Semantics, concurrency and verification

A short introdukction to the Concurrency Workbench (CWB)

Hans Hüttel

February 1998

The Concurrency Workbench (CWB) is a system which can check many different kinds of correctness of CCS processes. This includes checking for strong and weak bisimulation equivalence and model checking recursive Hennessy-Milner formulae. CWB also lets you simulate the behaviour of CCS processes.

The Concurrency Workbench documentation can be found in `/pack/FS/CWB/doc/`. A number of examples can be found in `/pack/FS/CWB/examples`.

1 Getting started

I would recommend that you run CWB from within Emacs, as there is a special Emacs model. Add the line

```
(load "/pack/FS/CWB/utilities/aau-cwb.el")
```

to your `.emacs` file; when you start Emacs anew, this will let you run CWB within Emacs.

2 Running the Concurrency Workbench

Start up Emacs, once you have included the line mentioned above to your `.emacs` file. Then type

```
M-x cwb
```

You should now get the CWB prompt:

```
Command:
```

You are now using the special Workbench mode for Emacs. This mode is documented like any other Emacs mode. What follows is a short overview of some important things that you will need to know in order to proceed from here.

2.1 Loading files

The Emacs command

```
C-c C-l
```

lets you load files containing definitions of CCS processes (and logical specifications. Such files should have the suffix `.cwb`. Sample files can be found via the course homepage (cf. section 2.2 below).

2.2 Writing process expressions

The directory

```
http://www.cs.auc.dk/~hans/Dat4/cwb-eksempler/
```

contains three examples of CWB versions of CCS processes, all of whom are taken from Milner, chapter 1. `new-vending.cwb`, is the vending maching from section 1.2, `semafor.cwb` containes the semaphores from sections 1.3 and 1.4 and `jobshop.cwb` is the legendary jobshop from section 1.3.

2.3 Checking correctness

The CWB contains a plethora of commands for checking various kinds of correctness. You can e.g. check for strong bisimulation equivalence using the command `strongeq` and obtain the maximal bisimulation containing any given pair of processes (provided that such a bisimulation exists). To do this, use the command `pb`. The precise syntax of these and other commands can be obtained using the on-line manual.

2.4 Help!

The CWB has an interactive on-line manual, which you can get access to by typing `help`. The manual contains information about CCS syntax (as seen from the CWB), commands etc.

3 Exercises

1. Load one of the examples mentioned above and use the CWB to find the transitions of the system in question.

2. Use the CWB to check the correctness claim for semaphores on page 33 of 'Communication and Concurrency' for the case $n = 3$.

3. Use the CWB to check the claim immediately below the figure on page 41 of 'Communication and Concurrency', namely that

$$(A \mid B) \setminus c \sim C_0$$

   where $\sim$ denotes strong bisimulation equivalence and A, B and C_0 are as defined in section 2.2. Also find the transitions of the two processes by using the CWB.