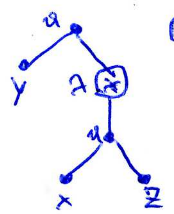
(λ) Summary of last time: λ-terms (set Λ): x, (MN), $(\lambda u.M)$

(λ) Syntactical identity: $(xz) \equiv (xz)$, $(xz) \not\equiv (xy)$

(Mλ) subterms, multi set Sub, tree of subterms:



$(y(\lambda x.(xz)))$

(Mλ)? Notation:

- $MN \overset{\text{stands for}}{=} (MN)$
- $MNL = ((MN)L)$
- $\lambda xy.M = (\lambda x.(\lambda y.M))$
- $\lambda x.MN = \lambda x.(MN)$

(Mλ) free, bound, binding variables, sets FV, B, B: $\underset{\text{free}}{y}(\lambda \underset{\text{binding}}{x}.\underset{\text{bound}}{x}\underset{\text{free}}{z})$

(Mλ) closed λ terms

(Mλ) renaming: $x(\lambda x.xy)^{x \to y} \equiv y(\lambda x.xy)$

(λ) alpha conversion, $=_\alpha$: $\lambda x.M =_\alpha \lambda y.M$ if status of free and bound variables doesn't change

Parts of λ-calculus: (λ)

Parts of meta-theory (to talk __about__ λ-calculus) (Mλ)

Substitution: the basis of actual computation (β-reduction)

Definition: $\boxed{\text{Substitution}}$ (1a) $x[x := N] \equiv N$

(1b) $\left(y[x := N] \equiv y \text{ if } x \not\equiv y \right)$ ← error in the book (I think so)

correct: If $V \in \Lambda$ is a variable (i.e. $V \in V$) then

$$V[x := N] = V \text{ if } x \not\equiv V$$

(2) $(PQ)[x := N] \equiv (P[x := N])(Q[x := N])$

(3) $(\lambda y.P)[x := N] \equiv \lambda z.(P^{y \to z}[x := N])$ if $\lambda z.P^{y \to z}$ is an α-variant of $\lambda y.P$ such that $z \notin FV(N)$ to avoid "capturing" of a free variable in N

actually only $=_\alpha$, but will identify $=_\alpha$ and $\equiv$ later

Note: • $P[x := N]$ is not a λ-term but a meta term, "something that stands for a λ-term"

• If $y \notin FV(N)$ or $x \notin FV(P)$ then (3) simplifies to $(\lambda y.P)[x := N] = \lambda y.(P[x := N])$

no risk of capturing, $P^{y \to y} = P$ — nothing to substitute

• Renaming is special case of substitution: $M^{x \to u} =_\alpha M[x := u]$ if conditions of renaming are met

Example: $(\lambda y.yx)[x := xy] \equiv ((\lambda z.zx)[x := xy]) \equiv \lambda z((zx)[x := xy])$

again, actually only $=_\alpha$ ← brackets necessary

$\equiv \lambda z.((z[x := xy])(x[x := xy])) \equiv \lambda z(xy)$

See also Examples 1.6.4.

Notation: Substitution is left-associative: $M[x := N][y := L] \overset{\text{stands for}}{=} (M[x := N])[y := L]$

Lemma (sequential substitution):

Let $x \not\equiv y$ (again, error in the book, redundant) and $x \notin FV(L)$. Then

$M[x := N][y := L] \equiv M[y := L][x := N[y := L]]$

Proof sketch on pages 13 and 14.

α-equivalence revisited

α-equivalent terms have identical trees apart from names of binding variables.

Show exactly the same pattern of free, bound and binding variables and thus

exactly the same behavior under β-reduction (defined soon)

Lemma: Let $M_1 =_\alpha M_2$, $N_1 =_\alpha N_2$. Then,

(1) $M_1 N_1 =_\alpha M_2 N_2$

(2) $\lambda x M_1 =_\alpha \lambda x.M_2$

(3) $M_1[x := N_1] =_\alpha M_2[x := N_2]$

Notation: • From now on consider α-equivalent terms to be syntactically identical: "$=_\alpha$" = "$\equiv$"

i.e. we write $\lambda x.x \equiv \lambda y.y$ from now on

• Barendregt convention: We choose the names of binding variables in a λ-term such that they are all different and such that they are all different from all free variables in the λ-term

Write $(\lambda xy.xz)(\lambda xz.z)$ as $(\lambda xy.xz)(\lambda u.v.v)$

Beta reduction: the actual computation

Definition: $\boxed{\text{One step } \beta\text{-reduction}}$, $\boxed{\to_\beta}$

(1) (Basis) $(\lambda x.M)N \to_\beta M[x:=N]$

(2) (Compatibility) If $M \to_\beta N$, then $ML \to_\beta NL$, $LM \to_\beta LN$ and $\lambda x.M \to_\beta \lambda x.N$

β-reduction within subterms

$\boxed{\text{Redex}}$: Subterm of the form $(\lambda x.M)N$ ("reducible expression")

$(\lambda x.M)N \to_\beta \underbrace{M[x:=N]}_{\boxed{\text{contractum}}}$ of the redex $(\lambda x.M)N$

i.e. $\ldots((\lambda x.M)N)\ldots \to_\beta \ldots(M[x:=N])\ldots$

(See Examples 1.8.2.)

Definition: $\boxed{\beta\text{-reduction}}$, $\boxed{\twoheadrightarrow_\beta}$, $\boxed{\beta\text{-conversion}}$, $\boxed{=_\beta}$

$\boxed{M \twoheadrightarrow_\beta N}$ if $\exists M_0, \ldots, M_n : M_0 \equiv M, M_n = N, M_i \to_\beta M_{i+1}$ for all $i \in \{0, \ldots, n-1\}$

$\boxed{M =_\beta N}$ if $\exists M_0, \ldots, M_n : M_0 = M, M_n = N, M_i \to_\beta M_{i+1}$ or $M_{i+1} \to_\beta M_i$ for all $i \in \{0, \ldots, n-1\}$

Example: $(\lambda x.(\lambda y.yx)z)v =_\beta (\lambda y.yv)z =_\beta (\lambda x.zx)v =_\beta zv$

Lemma: (1) $\twoheadrightarrow_\beta$ extends $\to_\beta$, $=_\beta$ extends $\twoheadrightarrow_\beta$ in both directions:

$M \to_\beta N \Rightarrow M \twoheadrightarrow_\beta N$, $M \twoheadrightarrow_\beta N$ or $N \twoheadrightarrow_\beta M \Rightarrow M =_\beta N$

(2) $\twoheadrightarrow_\beta$ is reflexive and transitive

$=_\beta$ is reflexive, transitive, symmetric, i.e. an equivalence relation

Examples: 1) $(\lambda x.(\lambda y.yx)z)v \to_\beta (\lambda y.yv)z$
$(\lambda x.(\lambda y.yx)z)v \to_\beta (\lambda x.zx)v$ $\quad \neq_\alpha$

Further reduction: $(\lambda y.yv)z \to_\beta zv$
$(\lambda x.zx)v \to_\beta zv$ $\quad =_\alpha$

2) $(\lambda x.xx)(\lambda x.xx) \to_\beta (\lambda x.xx)(\lambda x.xx)$

temporarily ignore Barendregt convention

Normal forms and confluence:

Compare: $(3+7)\cdot(8-2) \to 10\cdot(8-2) \to 10\cdot 6 \to 60$
$(3+7)\cdot(8-2) \to (3+7)\cdot 6 \to 10\cdot 6 \to 60$

$ax^2+bx+c \leftarrow a(x^2+\frac{b}{a}\cdot x)+c \leftarrow a\cdot(x^2+2\frac{b}{2a}\cdot x)+c \leftarrow$
$\leftarrow a\cdot(x+\frac{b}{2a})^2 - \frac{b^2}{4a}+c$

(motivation for inverse β-reduction)

$\Rightarrow c-\frac{b^2}{4a}$ is extreme value for $x=-\frac{b}{2a}$

Definition: M is $\boxed{\text{in } \beta\text{-normal form}}$ (short: in β-nf) if M does not contain any redex

M $\boxed{\text{has a } \beta\text{-normal form}}$ (short: has a β-nf) or is $\boxed{\beta\text{-normalising}}$ if $\exists N$ in β-nf: $M =_\beta N$. N is then called a $\boxed{\beta\text{-normal form of}}$ M (short: β-nf of)

A β-nf of M is considered as the "outcome" of M

Lemma: If M is in β-nf, then $(M \twoheadrightarrow_\beta N \Rightarrow M \equiv N)$

Examples:
1) $(\lambda x.(\lambda y.yx)z)v \twoheadrightarrow_\beta zv$, so zv is a β-nf of $(\lambda x.(\lambda y.yx)z)v$
2) Let $\Omega := (\lambda x.xx)(\lambda x.xx)$. Then Ω does not have a β-nf. $(\Omega \rightarrow_\beta \Omega)$
3) Let $\Delta := \lambda x.xxx$. Then $\Delta\Delta \rightarrow_\beta \Delta\Delta\Delta \rightarrow \Delta\Delta\Delta\Delta \rightarrow \dots$
and $\Delta\Delta$ again does not have a β-nf
4) $(\lambda u.v)\Omega \rightarrow_\beta (\lambda u.v)\Omega$
$(\lambda u.v)\Omega \rightarrow_\beta v$
} so $(\lambda u.v)\Omega$ has a β-nf but it may not be reached if the wrong redex is chosen repeatedly

Remark: From example 2) it follows that the converse of the above lemma is not true:

$\Omega \twoheadrightarrow_\beta \Omega$ and $\Omega \equiv \Omega$ but Ω is not in β-nf

Definition: $\boxed{\text{reduction path}}$

A ~~finite~~ reduction path from M is a sequence $N_0, N_1, \dots, N_n$ such that $N_0 \equiv M$ and $N_i \rightarrow_\beta N_{i+1}$ for $i \in \{0, \dots, n-1\}$

An infinite reduction path from M is an infinite sequence $N_0, N_1, \dots$ such that $N_0 \equiv M$ and $N_i \rightarrow_\beta N_{i+1}$ for $i \in \mathbb{N}_0$

Definition: $\boxed{\text{weak normalisation}}$, $\boxed{\text{strong normalisation}}$

1) M is $\boxed{\text{weakly normalising}}$ if $\exists N$ in β-nf : $M \twoheadrightarrow_\beta N$
2) M is $\boxed{\text{strongly normalising}}$ if there are no infinite reduction paths starting from M

Examples: $(\lambda u.v)\Omega$ is weakly normalising
$(\lambda x.(\lambda y.yx)z)v$ is strongly normalising
Ω and $\Delta\Delta$ are not weakly normalising (and thus not strongly normalising)

Theorem (Church–Rosser, confluence)

Let $M, N_1, N_2 \in \Lambda$ such that $M \twoheadrightarrow_\beta N_1$, $M \twoheadrightarrow_\beta N_2$

Then $\exists N_3 \in \Lambda : N_1 \twoheadrightarrow_\beta N_3$, $N_2 \twoheadrightarrow_\beta N_3$

$$\begin{tikzcd} & M \arrow[dl, "\beta"'] \arrow[dr, "\beta"] & \\ N_1 \arrow[dr, "\beta"'] & & N_2 \arrow[dl, "\beta"] \\ & N_3 & \end{tikzcd}$$

Proof: hard. Barendregt 1981, p. 62

Important consequence: $\boxed{\text{Outcome independent of order of calculation!}}$

Examples: $\boxed{\text{compare}}$

$$\begin{tikzcd} & (\lambda x.(\lambda y.yx)z)v \arrow[dl, "\beta"'] \arrow[dr, "\beta"] & \\ (\lambda y.yv)z \arrow[dr, "\beta"'] & & (\lambda x.zx)v \arrow[dl, "\beta"] \\ & zv & \end{tikzcd}$$

$$\begin{tikzcd} & (3+5)\cdot(7-3) \arrow[dl] \arrow[dr] & \\ 8\cdot(7-3) \arrow[dr] & & (3+5)\cdot 4 \arrow[dl] \\ & 8\cdot 4 \arrow[d] & \\ & 32 & \end{tikzcd}$$

Corollary: If $M =_\beta N$ then there is a L such that $M \twoheadrightarrow_\beta L$ and $N \twoheadrightarrow_\beta L$

Proof: see Corollary 1.9.9.

Lemma: 1) If N is a β-nf of M, then $M \twoheadrightarrow_\beta N$ (if an outcome exists it can be reached by "forward calculation")
2) A λ-term has at most one β-nf (there cannot be two different outcomes)

Proof: see Lemma 1.9.10

Fixed point theorem

Theorem

$\forall L \in \Lambda : \exists M \in \Lambda : LM =_\beta M$

Proof: Let $M :\equiv (\lambda x. L(xx))(\lambda x. L(xx))$

Then $M \to_\beta L((\lambda x. L(xx))(\lambda x. L(xx))) \equiv LM$,

so $LM =_\beta M$.

Definition: $\boxed{\text{fixed point combinator}}$, $\boxed{Y\text{-combinator}}$

$$Y :\equiv \lambda y. (\lambda x. y(xx))(\lambda x. y(xx))$$

If $L \in \Lambda$ then YL is a fixed point of Y:

$$YL \to_\beta (\lambda x. L(xx))(\lambda x. L(xx)) \to_\beta L((\lambda x. L(xx))(\lambda x. L(xx))) =_\beta L(YL)$$

Consequence: All recursive equations are solvable!

$M =_\beta \ldots M \ldots$

Define: $L :\equiv \lambda z. \ldots z \ldots$

Then $LM \to_\beta \ldots M \ldots$

Hence it suffices to find an M such that $M =_\beta LM$, but such an M always exists!
↑ to solve $M =_\beta \ldots M \ldots$

Examples: 1) Find $M \in \Lambda$ such that $Mx =_\beta xMx$:

Rephrase: Find $M \in \Lambda$ such that $M =_\beta \lambda x. xMx$
because then $Mx =_\beta xMx$

Define: $L := \lambda y. (\lambda x. xyx)$

Then $LM \to_\beta \lambda x. xMx$

Find M such that $M =_\beta LM$: $\underline{M :\equiv YL}$

2) fac $x =_\beta$ if (iszero x) then 1 else mult x (fac (pred x))

compare $!$: $\mathbb{N} \to \mathbb{N}$

$$n \mapsto n! := \begin{cases} 1 & \text{if } n = 0 \\ n(n-1)! & \text{if } n \neq 0 \end{cases}$$

Conclusion: see 1.11.