α-Equivalence is very natural, as terms that are α-equivalent “mean the same.” In fact, it is possible to give a syntax for lambda terms which does not distinguish terms that can be α-converted to each other. The best known replaces variables by their De Bruijn index.

When we write \( \lambda x.M \), we explicitly state that \( x \) is the parameter of the function, so that we can use \( x \) in \( M \) to refer to this parameter. In the de Bruijn index, however, parameters have no name and reference to them in the function body is denoted by a number denoting the levels of abstraction between them. For example, consider the example of \( \lambda x. \lambda y. yx \): the outer abstraction is on binds the variable \( x \); the inner abstraction binds the variable is \( y \); the sub-term \( yx \) lies in the scope of the inner abstraction: there is no abstraction between \( y \) and its abstract \( \lambda y \), but one abstract between \( x \) and its abstract \( \lambda x \). Thus we write \( 0 \ 1 \) for \( yx \), and \( \lambda. \lambda.0\) for the entire term.

**Definition syn.1.** De Bruijn terms are inductively defined as follows:

1. \( n \), where \( n \) is any natural number.
2. \( PQ \), where \( P \) and \( Q \) are both De Bruijn terms.
3. \( \lambda. N \), where \( N \) is a De Bruijn term.

A formalized translation from ordinary lambda terms to De Bruijn indexed terms is as follows:

**Definition syn.2.**

\[
\begin{align*}
F_{\Gamma}(x) &= \Gamma(x) \\
F_{\Gamma}(PQ) &= F_{\Gamma}(P)F_{\Gamma}(Q) \\
F_{\Gamma}(\lambda x. N) &= \lambda x.F_{\Gamma}(x)N
\end{align*}
\]

where \( \Gamma \) is a list of variables indexed from zero, and \( \Gamma(x) \) denotes the position of the variable \( x \) in \( \Gamma \). For example, if \( \Gamma \) is \( x, y, z \), then \( \Gamma(x) \) is 0 and \( \Gamma(z) \) is 2.

\( x, \Gamma \) denotes the list resulted from pushing \( x \) to the head of \( \Gamma \); for instance, continuing the \( \Gamma \) in last example, \( w, \Gamma \) is \( w, x, y, z \).

Recovering a standard lambda term from a de Bruijn term is done as follows:

**Definition syn.3.**

\[
\begin{align*}
G_{\Gamma}(n) &= \Gamma[n] \\
G_{\Gamma}(PQ) &= G_{\Gamma}(P)G_{\Gamma}(Q) \\
G_{\Gamma}(\lambda. N) &= \lambda x.G_{\Gamma}(x)N
\end{align*}
\]

where \( \Gamma \) is again a list of variables indexed from zero, and \( \Gamma[n] \) denotes the variable in position \( n \). For example, if \( \Gamma \) is \( x, y, z \), then \( \Gamma[1] \) is \( y \).

The variable \( x \) in last equation is chosen to be any variable that not in \( \Gamma \).
Here we give some results without proving them:

**Proposition syn.4.** If $M \xrightarrow{\alpha} M'$, and $\Gamma$ is any list containing $\text{FV}(M)$, then $F_{\Gamma}(M) \equiv F_{\Gamma}(M')$.

Photo Credits

Bibliography