Symbolic Expressions and Variable Binding
Lecture 1

Masahiko Sato

Graduate School of Informatics, Kyoto University

September 6–10, 2010

Plan of the 5 lectures

1. Overview
   - Background
   - History
   - Problems
   - Our approach
2. Traditional definition of Lambda terms
3. Lambda terms by de Bruijn indices
4. Lambda terms as abstract data type
5. Derivations as abstract data type

Background

A quotation from Knuth

A quotation from Knuth

I can't go to a restaurant and order food because I keep looking at the fonts on the menu.

A quotation from Knuth

I can't go to a restaurant and order food because I keep looking at the fonts on the menu.

Five minutes later I realize that it's also talking about food.

A quotation from Knuth

I can't go to a restaurant and order food because I keep looking at the fonts on the menu.

Five minutes later I realize that it's also talking about food.

Donald Knuth, All Questions Answered, Notices of the AMS, **49**, 2002.

A quotation from Knuth (cont.)

- What is a menu? Or, what is an object in general?
- Can we present a menu, and nothing more? Or, can we present an object, and nothing more?
- No, there is always something more, but we have an ability to forget about them.
- Forgetting is a way of abstractiton.
- An object is identified relative to the level of abstraction.
- The level of abstraction is usually determined outside the system in which obejects are identified.
- How can we manipulate the level of abstractness inside the system?

Quantification

Quantification is a process of binding a varible in an open expression (e.g., sentence). For example, consider an open sentence

$P[a]$: Natural number a is an odd prime number.

From $P[a]$, we obtain a universally quantified sentence

$\forall x.\, P[x]$: Every natural number is an odd prime number.

We can also obtain a existentially quantified sentence

$\exists x.\, P[x]$: Some natural number is an odd prime number.

Three principles of variable

We have the following three principles concerning variables.

1. A variable must be declared first.
2. Then it may be used,
3. within the scope of the declared variable.

Three principles of variable

We have the following three principles concerning variables.

1. A variable must be declared first.
2. Then it may be used,
3. within the scope of the declared variable.

Remark 1 A variable is usually declared toghther with the domain over which the variable ranges. But, this is a semantical aspect of variables. In this lecture, we are intested in syntactical aspects of variables.

Three principles of variable

We have the following three principles concerning variables.

1. A variable must be declared first.
2. Then it may be used,
3. within the scope of the declared variable.

Remark 1 A variable is usually declared toghther with the domain over which the variable ranges. But, this is a semantical aspect of variables. In this lecture, we are intested in syntactical aspects of variables.

Remark 2 Each usage of a variable is associated with a unique declaration of the variable which is determined by the scope of the variable.

Three principles of variable (cont.)

1. A variable must be declared first.
2. Then it may be used,
3. within the scope of the declared variable.

Example: Let n be a natural number. [Then we have:

$$\sum_{i=1}^{n} [i]_i = \frac{n(n+1)}{2}.$$

We also have $\sum_{i=1}^{n} [i+n]_i = \frac{n(n+1)}{2} + n^2.$ $\cdots$ $]_n$

The three principles of variable (cont.)

Symbols used in mathematics are usually classified into three kinds: constants, free variables, and bound variables.

The classification is not absolute but only relative to the practical usage of the language.

Constants have wider scope than free (sometimes called global) variables, and free variables have wider scope than bound (sometimes called local) variables.

Example: Let n be a natural number. [Then we have:

$$\sum_{i=1}^{n} [i]_i = \frac{n(n+1)}{2}.$$

We also have $\sum_{i=1}^{n} [i+n]_i = \frac{n(n+1)}{2} + n^2. \quad \cdots \quad]_n$

The three principles of variable (cont.)

Example: Let n be a natural number. [Then we have:

$$\sum_{i=1}^{n} [i]_i = \frac{n(n+1)}{2}.$$

We also have $\sum_{i=1}^{n} [i+n]_i = \frac{n(n+1)}{2} + n^2.$ $\cdots$ $]_n$

Remark 1 A bound variable is also called an apparent variable or a dummy variable since it does not contribute to the meaning of the expression containing it.

The three principles of variable (cont.)

Example: Let n be a natural number. [then we have:

$$\sum_{i=1}^{n}[i]_i = \frac{n\cdot(n+1)}{2}.$$

We also have $\sum_{i=1}^{n}[i+n]_i = \frac{n\cdot(n+1)}{2} + n^2.$ $\cdots$ $]_n$

Remark 1 A bound variable is also called an apparent variable or a dummy variable since it does not contribute to the meaning of the expression containing it.

Remark 2 Constants have the widest context.

History

- Frege, in his *Begriffsschrift* (1879), used latin letters for global variables and used german letters for local variables.

History

- Frege, in his *Begriffsschrift* (1879), used latin letters for global variables and used german letters for local variables.

See frege.pdf.

History

- Frege, in his *Begriffsschrift* (1879), used latin letters for global variables and used german letters for local variables.
- Gentzen also used different sets of variables for global and local variables.
- Whitehead-Russell (1910) and, later, Gödel and Church used only one sort of letters for both global and local variables.
- Quine and Bourbaki introduced graphical (two dimensional) notation for local variable binding.

History

- Frege, in his *Begriffsschrift* (1879), used latin letters for global variables and used german letters for local variables.
- Gentzen also used different sets of variables for global and local variables.
- Whitehead-Russell (1910) and, later, Gödel and Church used only one sort of letters for both global and local variables.
- Quine and Bourbaki introduced graphical (two dimensional) notation for local variable binding.

See quine.pdf and bourbaki.pdf.

W. Quine, Mathematical Logic, Harvard University Press, 1951.

N. Bourbaki, Elements of Mathimatics 1, Theory of Sets, Addison-Wesley, 1968 (English translation of French original, published in 1957.)

History

- Frege, in his *Begriffsschrift* (1879), used latin letters for global variables and used german letters for local variables.
- Gentzen also used different sets of variables for global and local variables.
- Whitehead-Russell (1910) and, later, Gödel and Church used only one sort of letters for both global and local variables.
- Quine and Bourbaki introduced graphical (two dimensional) notation for local variable binding.
- de Bruijn introduced his indices (and levels) and provided a canonical notation for α-equivalent terms.
- McCarthy introduced abstract syntax.
- Church introduced hihger order abstract syntax (HOAS).

  A. Church, A simple theory of types, JSL 5, 56–69, 1940.

History

- Frege, in his *Begriffsschrift* (1879), used latin letters for global variables and used german letters for local variables.
- Gentzen also used different sets of variables for global and local variables.
- Whitehead-Russell (1910) and, later, Gödel and Church used only one sort of letters for both global and local variables.
- Quine and Bourbaki introduced graphical (two dimensional) notation for local variable binding.
- de Bruijn introduced his indices (and levels) and provided a canonical notation for α-equivalent terms.
- McCarthy introduced abstract syntax.
- Church introduced hihger order abstract syntax (HOAS).
- Pitts (2003) emphasized the importance of equivariance properties on lambda terms.

Traditional Definition of λ-terms

1. If x is a variable, then x is a $\boldsymbol{\lambda}$-term.
2. If M and N are $\boldsymbol{\lambda}$-terms, then $M \cdot N$ is a $\boldsymbol{\lambda}$-term.
3. If x is a variable and M is a $\boldsymbol{\lambda}$-term, then $\boldsymbol{\lambda} x[M]$ is a $\boldsymbol{\lambda}$-term.

Problems with Substitution

$$\begin{aligned} [y/x](\boldsymbol{\lambda} y[x \cdot y]) &= \boldsymbol{\lambda} y[[y/x](x \cdot y)] \\ &= \boldsymbol{\lambda} y[y \cdot y] \quad \text{Wrong!} \end{aligned}$$

Problems with Substitution

$$\begin{aligned}
[y/x](\lambda y[x \cdot y]) &= \lambda y[[y/x](x \cdot y)] \\
&= \lambda y[y \cdot y] \quad \text{Wrong!}
\end{aligned}$$

$$\begin{aligned}
[y/x](\lambda y[x \cdot y]) &= \lambda y[[y/x](x \cdot y)] \\
&= \lambda y[y \cdot y]
\end{aligned}$$

Clash of free and bound variables.

(Can be avoided by using two colors or two sorts of letters for free and bound variables. But, see the next slide.)

Problems with Substitution

$$
\begin{aligned}
[y/x](\lambda y[x \cdot y]) &= \lambda y[[y/x](x \cdot y)] \\
&= \lambda y[y \cdot y] \quad \text{Wrong!}
\end{aligned}
$$

$$
\begin{aligned}
[y/x](\lambda y[x \cdot y]) &= \lambda y[[y/x](x \cdot y)] \\
&= \lambda y[y \cdot y]
\end{aligned}
$$

Clash of free and bound variables.

(Can be avoided by using two colors or two sorts of letters for free and bound variables. But, see the next slide.)

$$
\begin{aligned}
[y/x](\lambda z[x \cdot z]) &= \lambda z[[y/x](x \cdot z)] \\
&= \lambda z[y \cdot z]
\end{aligned}
$$

Renaming of bound variables.

Problems with Substitution (cont.)

Consider the $\boldsymbol{\beta}$-conversion rule:

$$\boldsymbol{\lambda x[M] \cdot N \rightarrow_{\beta} [N/x](M)}$$

Problems with Substitution (cont.)

Consider the β-conversion rule:

$$\lambda x[M] \cdot N \to_\beta [N/x](M)$$

In the Frege-Gentzen notation, the rule becomes:

$$\lambda x[M(x)] \cdot N \to_\beta [N/a](M(a))$$

where a is a fresh free variable.
Moreover, we must explain what is $M(x)$ and what is $M(a)$. But to do this precisely is not so easy.

Traditional Definition of λ-terms (cont.)

1. If x is a variable, then x is a λ-term.
2. If M and N are λ-terms, then $M \cdot N$ is a λ-term.
3. If x is a variable and M is a λ-term, then $\lambda x[M]$ is a λ-term.

We have to identify α-equivalent expressions, for example, $\lambda x[x]$ and $\lambda y[y]$ should be identified.

So, what we see is not in one-to-one correspondence with what it means.

Our approach

My view of mathematics

- Mathematics is a human linguistic activity.
  - A mathematical sentence has both syntax and semantics.
  - Hence, a mathematical sentence is a syntactical object and it talks about semantical objects.
  - We use our mother language, say English, to develop mathematics.
- Mathematics is formalizable.
  - Formalized mathematics is expressed in a formal language, where the formalized language is defined in a natural language, say, English.
  - We regard the formal language as a sub-language of English.
  - This view is possible since English is open ended.
- Mathematics is open ended.

Mathematical Objects

In mathematics we talk about mathematical objects, but what are mathematical objects and how they are constructed?

	Platonism	Constructivism	Formalism

Mathematical Objects

In mathematics we talk about mathematical objects, but what are mathematical objects and how they are constructed?

	Platonism	Constructivism	Formalism
Philosophy	Realism	Conceptualism	Nominalism

Mathematical Objects

In mathematics we talk about mathematical objects, but what are mathematical objects and how they are constructed?

	Platonism	Constructivism	Formalism
Philosophy	Realism	Conceptualism	Nominalism
Mathematics	Logicism	Intuitionism	Formalism

Mathematical Objects

In mathematics we talk about mathematical objects, but what are mathematical objects and how they are constructed?

	Platonism	Constructivism	Formalism
Philosophy	Realism	Conceptualism	Nominalism
Mathematics	Logicism	Intuitionism	Formalism
Comp. Sci.	Denotational semantics	Operational semantics	Axiomatic semantics

Mathematical Objects

In mathematics we talk about mathematical objects, but what are mathematical objects and how they are constructed?

	Platonism	Constructivism	Formalism
Philosophy	Realism	Conceptualism	Nominalism
Mathematics	Logicism	Intuitionism	Formalism
Comp. Sci.	Denotational semantics	Operational semantics	Axiomatic semantics
Ontology	Strong	Weak	Weakest

Mathematical Objects

In mathematics we talk about mathematical objects, but what are mathematical objects and how they are constructed?

	Platonism	Constructivism	Formalism
Philosophy	Realism	Conceptualism	Nominalism
Mathematics	Logicism	Intuitionism	Formalism
Comp. Sci.	Denotational semantics	Operational semantics	Axiomatic semantics
Ontology	Strong	Weak	Weakest
Computation	Neglected	Essential	Essential

Ontology concerns what and computation concerns how.

Mathematical Objects

In mathematics we talk about mathematical objects, but what are mathematical objects and how they are constructed?

	Platonism	Constructivism	Formalism
Philosophy	Realism	Conceptualism	Nominalism
Mathematics	Logicism	Intuitionism	Formalism
Comp. Sci.	Denotational semantics	Operational semantics	Axiomatic semantics
Ontology	Strong	Weak	Weakest
Computation	Neglected	Essential	Essential

Ontology concerns what and computation concerns how.

We classify mathematical objects into the following two kinds.

1. Mathematical objects of the first kind.
2. Mathematical objects of the second kind.

Objects of the first kind

Objects of the first kind are created by the fundamental principle of object creation:

> *Every object $\boldsymbol{a}$ is created from already created $\boldsymbol{n}$ objects $\boldsymbol{a_1, \ldots, a_n}$ $(\boldsymbol{n \geq 0})$ by applying a creation method M .*

We can visualize this *act* of creation by the following figure:

$$\frac{\boldsymbol{a_1} \quad \cdots \quad \boldsymbol{a_n}}{\boldsymbol{a}}\ M$$

or, by the equation:

$$\boldsymbol{a} = M\,(\boldsymbol{a_1, \ldots, a_n})$$

Objects of the first kind (cont.)

Equality and inequality relation on objects are defined simultaneously with the creation of objects.

Two objects:

$$M\,(a_1, \ldots, a_m) \text{ and } N\,(b_1, \ldots, b_n)$$

are equal ($=$) if and only if M and N are the same method, $m = n$ and $a_i = b_i$ $(1 \le i \le m)$.

Objects of the first kind (cont.)

Equality and inequality relation on objects are defined simultaneously with the creation of objects.

Two objects:

$$M\,(a_1, \ldots, a_m) \text{ and } N\,(b_1, \ldots, b_n)$$

are equal ($=$) if and only if M and N are the same method, $m = n$ and $a_i = b_i$ $(1 \le i \le m)$.

In other words, two objects are equal if they are created in exactly the same way, and the equality relation is decidable.

Objects of the first kind (cont.)

Equality and inequality relation on objects are defined simultaneously with the creation of objects.

Two objects:

$$M\,(a_1, \ldots, a_m) \text{ and } N\,(b_1, \ldots, b_n)$$

are equal ($=$) if and only if M and N are the same method, $m = n$ and $a_i = b_i$ ($1 \leq i \leq m$).

In other words, two objects are equal if they are created in exactly the same way, and the equality relation is decidable.

Moreover, given a creation method M and a sequence of (already created) objects, it is decidable whether M may be applied to these objects to create a new object. (Decidability of side condition.)

Objects of the first kind (cont.)

Mathematical objects of the first kind are constructed by the fundamental priciple of object creation:

- An object of the first kind is created from finitely many already created objects of the first kind.
- The creation is done by applying a creation method to existing objects.
- Both the creation method and the created object belongs to a specific class.
- The class is called the mother class of the created object.
- Thus, any object is created as an instance of its mother class.
- The equality relation $(=)$ on objects of the first kind is called the equality of the first kind.

Objects of the second kind

Let C be a class whose members are objects of the first kind, and let $=_2$ be a (partial) equivalence relation on C.

We can obtain objects of the second kind by identifying a and b in C if $a =_2 b$. When $=_2$ is a partial equivalence relation, an object a of the first kind in C is considered to be an object of the second kind if $a =_2 a$ holds.

In this setting, functions and relations on these objects must be defined so that the equality $=_2$ becomes congruence relation with respect to these functions and relations.

Well-definedness of these functions and relations are sometimes nontrivial.

Also, inductive arguments are not as smooth as for objects of the first kind, or even impossible.

Objects of the second kind (cont.)

Example: Rational numbers.

Let $\mathbb{Z}$ be the class of integers, and let $\mathbb{Z} \times \mathbb{Z}$ be the class whose members are a/b where $a, b \in \mathbb{Z}$. Define $=_2$ on $\mathbb{Z} \times \mathbb{Z}$ by:

$$a/b =_2 c/d \iff ad = bc \text{ and } b \neq 0 \text{ and } d \neq 0$$

We can define addition $(+)$ on rational numbers by putting:

$$a/b + c/d := (ad + bc)/bd.$$

This is a well-defined operation, since we have $a/b + c/d =_2 a'/b' + c'/d'$ if $a/b =_2 a'/b'$ and $c/d =_2 c'/d'$.

However, taking the denominator of a rational number is not a well-defined function on rational numbers. That is, from $1/2 =_2 2/4$, it does not follow that $2 = 4$.