INFR11248: Modelling Concurrent Systems
University of Edinburgh
2024-2025 SEM1

Homework 11

Gautam Yajaman

1. There are two runners R_1 and R_2 participating in the relay race, and while running they carry a baton. Runner 1 starts running, and at the halfway-point passes the baton to runner 2, who is waiting there. Then runner 2 runs the rest of the race and delivers the baton at the finish.

   We can describe the behaviour of the two runners by the CSP expressions:

   $$R_1 = start \to give \to STOP$$
   $$R_2 = give \to finish \to STOP$$

   Here *start* is the action of runner 1 starting the race and *finish* is the action of runner 2 of arriving at the finish. The action *give* represents the two runners attempting to synchronise a handover of the baton. In both cases, $STOP$ indicates that the given process is no longer doing any actions.

   A CSP expression describing the whole interaction is:

   $$(R_1||_{\{give\}}R_2)/\{give\}$$

   The subscript on the parallel composition indicates actions that **must** be synchronised. Any instance of *give* by both R_1 and R_2 must be synchronised, giving rise to a sychronised action *give* which would effectively do both of these transitions at the same time. Here, the encapsulation function ∂ is implicit in the parallel composition and eliminates all instances of *give* which are not the result of synchronisation. The operator / replaces all instances of *give* with τ, abstracting the handover.

   The final expression which we get is $start \to \tau \to finish \to STOP$. However, we can choose to focus on just the start and finish, and abstract this expression to $start \to finish \to STOP$

2. $(a \Box b) \sqcap c$ represents an internal choice being made, followed by an external choice. This can be written in CSS as $(\tau.(a + b) + \tau.c)$. The system internally chooses whether to give us the ability to choose between a and b or to give us c.

   $(a \sqcap b)\Box c$, we cannot naively write this as $(\tau.a + \tau.b) + c$ as the τ transitions would take away our ability to choose c. The idea is that we must be able to choose the option c or the option for the system to give us a or b, so if the system chooses a, it must preserve our option to choose c (a symmetric argument is made for b).

   Hence the expression must be $(\tau.(a + c) + \tau.(b + c)) + c$

3. Note that the expression $Y = (b.Y|d.Y|e.Y)\backslash\{d\}$ is a guarded recursive specification.

   RDP says that each recursive specification has a solution up to $\sim$, where $\sim$ is some equivalence.