### 1 CCS

CCS \([4]\) is parametrised with a set \(\mathcal{A}\) of names. The set \(\mathcal{A}\) of co-names is \(\bar{\mathcal{A}} := \{\bar{a} \mid a \in \mathcal{A}\}\), and \(\mathcal{L} := \mathcal{A} \cup \bar{\mathcal{A}}\) is the set of labels. The function \(\bar{\cdot}\) is extended to \(\mathcal{L}\) by declaring \(\bar{a} = a\). Finally, \(\text{Act} := \mathcal{L} \cup \{\tau\}\) is the set of actions. Below, \(a, b, c, \ldots\) range over \(\mathcal{L}\) and \(\alpha, \beta\) over \(\text{Act}\). A relabelling function is a function \(f: \mathcal{L} \to \mathcal{L}\) satisfying \(f(\bar{a}) = \bar{f(a)}\); it extends to \(\text{Act}\) by \(f(\tau) := \tau\). Let \(\mathcal{X}\) be a set \(X, Y, \ldots\) of process variables. The set \(\mathcal{E}\) of CCS terms or process expressions is the smallest set including:

- \(\alpha.E\) for \(\alpha \in \text{Act}\) and \(E \in \mathcal{E}\) \(\quad\) prefixing
- \(\sum_{i \in I} E_i\) for \(I\) an index set and \(E_i \in \mathcal{E}\) \(\quad\) choice
- \(E[F]\) for \(E, F \in \mathcal{E}\) \(\quad\) parallel composition
- \(E[L]\) for \(L \subseteq \mathcal{L}\) and \(E \in \mathcal{E}\) \(\quad\) restriction
- \(E[f]\) for \(f\) a relabelling function and \(E \in \mathcal{E}\) relabelling
- \(X\) for \(X \in \mathcal{X}\) \(\quad\) a process variable
- \(\text{fix}_X S\) for \(S: \mathcal{X} \to \mathcal{E}\) and \(X \in \text{dom}\(S\)\) \(\quad\) recursion.

One writes \(E_1 + E_2\) for \(\sum_{i \in I} E_i\) with \(I = \{1, 2\}\), and \(0\) for \(\sum_{i \in \emptyset} E_i\). A partial function \(S: \mathcal{X} \to \mathcal{E}\) is called a recursive specification. The variables in its domain \(\text{dom}(S)\) are called recursion variables and the equations \(Y = S(Y)\) for \(Y \in \text{dom}(S)\) are recursion equations. A recursive specification \(S: \mathcal{X} \to \mathcal{E}\) is traditionally written as \(\{Y = S(Y) \mid Y \in \text{dom}(S)\}\).

The operational semantics of CCS is given by the labelled transition relation \(\to \subseteq T_{\text{CCS}} \times \text{Act} \times T_{\text{CCS}}\) between closed CCS expressions. The transitions \(p : \to q\) with \(p, q \in T_{\text{CCS}}\) and \(\alpha \in \text{Act}\) are derived from the rules of Table I. Formally, a transition \(p : \to q\) is part of the transition relation of CCS if there exists a well-founded, upwards branching tree (a proof of the transition) of which the nodes are labelled by transitions, such that

- the root is labelled by \(p : \to q\), and
- if \(\varphi\) is the label of a node \(n\) and \(K\) is the set of labels of the nodes directly above \(n\), then \(\varphi\) is a rule from Table I with closed CCS expressions substituted for the variables \(E, F, \ldots\).

### 2 CSP

CSP \([1, 5, 2, 3]\) is parametrised with a set \(\mathcal{A}\) of communications; \(\text{Act} := \mathcal{A} \cup \{\tau\}\) is the set of actions. Below, \(a, b\) range over \(\mathcal{A}\) and \(\alpha, \beta\) over \(\text{Act}\). The set \(\mathcal{E}\) of CSP terms is the smallest set including:
As in [5], I here leave out the guarded choice \( (x : B \rightarrow P(x)) \) and the constant \( \text{RUN} \) of [1], and the inverse image and sequential composition operator, with constant \( \text{SKIP} \), of [1, 2]. The semantics of CSP was originally given in quite a different way [1, 2], but [5] provided an operational semantics of CSP in the same style as the one of CCS, and showed its consistency with the original semantics. It is this operational semantics I will use here; it is given by the rules in Table 2. Let \( \mathcal{A}' := \mathcal{A} \).

$$
\begin{align*}
\text{DIV} & \xrightarrow{\tau} \text{DIV} \\
(a \rightarrow E) & \xrightarrow{a} E \\
E\boxdot F & \xrightarrow{a} E' \\
E \sqcap F & \xrightarrow{\tau} E \\
E \sqcap F & \xrightarrow{\tau} F \\
E + F & \xrightarrow{a} F' \\
E \parallel F & \xrightarrow{a} F' \\
E \parallel F & \xrightarrow{\tau} F'
\end{align*}
$$

Table 2: Structural operational semantics of CSP

References


