Combinatory Logic

Olivier Dragon

November 4, 2004

Purpose

- **Higher-order logic: functions that can take functions as arguments and can return functions.**

(Very) Brief History

- **M. Schönfinkel (1924)**
- **H.B. Curry and A. Church (1930)**

Schönfinkel: S, K
Curry: $\mathsf{B}, \mathsf{C}, \mathsf{K}, \mathsf{W}$

Basic Definitions

- **Variables**
- **Constants**
- **Basic Combinators:** S, K
- **CL-Term**
  1. **All variables and constants including** K **and** S **are CL-terms**
  2. **If** X **and** Y **are CL-Terms, then so is** (XY)

Basic Definitions (2)

- **Occurance**
  1. X **occurs in** X
  2. **if** X **occurs in** U **or** V**,** X **occurs in** (UV)
- **Substitution**
  1. $[U/x]x \equiv x$
  2. $[U/x]a \equiv a, \mathrm{if a} \neq \mathrm{x}$
  3. $[U/x](VW) \equiv ([U/x]V[U/x]W)$

Typical Combinator Examples

K	**which forms constant functions**	$(\mathsf{K}a)(x) = a$
S	**stronger function composition operator**	$(\mathsf{S}(f, g))(x) = f(x, g(x))$
C	**commute the arguments of a function**	$(\mathsf{C}f)(x, y) = f(y, x)$
B	**compose two functions**	$(\mathsf{B}(f, g))(x) = f(g(x))$
I	**identity**	$\mathsf{I}f = f$
W	**diagonalising operator**	$(\mathsf{W}f)(x) = f(x, x)$

Weak Reduction

- $\mathsf{K}XY$ **weakly contracts to** X
- $\mathsf{S}XYZ$ **weakly contracts to** $XZ(YZ)$
- $\mathsf{B}XYZ$ **weakly reduces to** $X(YZ)$

$$
\begin{array}{lll}
\mathsf{B} & \equiv & \mathsf{S}(\mathsf{KS})\mathsf{K} \\
\mathsf{B}XYZ & \equiv & \mathsf{S}(\mathsf{KS})\mathsf{K}XYZ \\
 & \triangleright_{1\mathrm{W}} & \mathsf{KS}X(\mathsf{K}X)YZ \\
 & \triangleright_{1\mathrm{W}} & \mathsf{S}(\mathsf{K}X)YZ \\
 & \triangleright_{1\mathrm{W}} & \mathsf{K}XZ(YZ) \\
 & \triangleright_{1\mathrm{W}} & X(YZ) \\
\mathsf{B}XYZ & \triangleright_{\mathrm{W}} & X(YZ)
\end{array}
$$

Factorial Function Example

```
factorial(x)
    if (x = 0)
        return 1
    else
        return x * factorial(x-1)
```

Factorial in Combinatory Logic

Y	recursive operator	$\mathsf{Y}f = f(\mathsf{Y}f)$
D	conditional operator	$(\mathsf{D}pfg)x = \text{if } p(x) \text{ then } f(x) \text{ else } g(x)$
P	predecessor operator for integers	$\mathsf{P}x = x - 1$

$$\text{factorial} = \mathsf{Y}((\mathsf{B}(\mathsf{D}((=)0)(\mathsf{K}1))(\mathsf{B}(\mathsf{S}(*))(\mathsf{CBP})))$$

References

[1] J. Roger Hindley and Jonathan P. Seldin. *Introduction to Combinators and λ-Calculus*. Cambridge University Press, 1986.

[2] Fritz Ruehr. *The Evolution of a Haskell Programmer*. World Wide Web, `http://www.willamette.edu/~fruehr/haskell/evolution.html`, August 2001. Last Viewed November 2004.

[3] Wikipedia. *Combinatory Logic*. From Wikipedia, The Free Encyclopedia, `http://en.wikipedia.org/wiki/Combinatory_logic`, October 2004. Last Viewed November 2004.