Practical Class 1 — Using ProBE[1]

ProBE (Process Behaviour Explorer) is a piece of software which allows CSP definitions to be examined. This practical class introduces ProBE and uses it to explore the behaviour of the CSP processes which were defined in the lecture.

Getting Started

1. Create a directory in your home directory for CSP work, and move into it:

   ```
   mkdir CSP
   cd CSP
   ```

2. Start ProBE in the background:

   ```
   probe &
   ```

   You will see the initial window of ProBE, entitled "Launcher (no file)".

3. Copy the file `vm1.csp` to your CSP directory.

4. Start an `emacs` window in the background (or another editor, if you prefer) and view `vm1.csp`, which contains the following lines of machine-readable CSP:

   ```
   channel coin, choc

   VM = coin -> choc -> STOP
   ```

   The first line declares the events *coin* and *choc* (machine-readable CSP calls them *channels*, for reasons which will become clear later).

   The next line defines the process VM, as in the lecture.

[1] This practical sheet is provided as course material to accompany the book 'Concurrent and Real Time Systems'.

5. In the "Launcher" window, select "Open" from the "File" menu. You will get a window which allows you to select files from your directory. Select `vm1.csp`. The "Launcher" window will then show the process name VM, which is the only process defined by the file `vm1.csp`.

6. Double click on VM. You will get another window, entitled "Exploring VM". The name VM appears in a white region, and below it you will see the event name *coin* with a + before it. This indicates that the process VM can do the event *coin*.

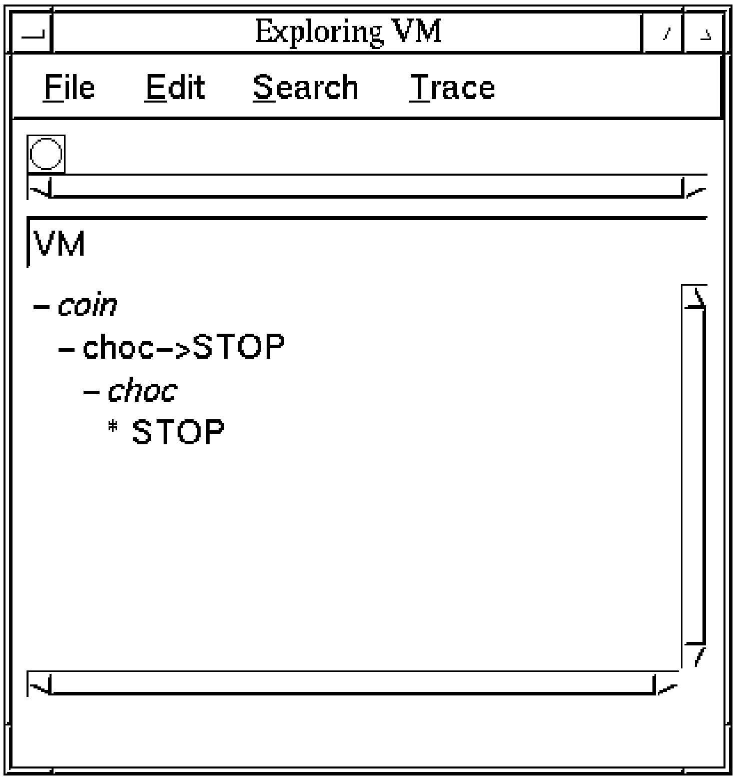
7. Click on the + next to *coin*. This shows you what happens when VM does the *coin* event — you will see that $choc \rightarrow STOP$ appears below *coin*. The + next to *coin* changes to a − . If you click on the − then the view goes back to how it was before, and if you then click on the + again, it will be expanded once more.

   Notice that *coin* is displayed in *italics*, because it is an event, and $choc \rightarrow STOP$ is displayed in normal style, because it is a process.

8. Click on the + next to $choc \rightarrow STOP$. *choc* will appear below it, to indicate that the process $choc \rightarrow STOP$ can do the event *choc*.

9. Expand *choc* (click on the +) and you will see $STOP$ below it. $STOP$ has a * next to it, which indicates that it cannot be expanded any further.

10. The explorer window should now look like this:



Contract and expand the events and processes a few times, until you are comfortable with the operation of the + and − marks.

11. Expand the view of *VM* completely, and double click on *STOP*. The white region of the window will now show *Stop*, and above it you will see three green circles separated by arrows. Each circle represents a process (or *state* of the system) and each arrow represents an event. They are not labelled, but clicking on a circle or arrow will show you its name in the white area.

12. Double clicking on a circle or arrow moves shows its name in the white area, and shows either the next event or the next process below, so that it can be expanded. Experiment with selecting various events and processes, and expanding and contracting the view, to see the various ways in which the *VM* process can be visualised.

Examining Traces

1. Copy the file `vm2.csp` into your directory. Load it into your editor. It contains the following definitions, which correspond to the reusable vending machine from the lecture.

   ```
   channel coin, choc

   VM = coin -> choc -> VM
   ```

2. In the launcher window, use “Open” from the “File” menu to load `vm2.csp`, then select (double click) the *VM* process as before. You will get a new explorer window, and you can close the old one by selecting “Close” from the “File” menu.

3. Explore the new definition of *VM* as before. Now that the process can continue performing events forever, the view of its behaviour can be expanded as much as you like.

4. As before, double clicking on the name of a process puts it into the white area and generates a sequence of circles and arrows above it. Now select “View” from the “Trace” menu. You will get a new window, entitled “Trace display”, which shows a vertical list of events. This is a *trace* of the process *VM*, i.e. a sequence of events which the process can do. In fact it is the sequence of events which lead from the initial state of *VM* to the state which is displayed in the white area of the explorer window.

5. Experiment with selecting different parts of the process, and viewing the traces. You will get a new trace display window every time; unwanted windows can be closed by selecting “Close” from the “File” menu.

Processes with Choice

1. Copy the file `tickets.csp` into your directory and load it into ProBE. Select the process *MACHINE*. Load the file into your editor — it contains the definitions

```
channel on, off, staines, ashford, pound, ticket

MACHINE = on -> TICKETS

TICKETS =   staines -> pound -> ticket -> TICKETS
         [] ashford -> pound -> pound -> ticket -> TICKETS
         [] off -> MACHINE
```

Notice that `[]` is used for choice, instead of `|`. The reason for this will eventually become clear.

2. Explore the process *MACHINE* as before. You will see that expanding *TICKETS* produces a choice of 3 events, any of which can be expanded. Try viewing traces of some of the processes.