Postgraduate Course: Process Algebra

Communicating Sequential Process

Supplementary Materials
(Case Study One)

Huibiao Zhu

Software Engineering Institute, East China Normal University

Case Study Explanation

We study the paper written by A. W. Roscoe:

Paper name: A CSP solution to the train problem

Author: A. W. Roscoe

Institute: Programming Research Group, Oxford University

Line

For each line λ, whose endpoints are α, β, we define

$$
\begin{aligned}
LINE(\lambda) &= get.\lambda.\alpha \rightarrow con.\lambda.\alpha \rightarrow BUSY(\lambda) \\
&\quad \Box get.\lambda.\beta \rightarrow con.\lambda.\beta \rightarrow BUSY(\lambda) \\
BUSY(\lambda) &= get.\lambda.\alpha \rightarrow dis.\lambda.\alpha \rightarrow BUSY(\lambda) \\
&\quad \Box get.\lambda.\beta \rightarrow dis.\lambda.\beta \rightarrow BUSY(\lambda) \\
&\quad \Box rel.\lambda.\alpha \rightarrow LINE(\lambda) \\
&\quad \Box rel.\lambda.\beta \rightarrow LINE(\lambda)
\end{aligned}
$$

Train (1)

The process $TRAIN(t, l)$ represents train t running along the directed line l, with no other line booked. The process $TRAIN(t, l, l')$ (where $E(l) = S(l')$) represents train t running along line l with line l' booked at $E(l)$.

Thus, when $\beta = E(l)$, $E(l) = S(l')$, we define

$$
\begin{aligned}
& TRAIN(t, l) \\
= & \ reverse.t \to TRAIN(t, \bar{l}) \\
& \Box(\Box_{S(m)=\beta}\, book.t.m \to req.t.m \to \\
& \qquad (con.t.m \to booked.t.m \to TRAIN(t, l, m) \\
& \qquad \Box ref.t.m \to refused.t.m \to TRAIN(t, l)))
\end{aligned}
$$

Train (2)

$$TRAIN(t, l, l')$$
$$= reverse.t \to rel.t.l' \to TRAIN(t, \bar{l})$$
$$\Box (goto.t.\beta \to arrive.t.l \to leave.t.l \to rel.t.l \to TRAIN(t, l'))$$

Note: Communication with driver has one of the forms ”reverse”, ”book”, ”booked”, ”refused”, *goto*. All other events are communications with a crossing point.

Crossing Points (1)

Crossing points also have unique names (typicaly α, β). they have two roles. Firstly they act as intermediaries between TRAINs and LINEs. Secondly they only control the timing of transitions from one line to another—specially they only allow one train to be using them at any time (to avoid crashes).

Crossing Points (2)

$CP(\alpha) = CP1(\alpha) \parallel CP2(\alpha),\quad \text{where}$

$$CP1(\alpha) = \Box_{S(l)=\alpha, t\in T}\ req.t.l \rightarrow get.(N(l)).\alpha \rightarrow$$
$$(con.(N(l)).\alpha \rightarrow con.t.l \rightarrow CP1(\alpha)$$
$$\Box dis.(N(l)).\alpha \rightarrow ref.t.l \rightarrow CP1(\alpha)\ )$$
$$\Box(\ \Box_{S(l)=\alpha, t\in T} rel.t.l \rightarrow rel.N(l).\alpha \rightarrow CP1(\alpha)\ )$$

$$CP2(\alpha) = \Box_{E(l)=\alpha, t\in T}\ arrive.t.l \rightarrow$$
$$(\ \Box_{S(m)=\alpha}\ enter.t.m \rightarrow leave.t.l \rightarrow CP2(\alpha)\ )$$

The Whole System

The solution to the problem is thus

LINES

$$TRAINS || CPS || LINKS$$

where $TRAINS$, CPS and $LINKS$ are respectively the parallel composition of all TRAINs, CPs and LINEs.

Two "User" Points of View to the Whole System

(1) The TRAINs can be thought of as using the "network" which consists of the CPs and LINEs.

$$NETWORK = (CPS||LINES) \setminus L$$

where L is the union of the alphabets of the LINEs.

(2) The Second level of user is provided by train drivers.

$$SYSTEM = (TRAINS \parallel NETWORK) \setminus B$$

where B is the set of all symbols of the form $req.t.m$, $con.t.m$, $ref.t.m$ or $rel.t.m$

Property Verification

Property 1: At no time can there be more than one train on any line.

$$s \in traces(SYSTEM) \Rightarrow$$

$$length(s \restriction \{leave.t.l,\ leave.t.\bar{l} :\ t \in T\})$$

$$\geq length(s \restriction \{enter.t.l,\ enter.t.\bar{l} :\ t \in T\}) - C$$

where C is 0 or 1 depending on whether there is or is not a train initially on l (or $\bar{l}$),