

Lambda calculus

Functional models of computation

Ivan Trepakov

NSU Sys.Pro

History

- 1928 - Hilbert's *Entscheidungsproblem*¹
 - Is there an *algorithm* for deciding whether a proposition in first-order logic is true or false?
- Replacement for set theory as foundation of mathematics
 - 1930 - Combinatory logic (*Curry, Schönfinkel*)
 - 1932 - λ -calculus (*Church*)
 - 1935 - Kleene-Rosser paradox
- Effective computability
 - 1935 - Untyped λ -calculus (*Church, Kleene, Rosser*)
 - 1936 - Turing machine
 - 1936 - Church-Turing thesis
- 1936 - Undecidability of first-order logic
 - Halting problem of Turing machine
 - Equivalence of λ -terms

¹German for “decision problem”

History

- 1928 - Hilbert's *Entscheidungsproblem*¹
 - Is there an *algorithm* for deciding whether a proposition in first-order logic is true or false?
- Replacement for set theory as foundation of mathematics
 - 1930 - Combinatory logic (*Curry, Schönfinkel*)
 - 1932 - λ -calculus (*Church*)
 - 1935 - Kleene-Rosser paradox
- Effective computability
 - 1935 - Untyped λ -calculus (*Church, Kleene, Rosser*)
 - 1936 - Turing machine
 - 1936 - Church-Turing thesis
- 1936 - Undecidability of first-order logic
 - Halting problem of Turing machine
 - Equivalence of λ -terms

David Hilbert

- [Haskell Curry](#)
- Wilhelm Ackermann
- John von Neumann
- Ernst Zermelo
- ...

Alonzo Church

- [Stephen Cole Kleene](#)
- [J. Barkley Rosser](#)
- Alan Turing
- Dana Scott
- Michael O. Rabin
- ...

¹German for "decision problem"

Grammar

$$term ::= \underbrace{var}_{\text{Variable}} \mid \underbrace{(term\ term)}_{\text{Application}} \mid \underbrace{(\lambda var. term)}_{\text{Abstraction}}$$

Examples

$$\lambda x. x \quad (\lambda x. xx)(\lambda y. yy) \quad \lambda f. \lambda x. f(fx)$$

Conventions

- Application is left associative
 $abc = (ab)c$
- Abstraction is right associative
 $\lambda x. \lambda y. x = \lambda x. (\lambda y. x)$
- Consecutive abstractions can be combined
 $\lambda x. \lambda y. x = \lambda xy. x$

Grammar

$$term ::= \underbrace{var}_{\text{Variable}} \mid \underbrace{(term\ term)}_{\text{Application}} \mid \underbrace{(\lambda var. term)}_{\text{Abstraction}}$$

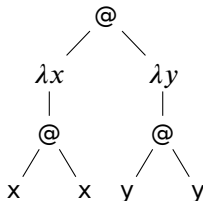
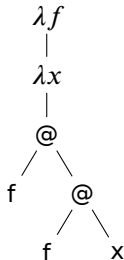
Examples

 $\lambda x. x \quad (\lambda x. xx)(\lambda y. yy) \quad \lambda f. \lambda x. f(fx)$

Conventions

- Application is left associative
 $abc = (ab)c$
- Abstraction is right associative
 $\lambda x. \lambda y. x = \lambda x. (\lambda y. x)$
- Consecutive abstractions can be combined
 $\lambda x. \lambda y. x = \lambda xy. x$

Tree representation

$$\begin{array}{c} \lambda x. x \\ | \\ \lambda x \\ | \\ x \end{array}$$
 $(\lambda x. xx)(\lambda y. yy)$  $\lambda f. \lambda x. f(fx)$ 

Free and bound variables

Free variables

Variable: $FV(x) = \{x\}$

Application: $FV(MN) = FV(M) \cup FV(N)$

Abstraction: $FV(\lambda x. M) = FV(M) \setminus \{x\}$

Bound variables

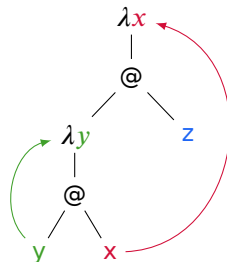
Variable: $BV(x) = \emptyset$

Application: $BV(MN) = BV(M) \cup BV(N)$

Abstraction: $BV(\lambda x. M) = BV(M) \cup \{x\}$

Example

$\lambda x. (\lambda y. yx)z$



Substitution

α -equivalence

β -conversion

β -reduction

β -abstraction

Normal order reduction

First Church-Rosser theorem

Second Church-Rosser theorem

Normal order reduction

Fixed-point combinator

Curry's Y -combinator

$$Y = \lambda f. (\lambda x. f(xx)) (\lambda x. f(xx))$$

Turing's Θ -combinator

$$\Theta = (\lambda xy. x(xxy)) (\lambda xy. x(xxy))$$

Undecidability

Church numerals

Relation to folds

Algebraic data types

Predecessor

Q&A