Quotation, Evaluation and Metavariables

Masahiko Sato

Graduate School of Informatics
Kyoto University

Quasi-Quotation and Metavariables

Masahiko Sato

Graduate School of Informatics
Kyoto University

Quiz 1

Mathematicians study mathematics.

Logicians study ????

My Answer

Logicians study metamathematics.

Quiz 2

Mathematicians study mathematics.

Logicians study metamathematics.

Computer scientists study ????

My Answer

Computer scientists study metametamathematics.

My Observations

- Metamathematics is mathematics about mathematics.
- Metametamathematics is mathematics about metamathematics.
- Therefore, they all study mathematics!
- But, computer scientists can view mathematics from the highest point of view!!

Name and Object

A statement about an object must contain a name of the object rather than the object itself. (Quine, *Mathematical Logic*) Thus we use the name to mention the object. For example,

Name and Object

A statement about an object must contain a name of the object rather than the object itself. Thus we use the name to mention the object. For example,

> (1) Professor Igarashi is a computer scientist.

is a meaningful (and probably true!) sentence. But,

Name and Object

A statement about an object must contain a name of the object rather than the object itself. Thus we use the name to mention the object. For example,

(1) Professor Igarashi is a computer scientist.

is a meaningful (and probably true!) sentence. But,

(2) Igarashi consists of 8 characters.

is not. Instead, we should say:

Name and Object

A statement about an object must contain a name of the object rather than the object itself. Thus we use the name to mention the object. For example,

> (1) Professor Igarashi is a computer scientist.

is a meaningful (and probably true!) sentence. But,

> (2) Igarashi consists of 8 characters.

is not. Instead, we should say:

> (3) 'Igarashi' consists of 8 characters.

The sentence (1) talks about an object which is a person. So we used his name to mention him. The sentence (3) talks about an object which is an expression consisting of 8 characters. So, we used its name, which was created by enclosing the expression within quotation marks, to mention the object.

Quasi-Quotation

Quasi-quotation, introduced by Quine, is a mechanism to construct a name for an inderminate expression containing metavariables whose values are expressions. For example,

$$\ulcorner \text{Professor } \phi \text{ is a } \psi. \urcorner$$

is a quasi-quotation. Such a quasi-quotation becomes a quotation if we give appropriate values to the metavariables in it. So,

Quasi-Quotation

Quasi-quotation, introduced by Quine, is a mechanism to construct a name for an inderminate expression containing metavariables whose values are expressions. For example,

$\ulcorner$ Professor ϕ is a ψ. $\urcorner$

is a quasi-quotation. Such a quasi-quotation becomes a quotation if we give appropriate values to the metavariables in it. So, if ϕ is 'Igarashi' and ψ is 'singer', we get the quotation:

'Professor Igarashi is a singer.'

Context

Similarly,

(4) $\ulcorner\ \forall x.\forall y.\ (x+y)^2 = x^2 + \phi + y^2\ \urcorner$

is a quasi-quotation, and it can be turned into a quotation by letting ϕ to be a specific expression, say, '$2xy$':

(5) '$\forall x.\forall y.\ (x+y)^2 = x^2 + 2xy + y^2$'.

Note that, unlike substitution, the variables 'x' and 'y' in '$2xy$' gets bound after ϕ is replaced by its value in the quasi-quotation.

(4) and (5) are often written as follows and expressions like (6) are called contexts.

(6) $C \equiv \forall x.\forall y.\ (x+y)^2 = x^2 + [\ ] + y^2$,

(7) $C[2xy] \equiv \forall x.\forall y.\ (x+y)^2 = x^2 + 2xy + y^2$.

Backquote Macro

Lisp's backquote macro provides a mechanism similar to quasi-quotation.
For example,

```
(let ((X 'Igarashi) (Y 'bridge))
  `(Professor ,X plays ,Y))
```

evaluates to

```
(Professor Igarashi plays bridge).
```

It is more general than quasi-quotation, since

```
`(+ x ,(+ 2 3))
```

evaluates to

```
(+ x 5).
```

Remarks

- Expressions inside quotations should not be evaluated. Therefore,

  $2 + 3 = 5$,

  is true, but

  $`2 + 3\text{'} = `5\text{'}$.

  is false.

- There are several attempts to formalize contexts, but it seems that none of these attempts satisfy the above mentioned property of quotation.

- We will introduce $\lambda\mathbf{q}$, a simply typed lambda calculus with quotation, based on these observations. It is actually a calculus of quasi-quotation.

Types and Variables of $\lambda\mathbf{q}$

Types of $\lambda\mathbf{q}$ are:

$$A, B ::= K \mid B \Rightarrow A \mid \Box A$$

where K is a set of atomic types.

We assume that for each type A, we have an infinite set of variables $\mathbf{V}_A$ and $\mathbf{V}_A$ and $\mathbf{V}_B$ are disjoint if A and B are distinct.

We will write x^A for x if $x \in \mathbf{V}_A$. A variable is also called a *hypothesis*. A hypothesis is sometimes called a *declaration*.

Hypothesis Sequence

A *hypothesis sequence* (HS) is a sequence of hypotheses:

$$x_1^{A_1}, \ldots, x_m^{A_m}$$

where $x_1, \ldots, x_m$ are distinct variables. We use Γ, Δ etc. to denote HSs.

Typing Rules

$$\frac{x^A \in \Gamma}{\Gamma \vdash_0 x : A}\ \mathsf{var}_0 \qquad \frac{x \in \mathbf{V}_A}{\Gamma \vdash_{n+1} x : A}\ \mathsf{var}_{n+1}$$

$$\frac{\Gamma, x : B \vdash_0 a : A}{\Gamma \vdash_0 \lambda x^B[a] : B \Rightarrow A} \Rightarrow I_0 \qquad \frac{\Gamma \vdash_{n+1} a : A \quad x \in \mathbf{V}_B}{\Gamma \vdash_{n+1} \lambda x^B[a] : B \Rightarrow A} \Rightarrow I_{n+1}$$

$$\frac{\Gamma \vdash_n a : B \Rightarrow A \quad \Gamma \vdash_n b : B}{\Gamma \vdash_n a(b) : A} \Rightarrow E$$

$$\frac{\Gamma \vdash_{n+1} a : A}{\Gamma \vdash_n \ulcorner a \urcorner : \Box A}\ \Box I \qquad \frac{\Gamma \vdash_n a : \Box A}{\Gamma \vdash_{n+1} [\![a]\!] : A}\ \Box E$$

- If $\Gamma \vdash_n a : A$ is derivable, then a is said to be a *depth n $\lambda\mathbf{q}$-term under* Γ. We are mainly interested in depth 0, and we write $\Gamma \vdash a : A$ for $\Gamma \vdash_0 a : A$.
- A term is *closed* if it is a term under the empty HS.
- $\ulcorner a \urcorner$ is called the *quotation* of a.
- $[\![a]\!]$ is called a *meta-term*.

Free Variables

Given a depth n term a, the set of its free variables $\mathrm{FV}_n(a)$ is defined as follows.

$$
\begin{array}{rcl}
\mathrm{FV}_n(x) & = & \left\{ \begin{array}{ll} \{x\} & \text{if } n = 0, \\ \emptyset & \text{if } n > 0, \end{array} \right. \\
\mathrm{FV}_n(\lambda x\,[a]\,) & = & \left\{ \begin{array}{ll} \mathrm{FV}_n(a) - \{x\} & \text{if } n = 0, \\ \mathrm{FV}_n(a) & \text{if } n > 0, \end{array} \right. \\
\mathrm{FV}_n(a\,(b)\,) & = & \mathrm{FV}_n(a) \cup \mathrm{FV}_n(b), \\
\mathrm{FV}_n(\ulcorner a \urcorner) & = & \mathrm{FV}_{n+1}(a), \\
\mathrm{FV}_n(\,[\![a]\!]\,) & = & \mathrm{FV}_{n-1}(a).
\end{array}
$$

For example, we have

$$\mathrm{FV}_0(\ulcorner x\,(\,[\![y]\!]\,)\urcorner) = \mathrm{FV}_1(x\,(\,[\![y]\!]\,)\,) = \mathrm{FV}_1(x) \cup \mathrm{FV}_1(\,[\![y]\!]\,) = \mathrm{FV}_0(y) = \{y\}.$$

Substitution

If a and b are depth 0 terms and x is a variable, then the result of *substituting* x with a in b, is defined as usual. It is written as $[x := a](b)$. For example,

$$\begin{aligned} & [x := \ulcorner y + x \urcorner](\ulcorner x + [\![x]\!] \urcorner) \\ \equiv \ & \ulcorner x + [\![\ulcorner y + x \urcorner]\!] \urcorner \end{aligned}$$

Curry-Howard Isomorphism

$$\frac{\Gamma \vdash_{n+1} a : A}{\Gamma \vdash_n \ulcorner a \urcorner : \Box A}\,\Box I \qquad \frac{\Gamma \vdash_n a : \Box A}{\Gamma \vdash_{n+1} [\![a]\!] : A}\,\Box E$$

From the above typing rules, we have the following reduction of derivations.

$$\dfrac{\dfrac{\dfrac{\vdots}{\Gamma \vdash_{n+1} a : A}}{\Gamma \vdash_n \ulcorner a \urcorner : \Box A}\,\Box I}{\Gamma \vdash_{n+1} [\![\ulcorner a \urcorner]\!] : A}\,\Box E \qquad \to \qquad \dfrac{\vdots}{\Gamma \vdash_{n+1} a : A}$$

We can define reduction rules of $\boldsymbol{\lambda}\mathbf{q}$ properly by applying Curry-Howard Isomorphism.

Reduction Rules

$$\frac{\Gamma, x^B \vdash_0 a : A}{\Gamma \vdash_0 \lambda x\,[a]\,(b) \;\to\; [x := a](b) : A}\,\beta \qquad \frac{\Gamma \vdash_{n+1} a : A}{\Gamma \vdash_{n+1} [\![\ulcorner a \urcorner]\!] \to a : A}\,q$$

$$\frac{\Gamma, x^B \vdash_0 a \to c : A}{\Gamma \vdash_0 \lambda x\,[a] \;\to\; \lambda x\,[c] \,:\, B \Rightarrow A} \qquad \frac{\Gamma \vdash_{n+1} a \to c : A \quad x \in \mathbf{V}_B}{\Gamma \vdash_{n+1} \lambda x\,[a] \;\to\; \lambda x\,[c] \,:\, B \Rightarrow A}$$

$$\frac{\Gamma \vdash_n a \to c : B \Rightarrow A \quad \Gamma \vdash_n b : B}{\Gamma \vdash_n a(b) \;\to\; c(b) \,:\, A} \qquad \frac{\Gamma \vdash_n a : B \Rightarrow A \quad \Gamma \vdash_n b \to c : B}{\Gamma \vdash_n a(b) \;\to\; a(c) \,:\, A}$$

$$\frac{\Gamma \vdash_{n+1} a \to c : A}{\Gamma \vdash_n \ulcorner a \urcorner \to \ulcorner c \urcorner : \Box A} \qquad \frac{\Gamma \vdash_n a \to c : \Box A}{\Gamma \vdash_{n+1} [\![a]\!] \;\to\; [\![c]\!] \,:\, A}$$

Examples

$$
\begin{aligned}
& \lambda X\,[\ulcorner \lambda x\,[\lambda y\,[x + [\![X]\!]]\,(3)]\urcorner]\,(\ulcorner x + y\urcorner) \\
\to\; & \ulcorner \lambda x\,[\lambda y\,[x + [\![\ulcorner x + y\urcorner]\!]]\,(3)]\urcorner \\
\to\; & \ulcorner \lambda x\,[\lambda y\,[x + x + y]\,(3)]\urcorner
\end{aligned}
$$

$$
\begin{aligned}
& \lambda Y\,[\lambda X\,[\ulcorner \lambda x\,[\lambda y\,[x + [\![X]\!]]\,(3)]\urcorner]\,(\ulcorner x + [\![Y]\!]\urcorner)]\,(\ulcorner y + z\urcorner) \\
\to\; & \lambda Y\,[\ulcorner \lambda x\,[\lambda y\,[x + [\![\ulcorner x + [\![Y]\!]\urcorner]\!]]\,(3)]\urcorner]\,(\ulcorner y + z\urcorner) \\
\to\; & \ulcorner \lambda x\,[\lambda y\,[x + [\![\ulcorner x + [\![\ulcorner y + z\urcorner]\!]\urcorner]\!]]\,(3)]\urcorner \\
\to\; & \ulcorner \lambda x\,[\lambda y\,[x + [\![\ulcorner x + y + z\urcorner]\!]]\,(3)]\urcorner \\
\to\; & \ulcorner \lambda x\,[\lambda y\,[x + x + y + z]\,(3)]\urcorner
\end{aligned}
$$

Properties of $\lambda\mathbf{q}$

- Subject Reduction
- Church-Rosser
- Strong Normalizability (??)

Related Works and Conclusions

- Talcott, Mason, Dami, Hashimoto-Ohori, Sands, Sato-Sakurai-Kameyama etc. studied contexts and proposed various calculi of contexts.
- But, none of these works viewed cotexts as quasi-quotation.
- I, as a computer scientist, observed what a logician, Quine, is doing, and formalized the notions of quasi-quotation and metavariable.