Encoding Distributed Areas into a Calculus Without Them

Feel free to skip the maths and just read the text

Tom Chothia

Abstract: Mathematics can provide a powerful tool to model computer processes and networks. It can also be used to prove designs correct and find faults in protocols. One of the best known systems for this kind of modelling is the π-calculus, see column three. This calculus models the inputs and output of processes running in parallel but it has no notion of separate locations. This makes it hard to model networks and Internet protocols that rely on certain processes being at different locations. The Local area calculus, (column one) adds a direct way to model areas, making it much easier to use when modelling distributed systems. The Local area calculus may be encoded into the π calculus, as we see in the second column. This means that local areas do not add any real expressiveness to the π Calculus and they are a convenient and fair short hand to use when modelling distributed systems.

The Local Area Calculus ...

$$
\begin{array}{rll}
\mathit{Process}\text{P, Q} ::= & 0 & \text{inactive process} \\
| & P \mid Q & \text{parallel composition} \\
| & \bar{a}\langle b\rangle & \text{output b on a} \\
| & a(b).P & \text{input b on a} \\
| & l\ [\ P\ ] & \text{local area/agent} \\
| & \nu\ a{:}\sigma.P & \text{fresh channel } a \text{ of type } \sigma
\end{array}
$$

The $La\pi$ calculus extends the π-calculus by adding a notion of a local areas. An area can be any environment used for computation, such as a user session, a computer or a network.The type σ of a fresh name controls how the name may be used to communicate. It restricts the name to work inside some areas but not others. For instance a name representing the standard output of a computer should be accessible to all sessions on that computer but not to sessions on other computers. Whereas a name representing an IP address should be usable everywhere.

$$
\begin{array}{lll}
\text{OUT} & \Gamma \vdash_\ell \bar{a}\langle b\rangle \xrightarrow{\bar{a}\langle b\rangle} 0 & \\
\text{IN} & \Gamma \vdash_\ell a(b).P \xrightarrow{a(b)} P & b \cap dom(\Gamma) = \emptyset \\
\text{PAR} & \dfrac{\Gamma \vdash_\ell P \xrightarrow{\alpha} P'}{\Gamma \vdash_\ell P \mid Q \xrightarrow{\alpha} P' \mid Q} & \\
\text{COMM} & \dfrac{\Gamma \vdash_\ell P \xrightarrow{\bar{a}\langle c\rangle} P' \quad \Gamma \vdash_\ell Q \xrightarrow{a(b)} Q'}{\Gamma \vdash_\ell P \mid Q \xrightarrow{\tau} P' \mid Q'\{c/b\}} & \\
\text{BIND} & \dfrac{\Gamma, a@m \vdash_\ell P \xrightarrow{\alpha} P'}{\Gamma \vdash_\ell \nu a{:}m.P \xrightarrow{\alpha} \nu a{:}m.P'} & a \notin fn(\alpha) \\
\text{AREA} & \dfrac{\Gamma \vdash_\ell P \xrightarrow{\alpha} P'}{\Gamma \vdash_m \ell[P] \xrightarrow{\alpha} \ell[P']} & \text{if } \alpha \text{ is } \bar{a}\langle b\rangle \text{ or } a(b) \text{ then } m \leq \Gamma(a)
\end{array}
$$

Can be Encoded into ...

$$
\begin{array}{rcll}
[\![\Gamma \vdash_\ell m{:}[Q]]\!]_\Delta & = & \nu\ e.[\![\Gamma \vdash_m Q]\!]_{\Delta \oplus \{l@e\}} & \\
[\![\Gamma \vdash_\ell a(b).P]\!]_\Delta & = & \mu\ X.e(\alpha, b).([\alpha = a].[\![\Gamma \oplus \{\vec{b}{:}\sigma\} \vdash P]\!]_\Delta & e = \Delta(m)\ \& \\
 & & \quad \mid [\alpha \neq a].(\overline{e}\langle \alpha, b\rangle \mid X)) & \Gamma(a) = \sigma@m \\
[\![\Gamma \vdash_\ell \overline{a}\langle b\rangle]\!]_\Delta & = & \overline{e}\langle a, b\rangle & e = \Delta(\Gamma(a)) \\
[\![\Gamma \vdash_\ell \nu\ a{:}\sigma.P]\!]_\Delta & = & \nu\ a.[\![\Gamma \oplus \{a{:}\sigma\} \vdash P]\!]_\Delta & \\
[\![\Gamma \vdash P \mid Q]\!]_\Delta & = & [\![\Gamma \vdash P]\!]_\Delta \mid [\![\Gamma \vdash Q]\!]_\Delta & \\
[\![\Gamma \vdash_\ell 0]\!]_\Delta & = & 0 &
\end{array}
$$

This encoding removes all mention of different areas from a $La\pi$ term, translating it into a π-term. An encoded process behaves in the same way as the process it encodes. This tells us that although local areas are very useful when modelling a real life system they do not add any real expressiveness to the π calculus.

The π-calculus

$$
\begin{array}{rll}
\mathit{Process}\text{P, Q} ::= & 0 & \text{inactive process} \\
| & P \mid Q & P \text{ and } Q \text{ running in parallel} \\
| & \bar{a}\langle \vec{b}\rangle & \text{output tuple} \\
| & a(\vec{b}).P & \text{input} \\
| & [a = b].P & \text{matching} \\
| & [a \neq b].P & \text{mismatching} \\
| & \nu\ a.P & \text{fresh channel } a
\end{array}
$$

A name with a bar above it is an output and a name with no bar is an input. If two processes running in parallel both try to communicate on the same name the data being outputted will replace the data in the input process. Matching and mismatching allow us to test names for equality. This small and simple calculus is incredibly powerful. It can express complicated systems with great precision in such a way that they can easily be analysed.

$$
\begin{array}{lll}
\text{OUT} & \overline{a}\langle \vec{b}\rangle \xrightarrow{\overline{a}\langle \vec{b}\rangle} 0 & \\
\text{IN} & a(\vec{b}).P \xrightarrow{a(\vec{b})} P & \\
\text{PAR} & \dfrac{P \xrightarrow{\alpha} P'}{\Gamma \vdash_\ell P \mid Q \xrightarrow{\alpha} P' \mid Q} & \\
\text{COMM} & \dfrac{P \xrightarrow{\bar{a}\langle \vec{c}\rangle} P' \quad Q \xrightarrow{a(\vec{b})} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'\{\vec{c}/\vec{b}\}} & \text{if } \alpha = a(\vec{b}) \text{ then then } \vec{b} \notin fn(Q) \\
\text{BIND} & \dfrac{P \xrightarrow{\alpha} P'}{\nu\ a.P \xrightarrow{\alpha} \nu\ a.P'} & a \notin fn(\alpha)
\end{array}
$$

Division of informatics