λ-Calculus Cheat Sheet

Overview of the Untyped Lambda Calculus

http://thesyntacticsugar.blogspot.com

Definitions

- V is the set of variables
- Λ is the set of lambda terms

The λ-terms are defined as:

$$x \in V \Rightarrow x \in \Lambda$$

If x is a variable, then x is a lambda term.

$$M, N \in \Lambda \Rightarrow (M\ N) \in \Lambda$$

If M and N are lambda terms, then (M N) is a lambda term, called an application. The function M is applied to the input N.

$$M \in \Lambda, x \in V \Rightarrow (\lambda x.M) \in \Lambda$$

If M is a lambda term, and x is a variable, then λx.M is a lambda term; an abstraction of the anonymous function x → M. A lambda abstraction **binds** the free variable x in M.

To summarize, a λ-term can be:

- a **variable**, e.g., x;
- an **application**, e.g., M N; or
- a function **abstraction**, e.g., λx.M.

Nothing else is a λ-term.

Alonzo Church (1903 – 1995)

Specification in Backus-Naur Form

```
<term>  ::=  <variable>
           | ( <term> <term> )
           | ( λ <variable> . <term> )
```

Syntax

- Application associates to the left:

  M P Q means ((M P) Q).

- λ-abstractions extend as far to the right as possible:

  λx.M P Q means λx.(M P Q).

Shorthand notation

- λxyz.x is shorthand for λx.λy.λz.x.
- Outermost parentheses are usually omitted.
- λx(M) is λx.M.

Free variables

The free variables (FV) of a λ-term is defined, inductively, as:

1. FV(v) = {v}
2. FV(M N) = FV(M) **U** FV(N)
3. FV(λx.M) = FV(M) − {x}

M is a closed λ-term (also known as *combinator*) if FV(M) = ∅. The set of closed λ-terms is denoted by Λ^o.

Standard combinators

I ≡ λx.x
K ≡ λxy.x
$\mathbf{K_*}$ ≡ λxy.y
S ≡ λxyz.xz(yz)

Y-Combinator

The combinator **Y** provides a way to represent recursion:

Y ≡ λf.(λx.f(xx))(λx.f(xx))

Alpha-convertibility

Terms that differ only by alpha-conversion (renaming of bound variables) are called *α-equivalent*:

$$\lambda z.z =_\alpha \lambda x.x$$

Beta-reduction

β-reduction formalizes the notion of a computation in the rule,

$$(\lambda x.M)N \rightarrow_\beta M[x := N]$$

where x := N denotes the substitution of any *free* occurrence of x in M with the value N.

M → N … M reduces to N in exactly one step.

M —» N … M reduces to N in zero or more steps.

Eta-convertibility

η-conversion expresses the idea of extensionality:

$$(\lambda x.M)x =_\eta M \qquad \text{if } x \notin FV(M)$$

Normal Form

A lambda expression is in *normal form* if no sub-expression can be reduced. A term is said to *have a normal form* if it can be reduced to a term in normal form.

Head Normal Form

An expression is in *head normal form* if the outermost application cannot be reduced, i.e., if there is no beta-redex in head position.

Boolean logic

True	**T**	≡ λxy.x
False	**F**	≡ λxy.y (equal to numeral 0)
And		≡ λpq.pqp
Or		≡ λpq.ppq
Not		≡ λpab.pba
If-then-else		≡ λpab.pab

Arithmetic

Addition	$\mathbf{A_+}$	≡ λxypq.xp(ypq)
Multiplication	$\mathbf{A_*}$	≡ λxyz.x(yz)
Exponentiation	$\mathbf{A_{exp}}$	≡ λxy.yx
Successor	$\mathbf{S^+}$	≡ λnfx.f(nfx)

Is-Zero Predicate

Is zero ≡ λn.n (λx.False) True

Substitution

The capture-avoiding substitution rules are defined as follows:

- For a **variable**, y;

$$y[x := N] \equiv \begin{cases} N & \text{if } x = y \\ y & \text{if } x \neq y \end{cases}$$

- For an **application**, (M N);

$$(M\ N)[x := P] \equiv M[x := P]\ N[x := P]$$

- For an **abstraction**, λy.M;

$$(\lambda y.M)[x := N] \equiv \begin{cases} \lambda y.M & \text{if } x = y \\ \lambda y.(M[x := N]) & \text{if } x \neq y \text{ and } y \notin FV(N) \end{cases}$$

Note: *M[x := N] is written M[N / x] in some books.*

Freshness condition

Alpha-conversion is sometimes necessary to avoid changing the meaning of functions (name clashes). For example:

$$(\lambda x.y)[y := x] \equiv \lambda x.(y[y := x]) \equiv \lambda x.x \Rightarrow \lambda x.y \equiv \lambda x.x\ (!!)$$

This substitution does **not** meet the condition;
(λy.M)[x := N] requires that y ∉ FV(N) (y is *fresh* for N).

After α-conversion of λx.y to λz.y, we get:

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

Fixed Point Theorem

$\forall F\ \exists X\ FX = X$ where F, X ∈ Λ

For all λ-terms F, there exists a λ-term X such that **λ** ⊢ FX = X.

Church numerals

$\mathbf{c_0}$ ≡ λf.λx.x
$\mathbf{c_1}$ ≡ λf.λx.f x
$\mathbf{c_2}$ ≡ λf.λx.f(f x)
$\mathbf{c_3}$ ≡ λf.λx.f(f (f x))

$\mathbf{c_n} \equiv \lambda f.\lambda x.f^n(x)$

Indefinite value

The indefinite value is denoted

Ω ≡ ωω

where **ω** ≡ λx.xx.

Church-Rosser Theorem

If a term M can be reduced to both N and P, then there must exist a term Q (possibly equal to N or P) to which both N and P can be reduced.