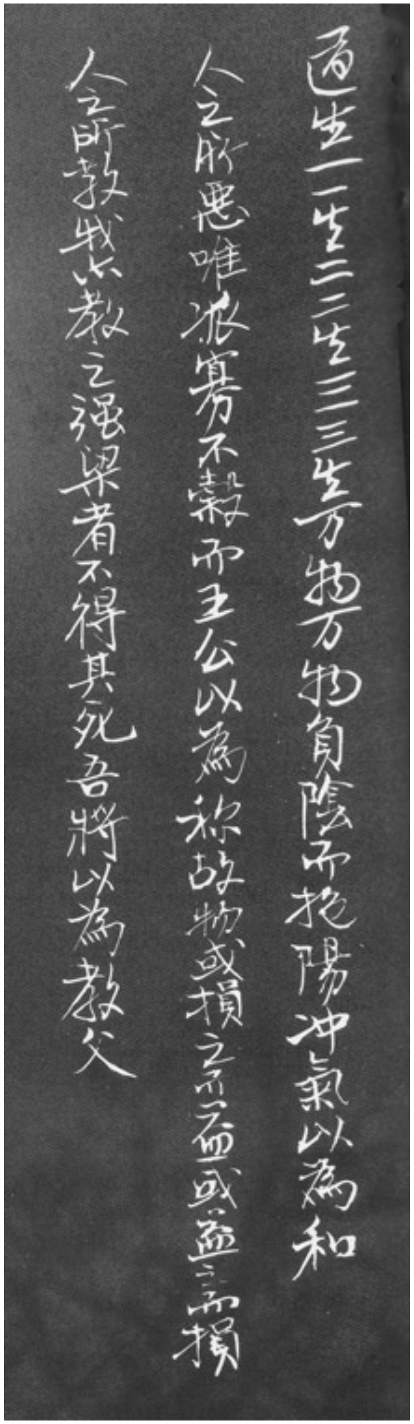

λ Calculus

--- Basics

Yan Huang

λ Calculus

Calculus

A branch of mathematics that studies *limits*, *derivatives*, *integrals*, and *infinite series*.

Examples

$$d(uv) = v(du) + u(dv)$$ The product rule

$$\frac{dy}{dx} = \frac{dy}{du}\frac{du}{dx}$$ The chain rule

Calculus

is just a bunch of *rules*

for manipulating

symbols.

Calculus

A branch of mathematics that studies *limits*, *derivatives*, *integrals*, and *infinite series*.

Examples

$$d(uv) = v(du) + u(dv)$$ The product rule

$$\frac{dy}{dx} = \frac{dy}{du}\frac{du}{dx}$$ The chain rule

λ Calculus Formalism (Grammar)

Key words: λ . ( ) **terminals**

Defining λ-*term*:

$$
\begin{aligned}
term \rightarrow\ & variable && \text{e.g., } x \quad y \\
|\ & (\ term\ ) && \text{e.g., } (x) \quad (y) \\
|\ & \lambda\ variable\ .\ term && \text{e.g., } \lambda\, x\ .\ x \\
|\ & term\ term && \text{e.g., } (\lambda\, x\ .\ x)\ y
\end{aligned}
$$

Humans can give meaning to those symbols to describe computations.

λ Calculus Formalism (Rules)

α-reduction (renaming)

$\lambda x . x$
$\rightarrow_\alpha \lambda y . y$

$$\lambda y . M \longrightarrow_\alpha \lambda v . M\,[y \mapsto v]$$

where v does not occur in M.

β-reduction (substitution)

$$(\lambda x . M)\, T \longrightarrow_\beta \boxed{M\,[x \mapsto T]}$$

Replace all bounded x of M with T

$$(\lambda\, x\, .\, x)\ y \rightarrow_{\beta} y$$

$$x\,[\,x \mapsto \underbrace{(\lambda y.\, y)}\,]$$

$$(\lambda\, x\, .\, x\,)\,(\lambda\, y\, .\, y) \rightarrow_\beta \quad \lambda y.\, y$$

Free and Bound variables

- Learn by examples
  - $\lambda\, x\, .\, x\, y$

    x is bound, while y is free;
  - $(\lambda\, x\, .\, x)\; (\lambda\, y\, .\, y\, x)$

    x is bound in the 1st function, but free in the 2nd function
  - $\lambda\, x\, .(\lambda\, y\, .\, y\, x)$

    x and y are both bound variables.

Be careful about β-Reduction

$$(\lambda\, x\, .\, M\,)\; T \rightarrow_{\beta} M\,[x \mapsto T\,]$$

Replace all bounded x of M with T

If a free variable v in T occurs bound in M, we rename all appearances of v in M before the substitution.

$$(\lambda x \,.\, (\underbrace{\lambda y \,.\, (x\, y)}))\, y \rightarrow_{\beta} \quad (\lambda x.\ (\lambda z.\ (x\, z)))\, y$$

$$\rightarrow_{\beta} \lambda z.\ y\, z$$

Syntactic Sugar

$$\lambda x\,.\,(\lambda y\,.\,M) \equiv \lambda x\,.\,\lambda y\,.\,M$$
$$\equiv \lambda x\,y\,.\,M$$

$$\lambda x\,.\,\lambda y\,.\,\lambda z\,.\,M \equiv \lambda x\,y\,\mathrm{z}.\,M$$

λ-terms extend as far *right* as possible.

Treat "**.**" as a *lowest-priority, right-associative* operator.

$$\mathbf{S} \equiv \lambda\, w\, y\, x\, .\, y\, (w\, y\, x)$$

$$\mathbf{Z} \equiv \lambda\, s\, z\, .\, z$$

$$\mathbf{SZ} \equiv (\lambda\, w\, y\, x\, .\, y\, (w\, y\, x))\ (\lambda\, s\, z\, .\, z)$$

$$\rightarrow_{\beta} \lambda\, y\, x\, .\, y\, ((\lambda\, s\, z\, .\, z)\, y\, x)$$

$$\rightarrow_{\beta} \lambda\, y\, x\, .\, y\, (x)$$

$$\equiv\ \lambda\, y\, x\, .\, y\, x$$

Computing Model for λ Calculus

- ***redex***: a *term* of the form $(\lambda x.\, M)N$

  Something that can be β-reduced

- An expression is in ***normal form*** if it contains no *redexes* (*redices*).
- To evaluate a λ-*term*, keep doing reductions until you get to *normal form*.

β-Reduction represents all the computation capability of λ Calculus.

Another exercise

$$(\lambda f.\ ((\lambda x. f\,(xx))\ (\lambda x. f\,(xx))))\ (\lambda z.z)$$

Possible Answer

$$\underline{(\lambda f\,.\,((\lambda x\,.\,f\,(x\,x))\;(\lambda x\,.\,f\,(x\,x))))}\;\underline{(\lambda z.z)}$$

$$\to_\beta\;\underline{(\lambda x\,.\underline{(\lambda z.z)}(x\,x))\;(\lambda x\,.\,\underline{(\lambda z.z)}(x\,x))}$$

$$\to_\beta\;(\lambda z\,.\,z)\;(\lambda x\,.\,(\lambda z.z)(x\,x))\;(\lambda x\,.\,(\lambda z.z)(x\,x))$$

$$\to_\beta\;(\lambda x\,.\,(\lambda z.z)(x\,x))\;(\lambda x\,.\,(\lambda z.z)(x\,x))$$

$$\to_\beta\;(\lambda z\,.\,z)\;(\lambda x.(\lambda z.z)(x\,x))\;(\lambda x\,.\,(\lambda z.z)(x\,x))$$

$$\to_\beta\;(\lambda x\,.\,(\lambda z.z)(x\,x))\;(\lambda x\,.\,(\lambda z.z)(x\,x))$$

$$\to_\beta\;\ldots$$

Alternate Answer

$$(\lambda f\,.\,((\lambda x\,.\,f\,(x\,x))\;(\lambda x\,.\,f\,(x\,x))))\;(\lambda z.z)$$

$$\rightarrow_\beta (\lambda x\,.\,(\lambda z.z)(x\,x))\;(\lambda x.\;(\lambda z.z)(x\,x))$$

$$\rightarrow_\beta (\lambda x\,.\,x\,x)\;(\lambda x\,.\,(\lambda z.z)(x\,x))$$

$$\rightarrow_\beta (\lambda x\,.\,x\,x)\;(\lambda x\,.\,x\,x)$$

$$\rightarrow_\beta (\lambda x\,.\,x\,x)\;(\lambda x\,.\,x\,x)$$

$$\rightarrow_\beta \ldots$$

Be very afraid!

Some λ-calculus terms can be β-reduced forever!

The order in which you choose to do the reductions might change the result!

Take on Faith

- All ways of choosing reductions will produce the same normal form (but some might never terminate).

- If we always *apply the outermost redex first*, we will find the normal form if there is one.
  - This is *normal order reduction* – corresponds to normal order (lazy) evaluation