Lambda-Definable Functions

β-Conversion $M =_\beta N$

Informally: $M =_\beta N$ holds if N can be obtained from M by performing zero or more steps of α-equivalence, β-reduction, or β-*expansion* (= inverse of a reduction).

E.g. $u\,((\lambda x\,y.\,v\,x)y) =_\beta (\lambda x.\,u\,x)(\lambda x.\,v\,y)$

because $(\lambda x.\,u\,x)(\lambda x.\,v\,y) \to u(\lambda x.\,v\,y)$

and so we have

$$
\begin{array}{rcll}
u\,((\lambda x\,y.\,v\,x)y) & =_\alpha & u\,((\lambda x\,y'.\,v\,x)y) & \\
& \to & u(\lambda y'.\,v\,y) & \text{reduction} \\
& =_\alpha & u(\lambda x.\,v\,y) & \\
& \leftarrow & (\lambda x.\,u\,x)(\lambda x.\,v\,y) & \text{expansion}
\end{array}
$$

β-Conversion $M =_\beta N$

is the binary relation inductively generated by the rules:

$$\frac{M =_\alpha M'}{M =_\beta M'} \qquad \frac{M \to M'}{M =_\beta M'} \qquad \frac{M =_\beta M'}{M' =_\beta M}$$

$$\frac{M =_\beta M' \qquad M' =_\beta M''}{M =_\beta M''} \qquad \frac{M =_\beta M'}{\lambda x.M =_\beta \lambda x.M'}$$

$$\frac{M =_\beta M' \qquad N =_\beta N'}{M\,N =_\beta M'\,N'}$$

Church-Rosser Theorem

Theorem. $\twoheadrightarrow$ is confluent, that is, if $M_1 \twoheadleftarrow M \twoheadrightarrow M_2$, then there exists M' such that $M_1 \twoheadrightarrow M' \twoheadleftarrow M_2$.

[Proof omitted.]

Church-Rosser Theorem

Theorem. $\twoheadrightarrow$ is confluent, that is, if $M_1 \twoheadleftarrow M \twoheadrightarrow M_2$, then there exists M' such that $M_1 \twoheadrightarrow M' \twoheadleftarrow M_2$.

Corollary. Two show that two terms are β-convertible, it suffices to show that they both reduce to the same term. More precisely: $M_1 =_\beta M_2$ iff $\exists M\,(M_1 \twoheadrightarrow M \twoheadleftarrow M_2)$.

Church-Rosser Theorem

Theorem. $\twoheadrightarrow$ is confluent, that is, if $M_1 \twoheadleftarrow M \twoheadrightarrow M_2$, then there exists M' such that $M_1 \twoheadrightarrow M' \twoheadleftarrow M_2$.

Corollary. $M_1 =_\beta M_2$ iff $\exists M\,(M_1 \twoheadrightarrow M \twoheadleftarrow M_2)$.

Proof. $=_\beta$ satisfies the rules generating $\twoheadrightarrow$; so $M \twoheadrightarrow M'$ implies $M =_\beta M'$. Thus if $M_1 \twoheadrightarrow M \twoheadleftarrow M_2$, then $M_1 =_\beta M =_\beta M_2$ and so $M_1 =_\beta M_2$.

Conversely, the relation $\{(M_1, M_2) \mid \exists M\,(M_1 \twoheadrightarrow M \twoheadleftarrow M_2)\}$ satisfies the rules generating $=_\beta$: the only difficult case is closure of the relation under transitivity and for this we use the Church-Rosser theorem: $M_1 \longrightarrow\!\!\!\!\!\rightarrow M \twoheadleftarrow\!\!\!\!\!\longleftarrow M_2 \longrightarrow\!\!\!\!\!\rightarrow M' \twoheadleftarrow\!\!\!\!\!\longleftarrow M_3$

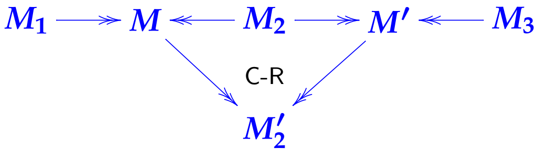
Church-Rosser Theorem

Theorem. $\twoheadrightarrow$ is confluent, that is, if $M_1 \twoheadleftarrow M \twoheadrightarrow M_2$, then there exists M' such that $M_1 \twoheadrightarrow M' \twoheadleftarrow M_2$.

Corollary. $M_1 =_\beta M_2$ iff $\exists M\,(M_1 \twoheadrightarrow M \twoheadleftarrow M_2)$.

Proof. $=_\beta$ satisfies the rules generating $\twoheadrightarrow$; so $M \twoheadrightarrow M'$ implies $M =_\beta M'$. Thus if $M_1 \twoheadrightarrow M \twoheadleftarrow M_2$, then $M_1 =_\beta M =_\beta M_2$ and so $M_1 =_\beta M_2$.

Conversely, the relation $\{(M_1, M_2) \mid \exists M\,(M_1 \twoheadrightarrow M \twoheadleftarrow M_2)\}$ satisfies the rules generating $=_\beta$: the only difficult case is closure of the relation under transitivity and for this we use the Church-Rosser theorem:



Church-Rosser Theorem

Theorem. $\twoheadrightarrow$ is confluent, that is, if $M_1 \twoheadleftarrow M \twoheadrightarrow M_2$, then there exists M' such that $M_1 \twoheadrightarrow M' \twoheadleftarrow M_2$.

Corollary. $M_1 =_\beta M_2$ iff $\exists M\,(M_1 \twoheadrightarrow M \twoheadleftarrow M_2)$.

Proof. $=_\beta$ satisfies the rules generating $\twoheadrightarrow$; so $M \twoheadrightarrow M'$ implies $M =_\beta M'$. Thus if $M_1 \twoheadrightarrow M \twoheadleftarrow M_2$, then $M_1 =_\beta M =_\beta M_2$ and so $M_1 =_\beta M_2$.

Conversely, the relation $\{(M_1, M_2) \mid \exists M\,(M_1 \twoheadrightarrow M \twoheadleftarrow M_2)\}$ satisfies the rules generating $=_\beta$: the only difficult case is closure of the relation under transitivity and for this we use the Church-Rosser theorem. Hence $M_1 =_\beta M_2$ implies $\exists M\,(M_1 \twoheadrightarrow M' \twoheadleftarrow M_2)$.

β-Normal Forms

Definition. A λ-term N is in β-normal form (nf) if it contains no β-redexes (no sub-terms of the form $(\lambda x.M)M'$). M has β-nf N if $M =_\beta N$ with N a β-nf.

β-Normal Forms

Definition. A λ-term N is in β-normal form (nf) if it contains no β-redexes (no sub-terms of the form $(\lambda x.M)M'$). M has β-nf N if $M =_\beta N$ with N a β-nf.

Note that if N is a β-nf and $N \twoheadrightarrow N'$, then it must be that $N =_\alpha N'$ (why?).

Hence if $N_1 =_\beta N_2$ with N_1 and N_2 both β-nfs, then $N_1 =_\alpha N_2$. (For if $N_1 =_\beta N_2$, then $N_1 \twoheadleftarrow M \twoheadrightarrow N_2$ for some M; hence by Church-Rosser, $N_1 \twoheadrightarrow M' \twoheadleftarrow N_2$ for some M', so $N_1 =_\alpha M' =_\alpha N_2$.)

So the β-nf of M is unique up to α-equivalence if it exists.

Non-termination

Some λ terms have no β-nf.

E.g. $\Omega \triangleq (\lambda x.x\,x)(\lambda x.x\,x)$ satisfies

- $\Omega \to (x\,x)[(\lambda x.x\,x)/x] = \Omega$,
- $\Omega \twoheadrightarrow M$ implies $\Omega =_\alpha M$.

So there is no β-nf N such that $\Omega =_\beta N$.

Non-termination

Some λ terms have no β-nf.

E.g. $\Omega \triangleq (\lambda x.x\,x)(\lambda x.x\,x)$ satisfies

- $\Omega \to (x\,x)[(\lambda x.x\,x)/x] = \Omega$,
- $\Omega \twoheadrightarrow M$ implies $\Omega =_\alpha M$.

So there is no β-nf N such that $\Omega =_\beta N$.

A term can possess both a β-nf and infinite chains of reduction from it.

E.g. $(\lambda x.y)\Omega \to y$, but also $(\lambda x.y)\Omega \to (\lambda x.y)\Omega \to \cdots$.

Non-termination

Normal-order reduction is a deterministic strategy for reducing λ-terms: reduce the "left-most, outer-most" redex first.

- left-most: reduce M before N in $M\,N$, and then
- outer-most: reduce $(\lambda x.M)N$ rather than either of M or N.

(cf. call-by-name evaluation).

Fact: normal-order reduction of M always reaches the β-nf of M if it possesses one.

Encoding data in λ-calculus

Computation in λ-calculus is given by β-reduction. To relate this to register/Turing-machine computation, or to partial recursive functions, we first have to see how to encode numbers, pairs, lists, ... as λ-terms.

We will use the original encoding of numbers due to Church...

Church's numerals

$$\begin{aligned}\underline{0} &\triangleq \lambda f\,x.x\\ \underline{1} &\triangleq \lambda f\,x.f\,x\\ \underline{2} &\triangleq \lambda f\,x.f(f\,x)\\ &\vdots\\ \underline{n} &\triangleq \lambda f\,x.\underbrace{f(\cdots(f}_{n\text{ times}}\,x)\cdots)\end{aligned}$$

Notation: $\begin{cases} M^0 N \triangleq N \\ M^1 N \triangleq M\,N \\ M^{n+1} N \triangleq M(M^n N) \end{cases}$

so we can write $\underline{n}$ as $\lambda f\,x.f^n x$ and we have $\boxed{\underline{n}\,M\,N =_\beta M^n\,N}$.

λ-Definable functions

Definition. $f \in \mathbb{N}^n \rightharpoonup \mathbb{N}$ is λ-definable if there is a closed λ-term F that represents it: for all $(x_1, \ldots, x_n) \in \mathbb{N}^n$ and $y \in \mathbb{N}$

- if $f(x_1, \ldots, x_n) = y$, then $F\, \underline{x_1} \cdots \underline{x_n} =_\beta \underline{y}$
- if $f(x_1, \ldots, x_n)\uparrow$, then $F\, \underline{x_1} \cdots \underline{x_n}$ has no β-nf.

For example, addition is λ-definable because it is represented by $P \triangleq \lambda x_1\, x_2 . \lambda f\, x.\, x_1\, f(x_2\, f\, x)$:

$$\begin{aligned} P\,\underline{m}\,\underline{n} &=_\beta \lambda f\, x.\, \underline{m}\, f(\underline{n}\, f\, x) \\ &=_\beta \lambda f\, x.\, \underline{m}\, f(f^n x) \\ &=_\beta \lambda f\, x.\, f^m(f^n x) \\ &= \lambda f\, x. f^{m+n} x \\ &= \underline{m+n} \end{aligned}$$

Computable = λ-definable

Theorem. A partial function is computable if and only if it is λ-definable.

We already know that

$$\begin{array}{ll} & \text{Register Machine computable} \\ = & \text{Turing computable} \\ = & \text{partial recursive.} \end{array}$$

Using this, we break the theorem into two parts:

- every partial recursive function is λ-definable
- λ-definable functions are RM computable

λ-Definable functions

Definition. $f \in \mathbb{N}^n \rightharpoonup \mathbb{N}$ is λ-definable if there is a closed λ-term F that represents it: for all $(x_1, \ldots, x_n) \in \mathbb{N}^n$ and $y \in \mathbb{N}$

- if $f(x_1, \ldots, x_n) = y$, then $F\, \underline{x_1} \cdots \underline{x_n} =_\beta \underline{y}$
- if $f(x_1, \ldots, x_n)\uparrow$, then $F\, \underline{x_1} \cdots \underline{x_n}$ has no β-nf.

This condition can make it quite tricky to find a λ-term representing a non-total function.

For now, we concentrate on total functions. First, let us see why the elements of **PRIM** (primitive recursive functions) are λ-definable.

Basic functions

- Projection functions, $\mathtt{proj}_i^n \in \mathbb{N}^n \to \mathbb{N}$:

$$\mathtt{proj}_i^n(x_1, \ldots, x_n) \triangleq x_i$$

- Constant functions with value $\mathbf{0}$, $\mathtt{zero}^n \in \mathbb{N}^n \to \mathbb{N}$:

$$\mathtt{zero}^n(x_1, \ldots, x_n) \triangleq \mathbf{0}$$

- Successor function, $\mathtt{succ} \in \mathbb{N} \to \mathbb{N}$:

$$\mathtt{succ}(x) \triangleq x + 1$$

Basic functions are representable

- $\mathtt{proj}_i^n \in \mathbb{N}^n \to \mathbb{N}$ is represented by $\lambda x_1 \ldots x_n . x_i$
- $\mathtt{zero}^n \in \mathbb{N}^n \to \mathbb{N}$ is represented by $\lambda x_1 \ldots x_n . \underline{0}$
- $\mathtt{succ} \in \mathbb{N} \to \mathbb{N}$ is represented by

$$\mathbf{Succ} \triangleq \lambda x_1\, f\, x . f\, (x_1\, f\, x)$$

since

$$\begin{aligned} \mathbf{Succ}\, \underline{n} &=_\beta \lambda f\, x .\, f\, (\underline{n}\, f\, x) \\ &=_\beta \lambda f\, x .\, f\, (f^n\, x) \\ &= \lambda f\, x .\, f^{n+1}\, x \\ &= \underline{n+1} \end{aligned}$$

Representing composition

If total function $f \in \mathbb{N}^n \to \mathbb{N}$ is represented by F and total functions $g_1, \ldots, g_n \in \mathbb{N}^m \to \mathbb{N}$ are represented by $G_1, \ldots, G_n$, then their composition $f \circ (g_1, \ldots, g_n) \in \mathbb{N}^m \to \mathbb{N}$ is represented simply by

$$\lambda x_1 \ldots x_m . F\,(G_1\, x_1 \ldots x_m) \ldots (G_n\, x_1 \ldots x_m)$$

because

$$\begin{array}{rl} & F\,(G_1\, \underline{a_1} \ldots \underline{a_m}) \ldots (G_n\, \underline{a_1} \ldots \underline{a_m}) \\ =_\beta & F\, \underline{g_1(a_1, \ldots, a_m)} \ldots \underline{g_n(a_1, \ldots, a_m)} \\ =_\beta & \underline{f(g_1(a_1, \ldots, a_m), \ldots, g_n(a_1, \ldots, a_m))} \\ = & \underline{f \circ (g_1, \ldots, g_n)(a_1, \ldots, a_m)} \end{array}$$

.

Representing composition

If total function $f \in \mathbb{N}^n \to \mathbb{N}$ is represented by F and total functions $g_1, \ldots, g_n \in \mathbb{N}^m \to \mathbb{N}$ are represented by $G_1, \ldots, G_n$, then their composition $f \circ (g_1, \ldots, g_n) \in \mathbb{N}^m \to \mathbb{N}$ is represented simply by

$$\lambda x_1 \ldots x_m . F\,(G_1\, x_1 \ldots x_m) \ldots (G_n\, x_1 \ldots x_m)$$

This does not necessarily work for partial functions. E.g. totally undefined function $u \in \mathbb{N} \rightharpoonup \mathbb{N}$ is represented by $U \triangleq \lambda x_1 . \Omega$ (why?) and $\mathtt{zero}^1 \in \mathbb{N} \to \mathbb{N}$ is represented by $Z \triangleq \lambda x_1 . \underline{0}$; but $\mathtt{zero}^1 \circ u$ is not represented by $\lambda x_1 . Z(U\, x_1)$, because $(\mathtt{zero}^1 \circ u)(n)\uparrow$ whereas $(\lambda x_1 . Z(U\, x_1))\, \underline{n} =_\beta Z\, \Omega =_\beta \underline{0}$. (What is $\mathtt{zero}^1 \circ u$ represented by?)