Introduction

Historically the lambda calculus and intuitionistic logic were developed separately. Haskell Curry and William Howard independently discovered a close similarity: types in a typed lambda calculus correspond to formulas in intuitionistic logic in such a way that a derivation of a formula corresponds directly to a typed lambda term with that formula as its type. Moreover, beta reduction in the typed lambda calculus corresponds to certain transformations of derivations.

For instance, a derivation of $\varphi \to \psi$ corresponds to a term $\lambda x^{\varphi} \cdot N^{\psi}$, which has the function type $\varphi \to \psi$. The inference rules of natural deduction correspond to typing rules in the typed lambda calculus, e.g.,

\[
\begin{array}{c}
\vdash [\varphi]^x \\
\vdots \\
\psi \\
\hline \\
\varphi \to \psi \to \text{Intro}
\end{array}
\]

\[
\begin{array}{c}
x : \varphi \Rightarrow N : \psi \\
\Rightarrow \lambda x^{\varphi} \cdot N^{\psi} : \varphi \to \psi
\end{array}
\]

where the rule on the right means that if $x$ is of type $\varphi$ and $N$ is of type $\psi$, then $\lambda x^{\varphi} \cdot N^{\psi}$ is of type $\varphi \to \psi$.

The $\to \text{Elim}$ rule corresponds to the typing rule for composition terms, i.e.,

\[
\begin{array}{c}
\varphi \to \psi \to \text{Elim} \\
\Rightarrow P : \varphi \to \psi \\
\Rightarrow Q : \varphi \\
\Rightarrow P^{\varphi \to \psi}Q^{\varphi} : \psi
\end{array}
\]

If a $\to \text{Intro}$ rule is followed immediately by a $\to \text{Elim}$ rule, the derivation can be simplified:

\[
\begin{array}{c}
\vdash [\varphi]^x \\
\vdots \\
\phi \\
\vdots \\
\psi \\
\hline x \varphi \to \psi \to \text{Intro} \\
\phi \varphi \to \psi \to \text{Elim}
\end{array}
\]

which corresponds to the beta reduction of lambda terms

\[(\lambda x^{\varphi} \cdot P^{\psi})Q \to \varphi \to P^{Q/x} \cdot \varphi_1 \cdot P^{Q/x}\]

Similar correspondences hold between the rules for $\land$ and “product” types, and between the rules for $\lor$ and “sum” types.

This correspondence between terms in the simply typed lambda calculus and natural deduction derivations is called the “Curry-Howard”, or “propositions as types” correspondence. In addition to formulas (propositions) corresponding to types, and proofs to terms, we can summarize the correspondences as follows:
The Curry-Howard correspondence is one of the cornerstones of automated proof assistants and type checkers for programs, since checking a proof witnessing a proposition (as we did above) amounts to checking if a program (term) has the declared type.

Photo Credits

Bibliography