Encoding of CSP Square operator to ACP

Leon Lee

October 31, 2024

1.1 Prerequisites

Definition 1.1.1: Sets of A

- A is the set of actions
- A_0 is the set of actions that actually get used in processes
- $H_0 = A - A_0$ is the set of working space operators or any other action that doesn't get used
- $H_1 = A_0 \cup \{\text{first}, \text{second}, \text{stop}, \text{skip}, \text{next}\}$ is the set of actions, plus some working operators

Definition 1.1.2: F1

Define a function $f_1 : A_0 \to H_0$ where

$$f_1(a_{\text{first}}) = a_{\text{first}}$$
$$f_1(a_{\text{second}}) = a_{\text{second}}$$
$$f_1(a_{\text{next}}) = a$$

Definition 1.1.3: Communication

Define communications where

$$a|\text{first} = a_{\text{first}}$$
$$\text{skip}|\text{first} = \text{skip}$$
$$\tau|\text{first} = \tau$$
$$a|\text{second} = a_{\text{second}}$$
$$a|\text{next} = a_{\text{next}}$$

Definition 1.1.4: Triggering in ACP

$$\phi_p := \rho_{f_1}[\partial_{H_1}(p||\text{first}(\text{next}^\infty))]$$

is a function that turns a process $p = p_1.p_2.p_3.\ldots$ into

$$p = p_{\text{first}}.p_2.p_3.p_4.\ldots$$

$$\phi_p := \rho_{f_1}[\partial_{H_1}(p||\text{first}(\text{second}(\text{next}^\infty)))]$$

is a function that turns a process $p = p_1.p_2.p_3.\ldots$ into

$$p = p_{\text{first}}.p_{\text{second}}.p_3.p_4.\ldots$$

1.2 Encoding

1.2.1 Strategy

- If there is a τ, put the starts of both processes onto it
- Then, append the rest of both processes onto that as well
- Then add any processes without a τ on to the start

1.3 Way to filter out processes with tau

Add new communications

- $a_{\text{first}}|b_{\text{second}} = a_{\text{second}}$
- $\text{skip}|\tau = \tau$
- $\tau_s|a_s = \tau_s + a$

Define function stop with the communications

$$\text{stop}|a = a, \quad a \neq \tau$$

The function

$$\partial_{H_0}(\rho_{f_1}[\partial_{H_1}(p||\text{first}(\text{next}^\infty))]||\text{stop})$$

will return an empty process if the first action is τ otherwise return the process

$$\begin{tikzcd}
\bullet \arrow[r, "\tau_f"] \arrow[d, "stop"'] & \bullet \arrow[r, "a"] \arrow[d] & \bullet \arrow[r, "b"] \arrow[d] & \bullet \arrow[d] \\
\bullet \arrow[r, "\tau_f"] & \bullet \arrow[r, "b"] & \bullet \arrow[r, "c"] & \bullet
\end{tikzcd}$$

$$\begin{tikzcd}
\bullet \arrow[r, "a_f"] \arrow[d, "stop"'] \arrow[dr, "a"] & \bullet \arrow[r, "b"] \arrow[d] & \bullet \arrow[r, "c"] \arrow[d] & \bullet \arrow[d] \\
\bullet \arrow[r, "a_f"'] & \bullet \arrow[r, "b"'] & \bullet \arrow[r, "c"'] & \bullet
\end{tikzcd}$$

1.4 Cut Tau Tree

Recall: $a_{\text{first}}|b_{\text{second}} = a_{\text{second}}$

$$\rho_{f_1}(\partial_{H_1}(p||\text{first}(\text{second}(\text{next}^\infty)))) \;||\; \rho_{f_1}(\partial_{H_1}(q||\text{first}(\text{second}(\text{next}^\infty))))$$

$$\begin{tikzcd}
\circ \arrow[r, "\tau"] \arrow[d, "c_f"'] & \bullet \arrow[r, "a_s"] \arrow[d, "c_f"'] \arrow[dr, "c_s"] & \bullet \arrow[r, "b"] \arrow[d, "c_f"'] & \bullet \arrow[d, "c_f"'] \\
\bullet \arrow[r, "\tau"] \arrow[d, "d"'] & \bullet \arrow[r, "a"] \arrow[d, "d"'] & \bullet \arrow[r, "b"] \arrow[d, "d"'] & \bullet \arrow[d, "d"'] \\
\bullet \arrow[r, "\tau"] \arrow[d, "e"'] & \bullet \arrow[r, "a"] \arrow[d, "e"'] & \bullet \arrow[r, "b"] \arrow[d, "e"'] & \bullet \arrow[d, "e"'] \\
\bullet \arrow[r, "\tau"'] & \bullet \arrow[r, "a"'] & \bullet \arrow[r] & \bullet
\end{tikzcd}
\qquad
\begin{tikzcd}
\circ \arrow[r, "\tau"] & \bullet \arrow[r, "a_s"] \arrow[d, "c_s"'] & \bullet \\
& \bullet &
\end{tikzcd}$$

If neither process has τ then this will just do nothing since the a_{first} will just get deleted at the end
Red parts are just ignoring the irrelevant bits that don't have any further way of connecting

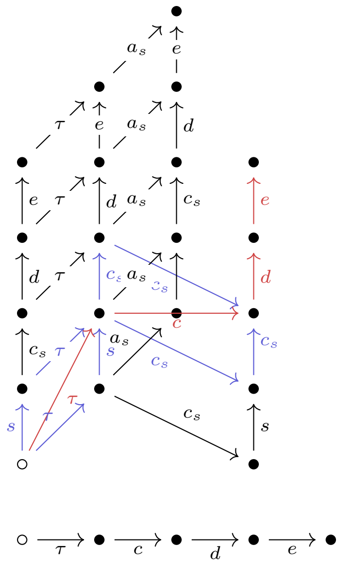
1.5 Reinserting the remaining states of P and Q

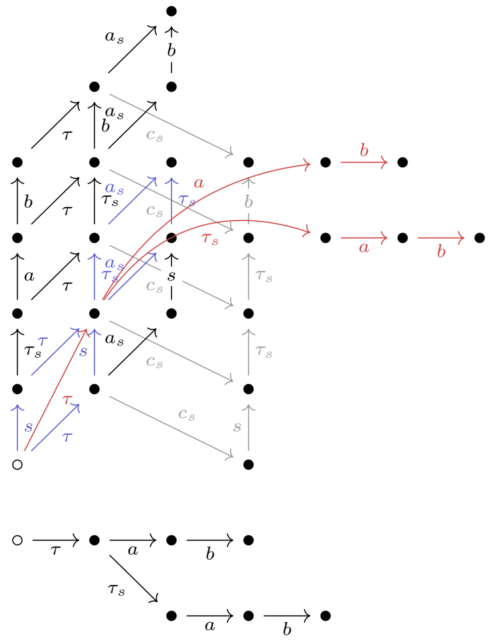
Recall: skip$|\tau = \tau$, $\tau_s | a_s = \tau_s + a$

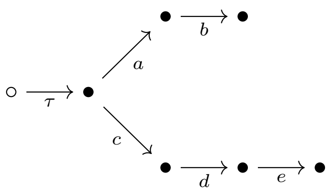
- things going up: both process with skip on the start and also prefixed

$$\rho_{f_1}(\partial_{H_1}(\text{skip}(p)||\text{first}(\text{second}(\text{next}^\infty))))$$
$$= \text{skip}.a_{\text{second}}.a_2.a_3.\ldots$$

- blue means a successful communication
- red means a communication path







1.6 Final conversion

Define

$$\Gamma_1(p) = \rho_{f_1}(\partial_{H_1}(\text{skip}(p)||\text{first}(\text{next}^\infty)))$$
$$\Gamma_2(p) = \rho_{f_1}(\partial_{H_1}(\text{skip}(p)||\text{first}(\text{second}(\text{next}^\infty))))$$

An encoding of the square operator in ACP is:

$$P\Box Q = \partial_{H_0}[\underbrace{(\Gamma_1(P)\ ||\ \text{stop})}_{\text{Filter from P}} + \underbrace{(\Gamma_2(Q)\ ||\ \text{stop})}_{\text{Filter from Q}} + (\underbrace{(\Gamma_2(P)\ ||\ \Gamma_2(Q))}_{\text{Cut Tau Tree}}\ ||\ \underbrace{\Gamma_2(\text{skip}(P)}_{\text{Reinsert P}}\ ||\ \underbrace{\Gamma_2(\text{skip}(Q))}_{\text{Reinsert Q}}))]$$