Viewing λ-terms through Maps
– Essence of de Bruijn index –

Masahiko Sato

Graduate School of Informatics, Kyoto University

TPP2012
Chiba University
November 22, 2012

Authors

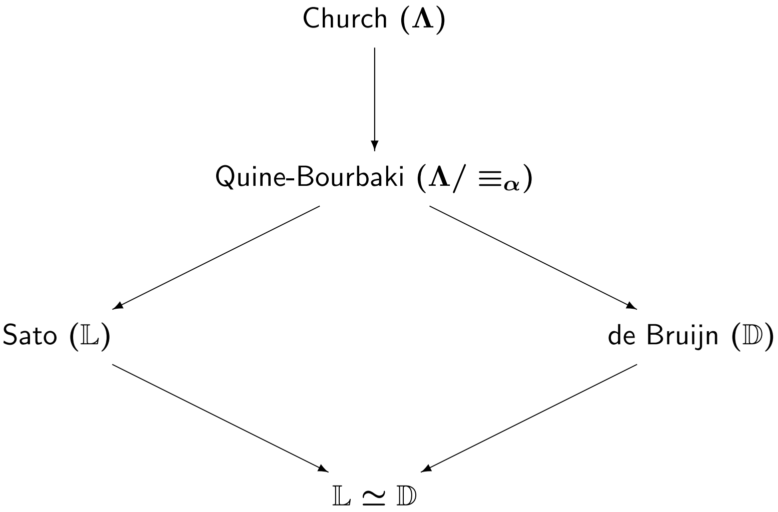
A joint work with the following people.

- Randy Pollack (Harvard University)
- Helmut Schwichtenberg (University of Munich)
- Takafumi Sakurai (Chiba University)
- James McKinna (Edinburgh University)

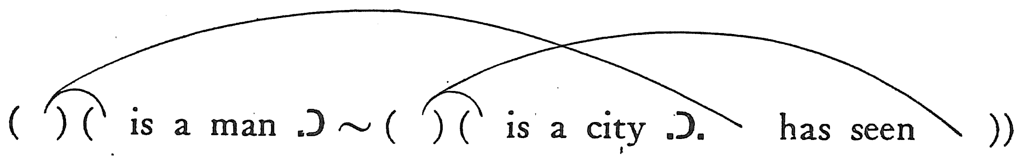
History

- 1930's. Church defined raw lambda terms ($\mathbf{\Lambda}$) and defined α-equivalence relation on them.
- 1940. Quine defined graphical representation of lambda terms. Later, in the 50's, Bourbaki rediscovered it.
- 1972. de Bruijn defined representation of lambda terms by indices ($\mathbb{D}$).
- 1980. Sato defined representation of lambda terms by map and skelton ($\mathbb{L}$).
- 2012. This talk clarifies the relationship among the above four representations.

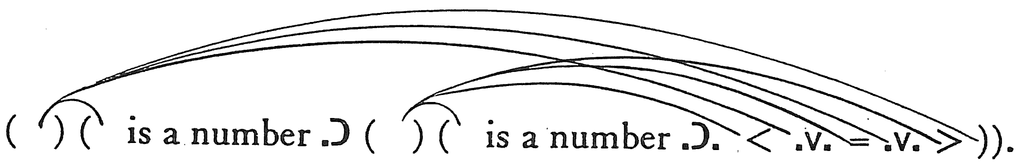
History (cont.)



sideration for established usage, the "variation" connoted belongs to a vague metaphor which is best forgotten. The variables have no meaning beyond the pronominal sort of meaning which is reflected in translations such as (20); they serve merely to indicate cross-references to various positions of quantification. Such cross-references could be made instead by curved lines or *bonds*; e.g., we might render (27) thus:



and (26) thus:



But these "quantificational diagrams" are too cumbersome to recommend themselves as a practical notation; hence the use of variables.

$$A$$

$$A'$$

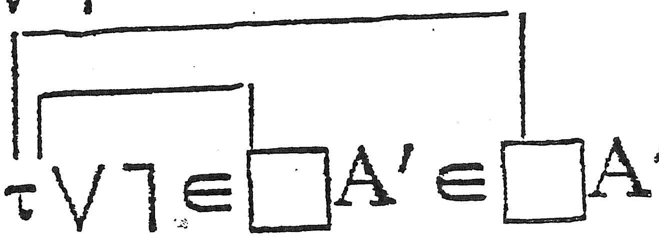
$$A''$$

$$\in AA'$$

$$\in AA''$$

$$\neg \in AA'$$

$$\vee \neg \in AA' \in AA''$$



Summary of the talk

Three datatypes

We will relate the three datatypes $(\mathbf{\Lambda}, \mathbb{L}, \mathbb{D})$ of expressions introduced by Church, S. and de Bruijn.

$\mathbf{\Lambda}$ = The datatype of raw $\boldsymbol{\lambda}$-terms.

$\mathbb{L}$ = The datatype of lambda-expressions.

$\mathbb{D}$ = The datatype of de Bruijn expressions.

Three types of binding

$\mathbf{\Lambda}$: binding by parameters $\boldsymbol{x} \in \mathbb{X}$.

$\mathbb{L}$: binding by maps $\boldsymbol{m} \in \mathbb{M}$.

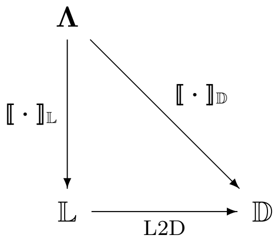
$\mathbb{D}$: binding by indices $\boldsymbol{i} \in \mathbb{I}$.

Summary of the talk (cont.)

$$
\begin{aligned}
\boldsymbol{K}, \boldsymbol{L} \in \boldsymbol{\Lambda} &::= \boldsymbol{x} \mid \boldsymbol{i} \mid \mathsf{app}(\boldsymbol{K}, \boldsymbol{L}) \mid \mathsf{lam}(\boldsymbol{x}, \boldsymbol{K}). \\
\boldsymbol{M}, \boldsymbol{N} \in \mathbb{L} &::= \boldsymbol{x} \mid \boldsymbol{i} \mid \mathsf{app}(\boldsymbol{M}, \boldsymbol{N}) \mid \mathsf{mask}(\boldsymbol{m}, \boldsymbol{M}) \quad (\boldsymbol{m} \mid \boldsymbol{M}). \\
\boldsymbol{D}, \boldsymbol{E} \in \mathbb{D} &::= \boldsymbol{x} \mid \boldsymbol{i} \mid \mathsf{app}(\boldsymbol{D}, \boldsymbol{E}) \mid \mathsf{bind}(\boldsymbol{D}). \\
\boldsymbol{x} &\in \mathbb{X}. \\
\boldsymbol{i} &\in \mathbb{I}. \\
\boldsymbol{m} &\in \mathbb{M}.
\end{aligned}
$$

The diagram

$$[\![\cdot]\!]_{\mathbb{L}} : \mathbf{\Lambda} \to \mathbb{L} = \{[\![M]\!]_{\mathbb{L}} \in \mathbb{L} \mid M \in \mathbf{\Lambda}\} \qquad \text{surjection.}$$
$$[\![\cdot]\!]_{\mathbb{D}} : \mathbf{\Lambda} \to \mathbb{D} = \{[\![M]\!]_{\mathbb{D}} \in \mathbb{D} \mid M \in \mathbf{\Lambda}\} \qquad \text{surjection.}$$
$$\mathrm{L2D} : \mathbb{L} \to \mathbb{D} \qquad \text{bijection.}$$



The Datatype $\mathbb{M}$ of Maps

Intuitive idea

$$\frac{}{0 \in \mathbb{M}} \qquad \frac{}{1 \in \mathbb{M}} \qquad \frac{\boldsymbol{m} \in \mathbb{M} \quad \boldsymbol{n} \in \mathbb{M}}{\mathsf{mapp}(\boldsymbol{m}, \boldsymbol{n}) \in \mathbb{M}}$$

Equality axiom on $\mathbb{M}$

$$\mathsf{mapp}(0, 0) = 0.$$

The Datatype $\mathbb{M}$ of Maps (cont.)

$$\frac{}{1 \in \mathbb{M}_{\mathbf{0}}}\ \text{mone}$$

$$\frac{\boldsymbol{m} \in \mathbb{M}_{\mathbf{0}}}{\mathsf{minl}(\boldsymbol{m}) \in \mathbb{M}_{\mathbf{0}}}\ \text{minl} \qquad \frac{\boldsymbol{n} \in \mathbb{M}_{\mathbf{0}}}{\mathsf{minr}(\boldsymbol{n}) \in \mathbb{M}_{\mathbf{0}}}\ \text{minr}$$

$$\frac{\boldsymbol{m} \in \mathbb{M}_{\mathbf{0}} \quad \boldsymbol{n} \in \mathbb{M}_{\mathbf{0}}}{\mathsf{mcons}(\boldsymbol{m}, \boldsymbol{n}) \in \mathbb{M}_{\mathbf{0}}}\ \text{mcons}$$

$$\frac{}{0 \in \mathbb{M}}\ \text{mzero} \qquad \frac{\boldsymbol{m} \in \mathbb{M}_{\mathbf{0}}}{\boldsymbol{m} \in \mathbb{M}}\ \text{mincl}$$

The Datatype $\mathbb{M}$ of Maps (cont.)

We define mapp : $\mathbb{M} \times \mathbb{M} \to \mathbb{M}$ as follows.

$$\text{mapp}(m, n) := \begin{cases} 0 & \text{if } m = n = 0, \\ \text{minl}(m) & \text{if } m \neq 0 \text{ and } n = 0, \\ \text{minr}(n) & \text{if } m = 0 \text{ and } n \neq 0, \\ \text{mcons}(m, n) & \text{if } m \neq 0 \text{ and } n \neq 0. \end{cases}$$

We will write $(m\ n)$ or mn for mapp(m, n).

Orthogonality relation

$$\frac{}{m \perp 0} \qquad \frac{}{0 \perp n} \qquad \frac{m \perp n \quad m' \perp n'}{mm' \perp nn'}$$

Example: $(\mathbf{1}\ \mathbf{0}) \perp (\mathbf{0}\ \mathbf{1})$ but not $(\mathbf{1}\ \mathbf{1}) \perp (\mathbf{0}\ \mathbf{1})$.

The Datatype $\mathbb{I}$ of Indices

$$\frac{}{\mathsf{box} \in \mathbb{I}}\ \mathsf{box} \qquad \frac{\boldsymbol{i} \in \mathbb{I}}{\mathsf{lift}(\boldsymbol{i}) \in \mathbb{I}}\ \mathsf{lift}$$

$$\boldsymbol{i}, \boldsymbol{j} \in \mathbb{I} ::= \mathsf{box} \mid \mathsf{lift}(\boldsymbol{i}).$$

We will write $\square$ for box.

The Datatype $\mathbb{X}$ of Parameters

We assume a countably infinite set $\mathbb{X}$ of parameters.
We will write $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ for parameters.
We assume that equality relation on $\mathbb{X}$ is decidable.

The Datatype Λ of Raw λ-terms

$$\frac{}{\boldsymbol{x} \in \boldsymbol{\Lambda}}\ \text{par} \qquad \frac{}{\boldsymbol{i} \in \boldsymbol{\Lambda}}\ \text{idx}$$

$$\frac{\boldsymbol{K} \in \boldsymbol{\Lambda} \quad \boldsymbol{L} \in \boldsymbol{\Lambda}}{\mathsf{app}(\boldsymbol{K}, \boldsymbol{L}) \in \boldsymbol{\Lambda}}\ \mathsf{app} \qquad \frac{\boldsymbol{x} \in \mathbb{X} \quad \boldsymbol{K} \in \boldsymbol{\Lambda}}{\mathsf{lam}(\boldsymbol{x}, \boldsymbol{K}) \in \boldsymbol{\Lambda}}\ \mathsf{lam}$$

$$\boldsymbol{K}, \boldsymbol{L} \in \boldsymbol{\Lambda} ::= \boldsymbol{x} \mid \boldsymbol{i} \mid \mathsf{app}(\boldsymbol{K}, \boldsymbol{L}) \mid \mathsf{lam}(\boldsymbol{x}, \boldsymbol{K}).$$

$$\boldsymbol{x} \in \mathbb{X}.$$

$$\boldsymbol{i} \in \mathbb{I}.$$

Remark. lam binds parameter $\boldsymbol{x}$ in $\boldsymbol{K}$.

The Datatype $\mathbb{L}$ of lambda-expressions

$$\frac{}{\boldsymbol{x} \in \mathbb{L}}\ \text{par} \qquad \frac{}{\boldsymbol{i} \in \mathbb{L}}\ \text{idx}$$

$$\frac{\boldsymbol{M} \in \mathbb{L} \quad \boldsymbol{N} \in \mathbb{L}}{\mathsf{app}(\boldsymbol{M}, \boldsymbol{N}) \in \mathbb{L}}\ \mathsf{app} \qquad \frac{\boldsymbol{m} \in \mathbb{M} \quad \boldsymbol{M} \in \mathbb{L} \quad \boldsymbol{m} \mid \boldsymbol{M}}{\mathsf{mask}(\boldsymbol{m}, \boldsymbol{M}) \in \mathbb{L}}$$

$$\frac{\boldsymbol{M} \in \mathbb{L}}{0 \mid \boldsymbol{M}} \qquad \frac{}{1 \mid \mathsf{box}}$$

$$\frac{\boldsymbol{m} \mid \boldsymbol{M} \quad \boldsymbol{n} \mid \boldsymbol{N}}{\mathsf{mapp}(\boldsymbol{m}, \boldsymbol{n}) \mid \mathsf{app}(\boldsymbol{M}, \boldsymbol{N})} \qquad \frac{\boldsymbol{m} \mid \boldsymbol{N} \quad \boldsymbol{n} \mid \boldsymbol{N} \quad \boldsymbol{m} \perp \boldsymbol{n}}{\boldsymbol{m} \mid \mathsf{mask}(\boldsymbol{n}, \boldsymbol{N})}$$

The Datatype $\mathbb{L}$ of lambda-expressions (cont.)

Notational Convention

- We use M, N, P as metavariables ranging over lambda-expressions.
- We write $(M\ N)$ and also MN for app(M, N).
- We write $[m\backslash M]$ and also $m\backslash M$ for mask(m, M).
- A lambda-expression of the form mask(m, M) is called an abstract.
- We use A, B as metavariables ranging over abstarcts, and write $\mathbb{A}$ for the subset of $\mathbb{L}$ consisting of all the abstracts.

Map and Skelton

We define $\mathrm{map} : \mathbb{X} \times \mathbb{L} \to \mathbb{M}$ and $\mathrm{skel} : \mathbb{X} \times \mathbb{L} \to \mathbb{L}$. We write $\boldsymbol{M_X}$ for $\mathrm{map}(\boldsymbol{X}, \boldsymbol{M})$, and $\boldsymbol{M^X}$ for $\mathrm{skel}(\boldsymbol{X}, \boldsymbol{M})$.

$$
\begin{aligned}
y_x &:= \begin{cases} 1 & \text{if } x = y, \\ 0 & \text{if } x \neq y. \end{cases} \\
i_x &:= 0. \\
(M\ N)_x &:= \mathsf{mapp}(M_x, N_x). \\
[m \backslash M]_x &:= M_x.
\end{aligned}
$$

$$
\begin{aligned}
y^x &:= \begin{cases} \Box & \text{if } x = y, \\ y & \text{if } x \neq y. \end{cases} \\
i^x &:= i. \\
(M\ N)^x &:= (M^x\ N^x). \\
[m \backslash M]^x &:= [m \backslash M^x].
\end{aligned}
$$

Lambda Abstraction in $\mathbb{L}$

We define lam $: \mathbb{X} \times \mathbb{L} \to \mathbb{L}$ by:

$$\mathsf{lam}(x, M) := [M_x \backslash M^x].$$

Examples. We assume that x, y and z are distinct parameters.

$$\begin{aligned}
\mathsf{lam}(x, x) &= 1\backslash\square. \\
\mathsf{lam}(x, y) &= 0\backslash y. \\
\mathsf{lam}(x, \mathsf{lam}(y, x)) &= \mathsf{lam}(x, 0\backslash x) \\
&= 1\backslash 0\backslash\square. \\
\mathsf{lam}(x, \mathsf{lam}(y, y)) &= \mathsf{lam}(x, \mathsf{lam}(1, \square)) \\
&= 0\backslash 1\backslash\square. \\
\mathsf{lam}(x, \mathsf{lam}(y, \mathsf{lam}(z, (xz\ yz)))) &= \\
&(10\ 00)\backslash(00\ 10)\backslash(01\ 01)\backslash(\square\square\ \square\square)
\end{aligned}$$

Instantiation and Substitution

We define the instantiation operation: inst $: \mathbb{A} \times \mathbb{L} \to \mathbb{L}$ as follows. We will write $A \blacktriangledown M$ for inst(A, M).

$$\begin{aligned}
[1\backslash\Box]\blacktriangledown P &:= P. \\
[0\backslash M]\blacktriangledown P &:= M. \\
[(m\ n)\backslash(M\ N)]\blacktriangledown P &:= ([m\backslash M]\blacktriangledown P\ [n\backslash N]\blacktriangledown P). \\
[m\backslash[n\backslash N]]\blacktriangledown P &:= [n\backslash[m\backslash N]\blacktriangledown P].
\end{aligned}$$

We can now define substitution operation:
subst $: \mathbb{L} \times \mathbb{X} \times \mathbb{L} \to \mathbb{L}$ as follows.

$$M\{x\backslash N\} := \mathsf{lam}(x, M)\blacktriangledown N.$$

Instantiation and Substitution (cont.)

Example.

$$
\begin{aligned}
\mathsf{lam}(y, yx)\{x \backslash y\} &= \mathsf{lam}(x, \mathsf{lam}(y, yx)) \blacktriangledown y \\
&= \mathsf{lam}(x, [10 \backslash \Box x]) \blacktriangledown y \\
&= [01 \backslash [10 \backslash \Box\Box]] \blacktriangledown y \\
&= 10 \backslash [01 \backslash \Box\Box] \blacktriangledown y \\
&= 10 \backslash ([0 \backslash \Box] \blacktriangledown y \ [1 \backslash \Box] \blacktriangledown y) \\
&= 10 \backslash \Box y \\
&= \mathsf{lam}(z, zy)
\end{aligned}
$$

Remark. By internalizing the instantiation operation, we can easily get an explicit instantiation calculus.

Instantiation and Substitution (cont.)

Substitution Lemma

If $x \neq y$ and $x \notin \mathsf{FP}(P)$, then

$$M\{x\backslash N\}\{y\backslash P\} = M\{y\backslash P\}\{x\backslash N\{y\backslash P\}\}.$$

Proof. By induction on $M \in \mathbb{L}$. Here, we only treat the case where $M = \mathsf{mcons}(m_1, m_2)\backslash M_1 M_2 = m_1 m_2\backslash M_1 M_2$.

$$\begin{aligned}
& M\{x\backslash N\}\{y\backslash P\} \\
&= [m_1 m_2\backslash M_1 M_2]\{x\backslash N\}\{y\backslash P\} \\
&= [m_1 m_2\backslash (M_1\{x\backslash N\}\ M_2\{x\backslash N\})]\{y\backslash P\} \\
&= [m_1 m_2\backslash (M_1\{x\backslash N\}\{y\backslash P\}\ M_2\{x\backslash N\}\{y\backslash P\})] \\
&= [(m_1\ m_2)\backslash (M_1\{y\backslash P\}\{x\backslash N\{y\backslash P\}\}\ M_2\{y\backslash P\}\{x\backslash N\{y\backslash P\} \\
&\qquad \text{(by IH)} \\
&= [(m_1\ m_2)\backslash (M_1\{y\backslash P\}\ M_2\{y\backslash P\})]\{x\backslash N\{y\backslash P\}\} \\
&= [(m_1\ m_2)\backslash (M_1\ M_2)]\{y\backslash P\}\{x\backslash N\{y\backslash P\}\} \\
&= M\{y\backslash P\}\{x\backslash N\{y\backslash P\}\}.
\end{aligned}$$

The $\mathbb{L}_\beta$-calculus

$$\frac{}{AM \to_\beta A\blacktriangledown M}\ \beta$$

$$\frac{M \to_\beta M'}{MN \to_\beta M'N}\ \mathsf{appl} \qquad \frac{M \in \mathbb{L}}{MN \to_\beta MN'}\ \mathsf{appr}$$

$$\frac{M \to_\beta N}{\mathsf{lam}(x, M) \to_\beta \mathsf{lam}(x, N)}\ \xi$$

Remark. Traditional way of formulating β-conversion rule is:

$$(\mathsf{lam}(x, M)\ N) \to_\beta M\{x\backslash N\}.$$

The $\mathbb{L}_{\beta\eta}$-calculus

The $\mathbb{L}_{\beta\eta}$-calculus is obtained from the $\mathbb{L}_{\beta}$-calculus by adding the following η-rule.

$$\frac{}{[01\backslash M\Box] \to_{\beta\eta} M}\ \eta$$

Remark. In the tradtional $\lambda_{\beta\eta}$-caclulus the η-rule is:

$$\mathsf{lam}(x, Mx) \to_{\beta\eta} M \quad \text{if } x \notin \mathsf{FP}(M).$$

But, we could state it without mentioning x and hence without mentioning the side condition on x and M, since we have

$$\mathsf{lam}(x, Mx) = [01\backslash M\Box] \quad \text{if } x \notin \mathsf{FP}(M).$$

In fact, the η-rule is a rule about abstracts M and has nothing to do with parameters x. The same remark applies to the β-conversion rule as well.

Interpretation of Λ in $\mathbb{L}$

We define the interpretation function $[\![-]\!]_{\mathbb{L}} : \Lambda \to \mathbb{L}_{\Lambda}$ as follows.

$$
\begin{aligned}
[\![x]\!]_{\mathbb{L}} &:= X. \\
[\![i]\!]_{\mathbb{L}} &:= i. \\
[\![MN]\!]_{\mathbb{L}} &:= ([\![M]\!]_{\mathbb{L}}\ [\![N]\!]_{\mathbb{L}}). \\
[\![\mathsf{lam}(x, M)]\!]_{\mathbb{L}} &:= \mathsf{lam}(x, [\![M]\!]_{\mathbb{L}}).
\end{aligned}
$$

Remark. Two raw λ-terms M and N are α-equivalent iff $[\![M]\!]_{\mathbb{L}} = [\![N]\!]_{\mathbb{L}}$.

The Datatype $\mathbb{D}$ of de Bruijn-expressions

$$\frac{}{\boldsymbol{x} \in \mathbb{D}}\ \text{par} \qquad \frac{}{\boldsymbol{i} \in \mathbb{D}}\ \text{idx}$$

$$\frac{\boldsymbol{D} \in \mathbb{D} \quad \boldsymbol{E} \in \mathbb{D}}{\mathsf{app}(\boldsymbol{D}, \boldsymbol{E}) \in \mathbb{D}}\ \mathsf{app} \qquad \frac{\boldsymbol{D} \in \mathbb{D}}{\mathsf{bind}(\boldsymbol{D}) \in \mathbb{D}}\ \mathsf{bind}$$

$$\boldsymbol{D}, \boldsymbol{E} \in \mathbb{D} ::= \boldsymbol{x} \mid \boldsymbol{i} \mid \mathsf{app}(\boldsymbol{D}, \boldsymbol{E}) \mid \mathsf{bind}(\boldsymbol{D}).$$

$$\boldsymbol{x} \in \mathbb{X}.$$

$$\boldsymbol{i} \in \mathbb{I}.$$

Summary of the Datatypes Λ, $\mathbb{L}$ and $\mathbb{D}$

$$
\begin{aligned}
K, L \in \Lambda &::= x \mid i \mid \mathsf{app}(K, L) \mid \mathsf{lam}(x, K). \\
M, N \in \mathbb{L} &::= x \mid i \mid \mathsf{app}(M, N) \mid \mathsf{mask}(m, M) \quad (m \mid M). \\
D, E \in \mathbb{D} &::= x \mid i \mid \mathsf{app}(D, E) \mid \mathsf{bind}(D). \\
x &\in \mathbb{X}. \\
i &\in \mathbb{I}. \\
m &\in \mathbb{M}.
\end{aligned}
$$

Interpretation of $\mathbb{L}$ in $\mathbb{D}$

We define the mask function

$$\mathsf{mask}_i : \mathbb{M} \times \mathbb{D} \to \mathbb{D} \ (i \in \mathbb{I})$$

as follows. We will write $[m \backslash D]$ or $m \backslash D$ for $\mathsf{mask}(m, D)$.

$$
\begin{aligned}
0 \backslash_i x &:= \mathsf{bind}(x). \\
0 \backslash_i j &:= \begin{cases} \mathsf{bind}(j) & \text{if } j < i, \\ \mathsf{bind}(j+1) & \text{if } j \geq i. \end{cases} \\
1 \backslash_i i &:= \mathsf{bind}(i). \\
mn \backslash_i DE &:= \mathsf{bind}(D'E') \\
& \qquad \text{if } m \backslash_i D = \mathsf{bind}(D') \text{ and } n \backslash_i E = \mathsf{bind}(E'). \\
m \backslash_i \mathsf{bind}(D) &:= \mathsf{bind}(m \backslash_{i+1} D).
\end{aligned}
$$

Interpretation of $\mathbb{L}$ in $\mathbb{D}$ (cont.)

We define the interpretation function $\mathsf{L2D} : \mathbb{L} \to \mathbb{D}$ as follows.

$$\begin{aligned}
\mathsf{L2D}(x) &:= x. \\
\mathsf{L2D}(i) &:= i. \\
\mathsf{L2D}(MN) &:= (\mathsf{L2D}(M)\ \mathsf{L2D}(N)). \\
\mathsf{L2D}(m\backslash M) &:= [m\backslash \mathsf{L2D}(M)].
\end{aligned}$$