

Lab Sheet: 5.2.2021

Task 0: Turn the CSP-M scripts from Canvas into files

Copy the CSP-M scripts from Canvas and save them into 6 files with names ending in “.csp”.

Task 1: Validation of the various ATM models (about 12-15 mins)

For ATM0, ATM1, ATM2, ATM4, ATM5, and ATM6 do the following:

- a) Write down an ‘expectation’ from the narrative that you were given on the slides.
- b) Validate if the model fulfils this expectation by
 - Either using the **:graph** command in FDR4 (which displays the labelled transition system of the process) or
 - Using the **:probe** command in FDR4 (which allows you to simulate the model).

What makes us sure that the model has the expected behaviour?

Task 2: Adding a new behaviour (about 10 mins)

Extend ATM4 with an alternative to print the balance. To this end,

- you will first need to add an event to the Display channel, add a channel Printer, add an event printSlipO.
- Then you can change the process ATM4.

Simulate, your process. Does it behave as expected?

Task 3: Observing internal and external choice and the effect of hiding (about 5 mins)

CSP distinguishes between observable events and non-observable events. The events we are declaring via the channel mechanism are all observable ones. One non-observable event is called tau, written as the Greek letter τ . In the graph of a process, there are two sources that can produce a τ :

- The internal choice operator.
- The hiding operator.

The event τ marks a system evolution that is not visible from an outside perspective. A typical example for such system evolution would be, say, an automated security update on a computer or phone: clearly, the update changes the system – after the update you are better protected against security attacks – but the user does not see it.

In this task, you shall inspect labelled transition system of processes in order to observe situations in which τ can arise.

- a) Consider the labelled transition systems that you obtain with the `:graph` command from the processes `PinVerification` (which belongs to the file `ATM6`) and `UserDialog` (which also belongs to the file `ATM6`). Spot how the internal choice operator differs from the external choice operator. Consider the labelled transition system of `ATM6`: what kind of choice do we obtain when running `UserDialog` and `PinVerification` in parallel? Any τ around? Why would such τ make sense in the combined process?
- b) Consider the labelled transition systems that you obtain with the `:graph` command from the processes `ATM6` and `ATM7`. Note how events that are observable in the labelled transition system in `ATM6` are turned into τ in the labelled transition system in `ATM7`.