be in a state where it is not requesting to enter the critical section at any moment in time. I will discuss this further in the upcoming paragraphs.

The point of this assignment was to add that any process can have a path where the the process will globally not request to enter the critical section. This is expressed through EG (N1) & EG (N2).
Liveness as mentioned in assignment 1 will be achieved through
(AG (T1 => (AF (C1)))) & (AG (T2 => (AF (C2))))

It is important to state before discussing the process in details that the process may be in 3 different states. Non-requesting (N), requesting (T), and in the critical section (C). There is also the flag for entering the critical section that must be acquired (F). All these will be accompanied by 1 or 2 denoting the process.

To take you through my graph step by step, I first stated that process 1 and 2 will be in a non-requesting state (N1,N2). From this state, we can go to two different states depending on who is requesting to enter the critical section. If we pick to go to the state where process 1 requests to enter the critical section (without loss of generality), it will look like (T1,N2) which means process 1 is requesting to enter the critical section and process 2 is in a non-requesting state. From this state, You can go to 1 other state which is to grant this process the flag to enter the critical section (T1,N2,F1).
From this state, you can go to two other states based on what is occurring in the program. Either process 2 can request also to enter the critical section or process 2 will remain non-requesting and thus the other state would be where process 1 is granted access to the critical section through the flag and is now in the critical section. The first state is (T1,T2,F1) and the other is (C1,N2,F1).

From the critical section, the process will then exit the critical section in the next state and then will release the flag (N1,N2,F1) then (N1,N2); and here it is clear that there exists a path globally where N2 is true and the same will be true for the other side where process 2 requests to enter the critical section. There is still the path from (T1,T2,F1). This means that both are requesting but only process 1 has the flag thus the next state will be one where process 1 enters the critical section but process 2 is still requesting and waiting (C1,T2,F1). Next, (N1,T2,F1) means that process 1 will finished from the critical section and it ready to release the flag.
Liveness is ensured because after this process 1 may not directly enter the critical section without releasing the flag so then process 2 has an equal change of getting the flag to enter the critical section. After this, since process 2 is requesting (C1,T2,F1), the next the next state will be to allow process 2 to enter the critical section but first it must release the flag (N1,T2). From here it gets the flag (N1,T2,F2) and the like process 1, it might enter the critical section while process 1 is non-requesting (N1,T2,F2), or process 1 may be requesting (T1,T2,F2). This is all like process 1. This will ensure that globally there will be a path where N1 is true.

I believe that I am satisfying all the options for the processes allowing them at any state, where it is logical, to complete any of the actions they desire. An example of this is where process 1 has the flag, but i still allow a state where process 2 is requesting and process 1 has the flag.
The reason why there is no state (T1,T2) because if this state is added, liveness will not be ensured since it creates a loop among (N1,N2), (T1,N2), (N1,T2), and (T1,T2). It results in an incorrect model after Model Repair with the liveness equation we provided.
In this model, I believe all possible possibilities have been taken care of without affecting liveness.