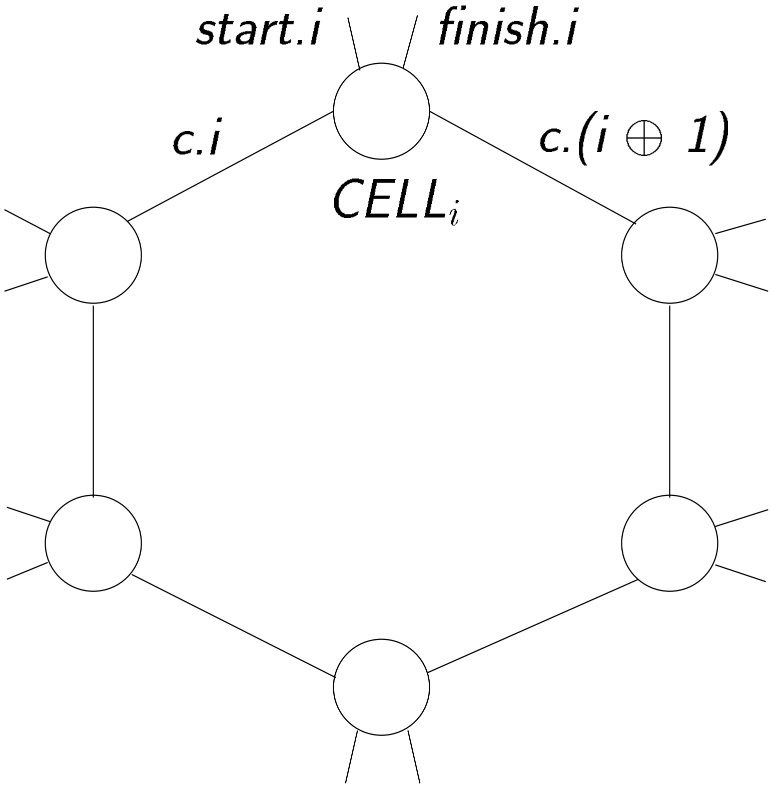
◇ The Cyclic Scheduler ◇

Suppose there are a number of processes which we need to control. Each process can be started by a *start* event and uses a *finish* event to indicate that it has finished. Suppose also that we want to start the processes in order, returning to the first when the last has been started; when a process has finished it can be started again, but only when its turn comes round.

We will define a *scheduler*, which uses *start* and *finish* events to control the processes. The scheduler will be implemented as a collection of cells (processes), each of which communicates with one of the processes being controlled, and also with other cells.

The next slide has a diagram of the case where there are 6 processes to control. $\oplus$ denotes addition modulo the number of processes.

The idea is that each cell waits for a signal on the c channel to its left, which means that the process to its left has been started. Then the cell starts its own process, and sends a signal on the c channel to its right, to tell the next cell that it can start its process. Also, each cell has to wait for its process to do *finish* before starting it again.



In order to start everything off, one cell must begin by starting its process instead of waiting.

$$\begin{aligned} \mathit{STARTCELL}_i \;=\;& \mathit{start.i} \rightarrow \mathit{c.}(i \oplus 1) \rightarrow \\ & (\;\; \mathit{finish.i} \rightarrow \mathit{c.i} \rightarrow \mathit{STARTCELL}_i \\ & \;\Box\;\; \mathit{c.i} \rightarrow \mathit{finish.i} \rightarrow \mathit{STARTCELL}_i) \end{aligned}$$

The other processes wait for $c.i$

$$\mathit{WAITCELL}_i = \mathit{c.i} \rightarrow \mathit{STARTCELL}_i$$

It is convenient to define

$$\begin{aligned} \mathit{CELL}_0 &= \mathit{STARTCELL}_0 \\ \mathit{CELL}_i &= \mathit{WAITCELL}_i \quad (i > 0) \end{aligned}$$

and then

$$\begin{aligned} \mathit{SCHED} = \;& (\|^{0 \leqslant i < n}_{\{\mathit{start.i}, \mathit{finish.i}, \mathit{c.i}, \mathit{c.}(i \oplus 1)\}} \mathit{CELL}_i) \setminus \\ & \{c.i \mid 0 \leqslant i < n\} \end{aligned}$$

There are three properties which we would like to verify for the scheduler. The first is that for each i, the events *start.i* and *finish.i* happen alternately, beginning with *start.i*. The second is that the events *start.0*, ..., *start.(n-1)* happen in the correct cyclic order. The third is deadlock-fredom.

For the first property, we can define a process specifying alternation of *start* and *finish* for each cell:

$$ALT_i = start.i \to finish.i \to ALT_i$$

and combine them in parallel to produce a specification for the scheduler as a whole.

$$ALTSPEC = \|_{\{start.i, finish.i\}}^{0 \leqslant i < n} ALT_i$$

In this parallel combination the alphabets are all disjoint, and no synchronisation is required. It is simply an independent parallel combination of the *ALT* processes.

The specification

$$ALTSPEC \sqsubseteq_t SCHED$$

can be checked with FDR.

For the second property, define

$$CYCLE_i \ = \ start.i \to CYCLE_{i \oplus 1} \ \ (0 \leqslant i < n)$$

and specify

$$CYCLE_0 \sqsubseteq SCHED \setminus \{finish.i \mid 0 \leqslant i < n\}.$$