Assignment 9: CSP

ETH Zurich

1 Moving objects

1.1 Background

This task has been adapted from [1]. An object starts on the ground, and may move up. At any time thereafter it may move up and down, except that when on the ground it cannot move any further down. But when it is on the ground, it may move around. Let n range over the natural numbers $0, 1, 2, \ldots$.

1.2 Task

For each n, introduce the process CT_n to describe the behaviour of the object when it is n moves off the ground.

1.3 Solution

$$CT_0 = (up \rightarrow CT_1 \,\Box\, around \rightarrow CT_0)$$
$$CT_{n+1} = (up \rightarrow CT_{n+2} \,\Box\, down \rightarrow CT_n)$$

2 Barriers and Workers

2.1 Background

In this task, we focus on worker-processes. The definition for a worker is a process that continually: joins a barrier with a tagged announcement, then leaves the barrier, then does some work. This can be given as a CSP process, $M_i = join_i \rightarrow leave \rightarrow work \rightarrow M_i$.

2.2 Task

1. Give a process with two workers composed together such that the workers can join the same barrier whenever they wish, but must leave together before independently doing their work.

2. Simplify the process from the above task (you can stop when you see a non-deterministic internal choice operator).

3. Give a process which has two teams of two workers. The teams must all synchronize to the same barrier, the members of the team must work together, but different teams work independently.

2.3 Solution

1. $M_1 \parallel_{\{leave\}} M_2$

2.

$$\begin{aligned}
X &= M_1 \parallel_{\{leave\}} M_2 \\
&= (join_1 \to leave \to work \to M_1) \parallel_{\{leave\}} (join_2 \to leave \to work \to M_2) \\
&= \Big(join_1 \to \Big((leave \to work \to M_1) \parallel_{\{leave\}} (join_2 \to leave \to work \to M_2)\Big)\Big) \,\Box \\
&\quad \Big(join_2 \to \Big((join_1 \to leave \to work \to M_1) \parallel_{\{leave\}} (leave \to work \to M_2)\Big)\Big) \\
&= \Big(join_1 \to join_2 \to \Big((leave \to work \to M_1) \parallel_{\{leave\}} (leave \to work \to M_2)\Big)\Big) \,\Box \\
&\quad \Big(join_2 \to join_1 \to \Big((leave \to work \to M_1) \parallel_{\{leave\}} (leave \to work \to M_2)\Big)\Big)
\end{aligned}$$

Then we let $Y = (leave \to work \to M_1) \parallel_{\{leave\}} (leave \to work \to M_2)$.

$$\begin{aligned}
Y &= (leave \to work \to M_1) \parallel_{\{leave\}} (leave \to work \to M_2) \\
&= leave \to \Big((work \to M_1) \parallel_{\{leave\}} (work \to M_2)\Big) \\
&= leave \to work \to \Big(\Big(M_1 \parallel_{\{leave\}} (work \to M_2)\Big) \sqcap \Big((work \to M_1) \parallel_{\{leave\}} M_2\Big)\Big)
\end{aligned}$$

3. $\Big(M_1 \parallel_{\{work,leave\}} M_2\Big) \parallel_{\{leave\}} \Big(M_3 \parallel_{\{work,leave\}} M_4\Big)$

References

[1] C.A.R. Hoare. Communicating Sequential Processes. Prentice Hall, 1985.