Lecture 5 **3.2.2025**

Today's agenda:
Bound, free variables
Common functions
Substitution, Beta-equality rules
--

Note on free/bound variables: we have seen this earlier in C programming language: global, local variables, integral calculus, limits, first order logic. When a variable is placed within the scope of, say, an integral or limit it becomes bound; otherwise free. So (λx. M) makes x bound.

Definition:

free(x) = {x}

free(M N) = free M $\cup$ free N // set union

free(λx. M) = free M \ {x} // set difference

Closed lambda term or a combinator: a pure lambda term with no free variables

Let us revisit the term: (λx. (λx. x + 1))

First we consider the term in red—(λx. x + 1)

We can use the rule: free(λx. M) = free M \ {x} in the above

So free(λx. x + 1) = free M \ {x}

= {x} \ {x} = $\varnothing$ [emptyset]

which is indeed true since there is no free variable in λx. x + 1

Now we take the entire term (λx. λx. x + 1)

We use the rule again: the term in red has no free variable; so $\varnothing$ \ {x} = $\varnothing$

This means that there is no free variable in the given term.

Thus when we do ((λx. (λx. x + 1)) 1) we get (λx. x + 1), since there is no free occurrence of the outer x in the body [red].

However, if we do ((λx. x + 1) 1), we get 1+1. Here x occurs free in "x+1" so it is substituted by 1. Note that this is not the case in the previous example.

Another example: (λy. (λz. ((x z) (y z)))) give the scopes of the variables.

<u>Occurrence of a variable is free if it is not within the scope of any binding within the term.</u>

Examples of some common functions:

1. **Identity function:** $I = \lambda x.\ x$
   What is id M ? $(\lambda x.\ x)\ M = M$

2. **First**: $K = \lambda\, x.\ \lambda\, y.\ x$
   First M N = $(\ (\lambda\, x.\ (\lambda\, y.\ x)\ M)\ N) = ((\lambda\, y.\ M)\ N) = M$

3. **Second:** $\lambda\, x.\ \lambda\, y.\ y$

4. **Apply**: $\lambda\, f.\ \lambda\, x.\ f\ x$ HOF

   See the difference between f and x. here x is a variable, f is a function.
   So the arguments of Apply are (i) function (ii) variable

5. **Twice**: $\lambda\, f.\ \lambda\, x.\ f\ (f\ x)$ parenthesis is required [why is it so?] HOF

6. **Comp** = $\lambda\, f.\ \lambda\, g.\ \lambda\, x.\ g\ (f\ x)$ parenthesis is required HOF

   the arguments of Comp are (i) function (ii) function (iii) variable

Thus by looking at the arguments, we can figure out whether it is a variable or a function.

Higher order functions (HOF) are an integral part of LC and any functional PL.

Substitution:

What happens when an abstraction $(\lambda x.\ M)$ is applied to an argument N? The result is obtained by substituting <u>all free occurrences</u> of x in M by N.

e.g., $((\lambda x.\ x + x)\ 2)$ here x occurs free in M; so after substitution the term becomes $2 + 2$; it does not become $2 + x$ or $x + 2$

Formally,

β-equality:

$$((\lambda x.\ M)\ N) =_\beta M\ [N/x] \qquad (\beta\text{-axiom})$$

M [N/x] means replace/substitute <u>all free occurrences</u> of x in M by N

Thus, if x does not occur free in M, then $((\lambda x.\ M)\ N)$ will be M.

$((\lambda x.\ x)\ u) =_\beta u$ and $((\lambda x.\ y)\ u) =_\beta y$

$((\lambda x.\ x + 1)\ 2) =_\beta 2 + 1$ $((\lambda x.\ x + x)\ 2) =_\beta 2 + 2$

Rewriting $((\lambda x.\ M')\ N)$ to $M'[N/x]$ is called beta-reduction. [beta-reduction means term-rewriting]

In order to rename bound variables systematically:

$((\lambda x.\ M)\ N) =_\beta \lambda z.\ M\ [z/x]$ provided that z is not free in M (α-axiom)

e.g., $\lambda x.\ x =_\beta \lambda y.\ y$ and $\lambda x.\ \lambda y.\ x =_\beta \lambda u.\ \lambda v.\ u$

in $\lambda x.\ x + y$ we cannot rename x by y because y is free in M

$((\lambda x.\ M)\ N) =_\beta M\ [N/x]$ (β-axiom)

$((\lambda x.\ M)\ N) =_\beta \lambda z.\ M\ [z/x]$ provided that z is not free in M (α-axiom)

$M =_\beta M$ (idempotence axiom)

$$\frac{M =_\beta N}{N =_\beta M}$$ (commutative rule)

To be read as: if $M =_\beta N$ then $N =_\beta M$

$$\frac{M =_\beta N \qquad N =_\beta P}{M =_\beta P}$$ (transitive rule)

To be read as: if $M =_\beta N$ **and** $N =_\beta P$ then $M =_\beta P$

$$\frac{M =_\beta M' \qquad N =_\beta N'}{M\ N =_\beta M'\ N'}$$ (congruence rule)

$$\frac{M =_\beta M'}{\lambda x.\ M =_\beta \lambda x.\ M'}$$ (congruence rule)

Axioms and rules for beta-equality

End of lecture.