Question 1

To check for 1 channel having no communication aborted, which means that any communication that sends a start signal will eventually send a finish signal:

There are 4 channels deployed in this system: up[0], up[1], down[0], down[1]. To check for any channel having no communication aborted, the above check will be checked against all channels.

Assuming that all messaging channels are lose-less, checking the above LTL property shows that the above property holds, therefore, there is no communication aborted in the protocol.

Text

Description automatically generated

Question 2

1. To check if the solution preserves mutual exclusion, both processes should not have access to the critical section at the same time. I have created a local variable “localkey” for each process and if the “localkey” is set to true, it is currently executing the critical section, otherwise it will be set to false.

Therefore, to check using the LTL property, I will check that both processes should not have “localkey” set to true at the same time.

By checking this LTL property against the SMV model, I can verify that the solution preserves mutual exclusion.

Text

Description automatically generated

1. However, this solution is not a good solution. 1 process can keep acquiring the lock, starving the other process from the critical section. This LTL property can check if both processes eventually acquires the lock and run the critical section:

This property is not true given the counterexample below.

1. Process 1 acquires the lock.
2. Process 2 waits for the lock.
3. Process 1 releases the lock.
4. Loop back to 1.

Therefore, I can conclude that starvation occurs in this mutual exclusion solution.

Text

Description automatically generated

Question 3

In my solution, each process keeps track of its state.

States:

* c0: The process is running c0 codes.
* c1: The process is running the first if-statement of c1.
* crit: This process is running the code marked as */\* critical section \*/*.

The codes */\* non-critical section \*/* is not given any state as it has been abstracted out from the SMV solution. This will not cause any change to the mutual exclusion problem.

1. To check if the solution preserves mutual exclusion, both processes should not have access to the critical section at the same time. To check using the LTL property, I will check that both processes should not have “crit” state at the same time.

By checking this LTL property against the SMV model, I can verify that the solution preserves mutual exclusion.

Text

Description automatically generated

1. To prove that this solution is starvation free, assuming that both processes have a fair chance of executing, I can check it using the LTL property that eventually both processes should be able to run the “crit” state.

By checking this LTL property against the SMV model, I can verify that the solution is starvation free.

Text

Description automatically generated