Practical Class 6 — Deadlocks[1]

This practical class uses FDR to analyse systems for deadlock.

Getting Started

1. Start FDR and an editor.

The Dining Philosophers

1. Copy the file `dphil.fdr2` to your directory, and load it into your editor and FDR. It contains the definitions for the dining philosophers, as in the lecture notes.

2. Click on the "Deadlock" tab in the main window. Select "COLLEGE" from the "Implementation" menu. Click on "Check". After a pause, you should see a red cross next to "COLLEGE deadlock-free" in the lower window, indicating that a deadlock has been discovered.

3. Double-click on the "COLLEGE deadlock-free" line, to bring up a debug window. (Alternatively, single-click, then select "Debug" from the "Assert" menu.) The debug window shows a trace which leads to a deadlock. This is not necessarily the only possibility of deadlock; simply the first one which FDR finds.

4. The file `dphil_butler.fdr2` contains the definitions for the dining philosopher with the butler added, from lectures. Use FDR to check NEWCOLLEGE for deadlock; you should find that it is deadlock-free.

5. Use ProBE to explore the behaviour of NEWCOLLEGE, to see how deadlock is avoided.

6. The other suggestion for avoiding deadlock in the dining philosophers example was to make one of the philosophers pick up the forks in the opposite order. Modify the definitions in `dphil.fdr2` so that they implement this suggestion, and check for deadlock again.

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