Summary of implementation methods

1. ISWIM $\longrightarrow$ λ-calculus $\longrightarrow$ SECD machine;
2. λ-calculus $\longrightarrow$ de Bruijn notation;
3. λ-calculus $\longrightarrow$ combinators $\longrightarrow$ graph reduction.

Combinators

Definition

The **CL terms** are constructed recursively from a given set of variables V. They may take one of the following forms:

- x variable;
- **K**, called the K combinator;
- **S**, called the S combinator;
- (PQ) application, for CL terms P and Q.

Axioms for reduction

$$\mathbf{K}PQ \rightarrow_w P$$
$$\mathbf{S}PQR \rightarrow_w (PR)(QR)$$

plus closure under application

From combinators to λ-terms

The translation $(\_)_\lambda$: set of CL terms $\longrightarrow$ set of λ-terms is defined inductively by

$$\begin{aligned}
(x)_\lambda &\equiv x \\
(\mathbf{K})_\lambda &\equiv \lambda xy.x \\
(\mathbf{S})_\lambda &\equiv \lambda xyz.xz(yz) \\
(PQ)_\lambda &\equiv (P)_\lambda(Q)_\lambda
\end{aligned}$$

From λ-terms to combinators

The translation $(\_)_{cl}$: set of λ-terms $\longrightarrow$ set of CL terms is defined inductively by

$$\begin{aligned}
(x)_{cl} &\equiv x \\
(MN)_{cl} &\equiv (M)_{cl}(N)_{cl} \\
(\lambda x.M)_{cl} &\equiv \lambda^* x.(M)_{cl}
\end{aligned}$$

Proposition

For all CL terms P and Q, we have

1. $\mathsf{FV}(\lambda^* x.P) = \mathsf{FV}(P) - \{x\}$.
2. $(\lambda^* x.P)x \twoheadrightarrow_w P$.
3. $(\lambda^* x.P)Q \twoheadrightarrow_w P[Q/x]$.
4. $\lambda^* x.P \equiv \lambda^* y.P[y/x]$

Abstraction on combinators

Definition

The operation $\lambda^* x$ on combinators, where x is a variable, is defined recursively as follows:

$\lambda^* x.x$ is **I**;

$\lambda^* x.P$ is **K**P if $x \notin \mathsf{FV}(P)$;

$\lambda^* x.PQ$ is **S**$(\lambda^* x.P)(\lambda^* x.Q)$.

Notation $\lambda^* x_1 \ldots x_n.P \equiv \lambda^* x_1.(\lambda^* x_2. \ldots \lambda^* x_n.P \ldots))$.

Facts

Assume an extra rule for combinators, called the **principle of extensionality**. Then the following results hold:

1. $\mathsf{FV}(M) = \mathsf{FV}(M_{cl})$;
2. $\mathsf{FV}(P) = \mathsf{FV}(P_\lambda)$;
3. $M = N$ if and only if $M_{cl} = N_{cl}$;
4. $P = Q$ if and only if $P_\lambda = Q_\lambda$;
5. $(M_{cl})_\lambda = M$ and $(P_\lambda)_{cl} = P$.