MATHEMATICS FOR INFORMATION SCIENCE
NO.7 LAMBDA CALCULUS AND COMPUTABILITY

Tatsuya Hagino

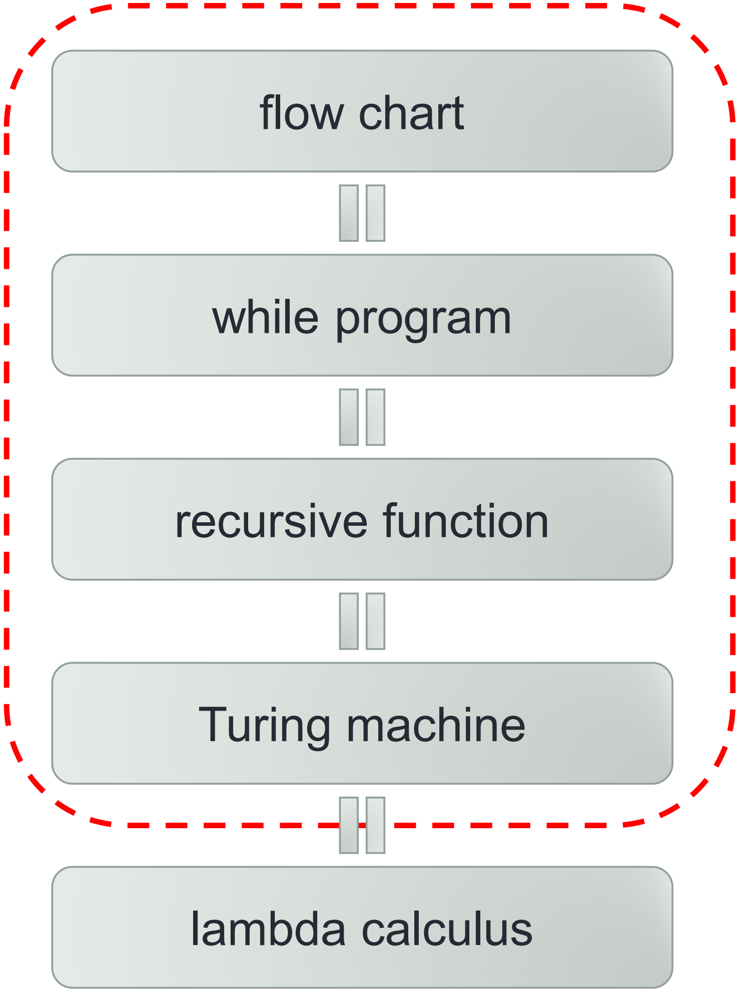
hagino@sfc.keio.ac.jp

Slides URL

https://vu5.sfc.keio.ac.jp/slide/

So far

- Computation
  - flow chart program
  - while program
  - recursive function
    - primitive recursive function
    - minimization operator
  - Turing machine
    - undecidable problems
  - Lambda Calculus
    - function abstraction
    - function application



λ Representation

- λ representation F of a partial function $f: N^n \to N$ is:
  - If $f(k_1, k_2, \dots, k_n) = k$, then

$$F[k_1][k_2] \dots [k_n] \overset{\alpha\beta}{\Rightarrow} [k]$$

  where $[k_i]$ is the λ representation of natural number k_i.

- $[f] \equiv F$ is a λ representation of f.

True, False and Pairs

- λ representation of true and false:
  - $[\text{true}] \equiv \lambda x\ y.\ x$
  - $[\text{false}] \equiv \lambda x\ y.\ y$
  - $[\text{true}]M\ N \equiv\ (\lambda x\ y.\ x)M\ N \overset{\alpha\beta}{\Rightarrow}\ M$
  - $[\text{false}]M\ N \overset{\alpha\beta}{\Rightarrow}\ N$
- Pairs:
  - $[M, N] \equiv \lambda x.\ x\ M\ N$
  - $\pi_1 \equiv \lambda x.\ x[\text{true}]$
  - $\pi_2 \equiv \lambda x.\ x[\text{false}]$
  - $\pi_1[M, N] \equiv\ (\lambda x.\ x[\text{true}])(\lambda x.\ x\ M\ N) \overset{\alpha\beta}{\Rightarrow}\ (\lambda x.\ x\ M\ N)[\text{true}] \overset{\alpha\beta}{\Rightarrow}\ [\text{true}]M\ N \overset{\alpha\beta}{\Rightarrow}\ M$
  - $\pi_2[M, N] \overset{\alpha\beta}{\Rightarrow}\ N$

Natural Number

- Natural number:
  - $[0] \equiv \lambda x\ y.\ y$
  - $[1] \equiv \lambda x\ y.\ x\ y$
  - $[2] \equiv \lambda x\ y.\ x(x\ y)$
  - $[3] \equiv \lambda x\ y.\ x\big(x(x\ y)\big)$

    $\vdots$
  - $[n] \equiv \lambda x\ y.\ \overbrace{x\big(x(\cdots(x}^{n}\ y)\cdots)\big)$
- Arithmetic:
  - $[\mathrm{suc}] \equiv \lambda x\ y\ z.\ y(x\ y\ z)$
  - $[\mathrm{add}] \equiv \lambda x\ y\ z\ w.\ x\ z(y\ z\ w)$
  - $[\mathrm{mul}] \equiv \lambda x\ y\ z\ w.\ x(y\ z)w$
  - $[\mathrm{pred}] \equiv \lambda x\ y\ z.\ x\big(\lambda u\ v.\ v(u\ y)\big)(\lambda a.\ z)(\lambda a.\ a)$
  - $[\mathrm{zero?}\,] \equiv \lambda x.\ x(\lambda x.\ [\mathrm{false}])[\mathrm{true}]$

Calculation in λ Representation

- $\lceil \mathrm{suc} \rceil \lceil 2 \rceil \equiv (\lambda x\ y\ z.\ y(x\ y\ z))(\lambda x\ y.\ x(x\ y))$

$$\stackrel{\alpha\beta}{\Rightarrow} \ldots$$

$$\stackrel{\alpha\beta}{\Rightarrow} \ldots$$

$$\stackrel{\alpha\beta}{\Rightarrow} \lambda x\ y.\ x(x(x\ y))$$

$$\equiv \lceil 3 \rceil$$

- $\lceil \mathrm{add} \rceil \lceil 3 \rceil \lceil 2 \rceil \equiv (\lambda x\ y\ z\ w.\ x\ z(y\ z\ w))\left(\lambda x\ y.\ x(x(x\ y))\right)(\lambda x\ y.\ x(x\ y))$

$$\stackrel{\alpha\beta}{\Rightarrow} \ldots$$

$$\stackrel{\alpha\beta}{\Rightarrow} \ldots$$

$$\stackrel{\alpha\beta}{\Rightarrow} \ldots$$

$$\stackrel{\alpha\beta}{\Rightarrow} \lambda x\ y.\ x\left(x\left(x(x(x\ y))\right)\right) \equiv \lceil 5 \rceil$$

Recursive Functions

- Primitive Recursive Functions:
  - Basic Functions
    - $zero : N^0 \rightarrow N \qquad zero() = 0$
    - $suc : N \rightarrow N \qquad suc(x) = x + 1$
    - $\pi_i^n : N^n \rightarrow N \qquad \pi_i^n(x_1, \dots, x_n) = x_i$
  - Composition of primitive recursive functions
    - $f(x_1, x_2, \dots, x_n) = g\big(h_1(x_1, x_2, \dots, x_n), \dots, h_m(x_1, x_2, \dots, x_n)\big)$
  - Define a function using primitive recursion
    - $f(x_1, \dots, x_n, zero()) = g(x_1, \dots, x_n)$
    - $f\big(x_1, \dots, x_n, suc(y)\big) = h\big(x_1, \dots, x_n, y, f(x_1, \dots, x_n, y)\big)$

- Recursive Functions:
  - Minimization operator
    - $f(x_1, \dots, x_n) = \mu_y(g(x_1, \dots, x_n, y) = 0)$

Primitive Recursive Function

- Basic functions:
  - $\lceil \mathrm{zero} \rceil \equiv \lceil 0 \rceil \equiv \lambda x\ y.\, y$
  - $\lceil \mathrm{suc} \rceil \equiv \lambda x\ y\ z.\, y(x\ y\ z)$
  - $\lceil \pi_i^n \rceil \equiv \lambda x_1\ x_2 \cdots\ x_n.\, x_i$

- Function composition:
  - $f(x_1, x_2, \ldots, x_n) = g\big(h_1(x_1, x_2, \ldots, x_n), \ldots, h_m(x_1, x_2, \ldots, x_n)\big)$ のとき,

$$\lceil f \rceil \equiv \lambda x_1\ x_2 \cdots\ x_n.\, \lceil g \rceil(\lceil h_1 \rceil x_1\ x_2 \cdots\ x_n) \cdots (\lceil h_m \rceil x_1\ x_2 \cdots x_n)$$

- Example:
  - $\mathrm{dsuc}(x) = \mathrm{suc}\big(\mathrm{suc}(x)\big)$
  - $\lceil \mathrm{dsuc} \rceil \equiv \lambda x.\, \lceil \mathrm{suc} \rceil(\lceil \mathrm{suc} \rceil x)$

Primitive Recursion

- For simplicity, let us consider for only two argument case.
  - $f(x, zero()) = g(x)$
  - $f(x, suc(y)) = h(x, y, f(x, y))$

- Construction of $F \equiv \lceil f \rceil$
  - F must satisfy:
    $F\ x\ y \overset{\alpha\beta}{\Rightarrow} \lceil \text{zero?} \rceil y (\lceil g \rceil x) \left(\lceil h \rceil x (\lceil \text{pred} \rceil y) (F\ x (\lceil \text{pred} \rceil y)) \right)$
  - F is a fixed point of the following M:
    $M \equiv \lambda f\ x\ y. \lceil \text{zero?} \rceil y (\lceil g \rceil x) \left(\lceil h \rceil x (\lceil \text{pred} \rceil y) (f\ x (\lceil \text{pred} \rceil y)) \right)$

  - Using Curry's fixed point operator $Y \equiv \lambda y. \left(\lambda x. y(xx)\right)\left(\lambda x. y(xx)\right)$:
    $F \equiv Y\ M$
    $F \equiv Y \left(\lambda f\ x\ y. \lceil \text{zero?} \rceil y (\lceil g \rceil x) \left(\lceil h \rceil x (\lceil \text{pred} \rceil y) (f\ x (\lceil \text{pred} \rceil y)) \right) \right)$

Minimization Operator

- For simplicity, let us consider for only one argument case
  - $f(x) = \mu_y(g(x,y) = 0)$

- Construction of $F \equiv \lceil f \rceil$
  - Let H be a λ expression which satisfies:
    $H\ x\ y \overset{\alpha\beta}{\Rightarrow} \lceil \mathrm{zero?} \rceil (\lceil g \rceil x\ y) y \big(H\ x(\lceil \mathrm{suc} \rceil y)\big)$

    H can be defined using Curry's fixed point operator Y:
    $H \equiv Y\left(\lambda h\ x\ y. \lceil \mathrm{zero?} \rceil (\lceil g \rceil x\ y) y \big(h\ x(\lceil \mathrm{suc} \rceil y)\big)\right)$

  - $F \equiv \lambda x. H\ x \lceil 0 \rceil$
    $F \equiv \lambda x. Y\left(\lambda h\ x\ y. \lceil \mathrm{zero?} \rceil (\lceil g \rceil x\ y) y \big(h\ x(\lceil \mathrm{suc} \rceil y)\big)\right) x \lceil 0 \rceil$

Computability

- Any computable function can be represented as a λ expression.
  - A computable function is a recursive function.
  - A recursive function has a λ representation.

- A function with a λ representation is computable.
  - A λ representation can be evaluated (or executed) using left most β reduction.
  - A λ expression can be encoded into a number.
  - Write a program to simulate the rightmost β reduction.

η (eta) Conversion

- Extensionality of functions
  - Two functions are the same if and only if they give the same result for all arguments.
  - $f = g \Leftrightarrow \forall x\big(f(x) = g(x)\big)$

- Extensionality does not hold for $\alpha\beta$ equivalence.
  - If $P \overset{\alpha\beta}{\Leftrightarrow} Q$, for any $x \notin \mathrm{FV}(PQ)$, $P\ x \overset{\alpha\beta}{\Leftrightarrow} Q\ x$
  - The converse does not hold: $\lambda x. y\ x \overset{\alpha\beta}{\Leftrightarrow} y$ is not true.

- η conversion:

$$\lambda x. P\ x \overset{\eta}{\rightarrow} P$$

where $x \notin \mathrm{FV}(P)$

Summary

- Lambda expression
- Conversion and reduction
  - α conversion
  - β reduction
  - η reduction
- Computability
  - λ representation
  - λ representation of natural number