Confluence Detection

Wieger Wesselink

September 2019

1 Confluence

Consider the following untimed linear process specification $P$, with initial state $d_0$.

$$P(d) = \sum_{i \in I} \sum_{e_i} c_i(d, e_i) \rightarrow a_i(f_i(d, e_i)) \cdot P(g_i(d, e_i))$$

We distinguish different kinds of confluence. Let summand $j \in I$ be the index of a $\tau$-summand, and $i \in I$ be the index of an arbitrary summand. In the sequel we abbreviate $c_i(d, e_i) \cdot f_i(d, e_i)$ and $g_i(d, e_i)$ with $c_i, f_i$ and $g_i$. Note that for the $\tau$-summand with index $j$ we have $a_j = \tau$ and $f_j = \emptyset$.

1.1 Trivial confluence

Trivial confluence is defined as

$$C_{\text{trivial}}(i, j) = \forall d, e_i, e_j : (c_i \land c_j) \Rightarrow (a_i = \tau) \land (g_i = g_j)$$

Note that trivial confluence only applies to $\tau$-summands. In isolation it is not a very useful property to check.

1.2 Triangular confluence

Triangular confluence is defined as

$$C_{\text{triangular}}(i, j) = \forall d, e_i, e_j : (c_i \land c_j) \Rightarrow (c_i[d := g_j, e_i := e_j'] \land (f_i = f_i[d := g_j]) \land (g_i[d := g_j] = g_i))$$

1.3 Commutative confluence

Commutative confluence is defined as

$$C_{\text{commutative}}(i, j) = C_{\text{trivial}}(i, j) \lor \forall d, e_i, e_j : (c_i \land c_j) \Rightarrow \exists e_i', e_j :$$

$$\begin{align*}
&c_i[d := g_j, e_i := e_i'] \\
&\land c_j[d := g_i, e_j := e_j'] \\
&\land (f_i = f_i[d := g_j]) \\
&\land (g_i[d := g_j, e_i := e_i'] = g_j[d := g_i, e_j := e_j'])
\end{align*}$$

(1)

The reason for adding the term $C_{\text{trivial}}(i, j)$ is probably that otherwise a simple $\tau$-summand like

$$(n = 0) \rightarrow \tau \cdot P(n = 1)$$

is not even confluent with itself.
1.4 Square confluence

Square confluence is defined as

\[ C_{\text{square}}(i, j) = C_{\text{trivial}}(i, j) \lor \forall d, e_i, e_j : (c_i \land c_j) \Rightarrow c_i[d := g_j] \land c_j[d := g_i] \land (f_i = f_i[d := g_j]) \land (g_i[d := g_j] = g_j[d := g_i]) \]

It is obtained from \( C_{\text{commutative}}(i, j) \) by taking \( e'_i = e_i \) and \( e'_j = e_j \).
Figure 1: square commutative confluence, with $d_1 = g_j[d := g_i]$

Figure 2: triangular confluence

Figure 3: trivial confluence, with $a_j = \tau$