$\{\rightarrow, \multimap\}$ is Full in $\{!, \multimap\}$*

Masahito Hasegawa
RIMS, Kyoto University

21 January 2002

Abstract

We show that the $\{\rightarrow, \multimap\}$-fragment of Intuitionistic Linear Logic is *full* in the $\{!, \multimap\}$-fragment, both formulated as linear lambda calculi. The proof is a mild extension of our previous technique used for showing the fullness of Girard's translation from Intuitionistic Logic into Intuitionistic Linear Logic, and makes use of double-parameterized logical predicates.

1 Linear Lambda Calculi

Our presentation is based on a dual-context type system for intuitionistic linear logic (called DILL) due to Barber and Plotkin [1]. In this formulation of the linear lambda calculi, a typing judgement takes the form $\Gamma\ ;\ \Delta \vdash M : \tau$ in which Γ represents an intuitionistic (or additive) context whereas Δ is a linear (multiplicative) context.

A set of *base types* (b ranges over them) is fixed throughout this paper.

1.1 $\boldsymbol{\lambda}^{\rightarrow,\multimap}$

The first system we consider is a fragment with connectives $\rightarrow$ (intuitionistic or non-linear arrow type) and $\multimap$ (linear arrow type). We use $\boldsymbol{\lambda} x^\sigma.M$ and $M \,@\, N$ for the non-linear lambda abstraction and application respectively, while $\lambda x^\sigma.M$ and $M\,N$ for the linear ones.

Types and Terms

$$\begin{array}{rcl} \sigma & ::= & b \mid \sigma \rightarrow \sigma \mid \sigma \multimap \sigma \\ M & ::= & x \mid \boldsymbol{\lambda} x^\sigma.M \mid M \,@\, M \mid \lambda x^\sigma.M \mid M\,M \end{array}$$

*The content of this paper is a very mild extenstion of our previous work presented in [2] – the structure and most of the details of the proof are essentially the same (or even identical). The reason of this reworking is rather technical; we need the main result of this paper in our another work [3].

Typing

$$\frac{}{\Gamma\ ;\ x:\sigma \vdash x:\sigma} \qquad \frac{}{\Gamma_1, x:\sigma, \Gamma_2\ ;\ \emptyset \vdash x:\sigma}$$

$$\frac{\Gamma, x:\sigma_1\ ;\ \Delta \vdash M:\sigma_2}{\Gamma\ ;\ \Delta \vdash \boldsymbol{\lambda} x^{\sigma_1}.M : \sigma_1 \to \sigma_2} \qquad \frac{\Gamma\ ;\ \Delta \vdash M:\sigma_1 \to \sigma_2 \quad \Gamma\ ;\ \emptyset \vdash N:\sigma_1}{\Gamma\ ;\ \Delta \vdash M\,@\,N:\sigma_2}$$

$$\frac{\Gamma\ ;\ \Delta, x:\sigma_1 \vdash M:\sigma_2}{\Gamma\ ;\ \Delta \vdash \lambda x^{\sigma_1}.M : \sigma_1 \multimap \sigma_2} \qquad \frac{\Gamma\ ;\ \Delta_1 \vdash M:\sigma_1 \multimap \sigma_2 \quad \Gamma\ ;\ \Delta_2 \vdash N:\sigma_1}{\Gamma\ ;\ \Delta_1 \sharp \Delta_2 \vdash MN:\sigma_2}$$

where $\Delta_1 \sharp \Delta_2$ is a merge of Δ_1 and Δ_2 [1]. Thus, $\Delta_1 \sharp \Delta_2$ represents one of possible merges of Δ_1 and Δ_2 as finite lists. We assume that, when we introduce $\Delta_1 \sharp \Delta_2$, there is no variable occurring both in Δ_1 and in Δ_2. We write $\emptyset$ for the empty context. We note that any typing judgement has a unique derivation (hence a typing judgement can be identified with its derivation).

Axioms

$$\begin{array}{lcl} (\lambda x.M)\,N & = & M[N/x] \\ \lambda x.M\,x & = & M \\ (\boldsymbol{\lambda} x.M)\,N & = & M[N/x] \\ \boldsymbol{\lambda} x.M\,@\,x & = & M \quad (x \notin FV(M)) \end{array}$$

The equality judgement $\Gamma\ ;\ \Delta \vdash M = N : \sigma$ is defined as usual.

1.2 $\boldsymbol{\lambda}^{!,\multimap}$

The second fragment is that with the exponential ! and the linear arrow type $\multimap$. The same system has been used in [2] as the target calculus of Girard's translation.

Types and Terms

$$\begin{array}{lcl} \tau & ::= & b \mid !\tau \mid \tau \multimap \tau \\ M & ::= & x \mid !M \mid \mathsf{let}\ !x^{\tau}\ \mathsf{be}\ M\ \mathsf{in}\ M \mid \lambda x^{\tau}.M \mid M\,M \end{array}$$

Typing

$$\frac{}{\Gamma\ ;\ x:\tau \vdash x:\tau} \qquad \frac{}{\Gamma_1, x:\tau, \Gamma_2\ ;\ \emptyset \vdash x:\tau}$$

$$\frac{\Gamma\ ;\ \emptyset \vdash M:\tau}{\Gamma\ ;\ \emptyset \vdash !M : !\tau} \qquad \frac{\Gamma\ ;\ \Delta_1 \vdash M : !\tau_1 \quad \Gamma, x:\tau_1\ ;\ \Delta_2 \vdash N:\tau_2}{\Gamma\ ;\ \Delta_1 \sharp \Delta_2 \vdash \mathsf{let}\ !x^{\tau_1}\ \mathsf{be}\ M\ \mathsf{in}\ N : \tau_2}$$

$$\frac{\Gamma\ ;\ \Delta, x:\tau_1 \vdash M:\tau_2}{\Gamma\ ;\ \Delta \vdash \lambda x^{\tau_1}.M : \tau_1 \multimap \tau_2} \qquad \frac{\Gamma\ ;\ \Delta_1 \vdash M:\tau_1 \multimap \tau_2 \quad \Gamma\ ;\ \Delta_2 \vdash N:\tau_1}{\Gamma\ ;\ \Delta_1 \sharp \Delta_2 \vdash MN:\tau_2}$$

Axioms

$$\begin{array}{lcl}
(\lambda x.M)N & = & M[N/x] \\
\lambda x.Mx & = & M \\
\mathsf{let}\ !x\ \mathsf{be}\ !M\ \mathsf{in}\ N & = & N[M/x] \\
\mathsf{let}\ !x\ \mathsf{be}\ M\ \mathsf{in}\ !x & = & M \\
C[\mathsf{let}\ !x\ \mathsf{be}\ M\ \mathsf{in}\ N] & = & \mathsf{let}\ !x\ \mathsf{be}\ M\ \mathsf{in}\ C[N]
\end{array}$$

where $C[-]$ is a linear context (no ! binds $[-]$); formally it is generated from the following grammar.

$$C\ ::=\ [-]\ |\ \lambda x.C\ |\ CM\ |\ MC\ |\ \mathsf{let}\ !x\ \mathsf{be}\ C\ \mathsf{in}\ M\ |\ \mathsf{let}\ !x\ \mathsf{be}\ M\ \mathsf{in}\ C$$

1.3 Translation from $\boldsymbol{\lambda}^{\rightarrow,\multimap}$ to $\boldsymbol{\lambda}^{!,\multimap}$

We now give the translation $(-)^\circ$ from $\lambda^{\rightarrow,\multimap}$ to $\lambda^{!,\multimap}$, which is a straightforward extension of Girard's translation from the simply typed lambda calculus into $\lambda^{!,\multimap}$.

$$\begin{array}{rcl}
b^\circ & \equiv & b \\
(\sigma_1 \rightarrow \sigma_2)^\circ & \equiv & !\sigma_1^\circ \multimap \sigma_2^\circ
\end{array}$$

$$\begin{array}{rcl}
x^\circ & \equiv & x \\
(\lambda x^\sigma.M)^\circ & \equiv & \lambda x^{\sigma^\circ}.M^\circ \\
(M^{\sigma_1\multimap\sigma_2}\,N^{\sigma_1})^\circ & \equiv & M^\circ\,N^\circ \\
(\boldsymbol{\lambda} x^\sigma.M)^\circ & \equiv & \lambda y^{!\sigma^\circ}.\mathsf{let}\ !x^{\sigma^\circ}\ \mathsf{be}\ y\ \mathsf{in}\ M^\circ \\
(M^{\sigma_1\rightarrow\sigma_2}\,@\,N^{\sigma_1})^\circ & \equiv & M^\circ\,(!N^\circ)
\end{array}$$

Proposition 1.1 (type soundness) *If* $\Gamma\ ;\ \Delta \vdash M : \sigma$ *is derivable in* $\lambda^{\rightarrow,\multimap}$*, so is* $\Gamma^\circ\ ;\ \Delta^\circ \vdash M^\circ : \sigma^\circ$ *in* $\lambda^{!,\multimap}$.

Proof: For instance, the derivation

$$\frac{\begin{array}{c}\vdots\\ \Gamma, x:\sigma_1\ ;\ \Delta \vdash M : \sigma_2\end{array}}{\Gamma\ ;\ \Delta \vdash \boldsymbol{\lambda} x^{\sigma_1}.M : \sigma_1 \rightarrow \sigma_2}$$

is sent to

$$\frac{\dfrac{\overline{\Gamma^\circ\ ;\ y:!\sigma_1^\circ \vdash y:!\sigma_1^\circ} \quad \begin{array}{c}\vdots\\ \Gamma^\circ, x:\sigma_1^\circ\ ;\ \Delta^\circ \vdash M^\circ : \sigma_2^\circ\end{array}}{\Gamma^\circ\ ;\ \Delta^\circ, y:!\sigma_1^\circ \vdash \mathsf{let}\ !x^{\sigma_1^\circ}\ \mathsf{be}\ y\ \mathsf{in}\ M^\circ : \sigma_2^\circ}}{\Gamma^\circ\ ;\ \Delta^\circ \vdash (\boldsymbol{\lambda} x^{\sigma_1}.M)^\circ \equiv \lambda y^{!\sigma_1^\circ}.\mathsf{let}\ !x^{\sigma_1^\circ}\ \mathsf{be}\ y\ \mathsf{in}\ M^\circ :!\sigma_1^\circ \rightarrow \sigma_2^\circ}$$

while

$$\frac{\begin{array}{c}\vdots\\ \Gamma\ ;\ \Delta \vdash M : \sigma_1 \rightarrow \sigma_2\end{array} \quad \begin{array}{c}\vdots\\ \Gamma\ ;\ \emptyset \vdash N : \sigma_1\end{array}}{\Gamma\ ;\ \Delta \vdash M\,@\,N : \sigma_2}$$

is sent to

$$\dfrac{\dfrac{\vdots}{\Gamma^\circ \ ; \ \Delta^\circ \vdash M^\circ :!\sigma_1^\circ \multimap \sigma_2^\circ} \quad \dfrac{\dfrac{\vdots}{\Gamma^\circ \ ; \ \emptyset \vdash N^\circ : \sigma_1^\circ}}{\Gamma^\circ \ ; \ \emptyset \vdash !N^\circ :!\sigma_1^\circ}}{\Gamma^\circ \ ; \ \Delta^\circ \vdash (M \,@\, N)^\circ \equiv M^\circ \,(!N^\circ) : \sigma_2^\circ}$$

Lemma 1.2 (compatibility with substitution) $M^\circ[N^\circ/x] = (M[N/x])^\circ$.

Proof: Easy. □

Proposition 1.3 (equational soundness) *If* $\Gamma \ ; \ \Delta \vdash M = N : \sigma$ *holds in* $\lambda^{\rightarrow,\multimap}$*, so does* $\Gamma^\circ \ ; \ \Delta^\circ \vdash M^\circ = N^\circ : \sigma^\circ$ *in* $\lambda^{!,\multimap}$.

Proof: The crucial cases are:

$$\begin{array}{rcl} ((\boldsymbol{\lambda} x.M) \,@\, N)^\circ & \equiv & (\lambda y.\mathsf{let}\ !x\ \mathsf{be}\ y\ \mathsf{in}\ M^\circ)\,(!N^\circ) \\ & = & \mathsf{let}\ !x\ \mathsf{be}\ !N^\circ\ \mathsf{in}\ M^\circ \\ & = & M^\circ[N^\circ/x] \\ & = & (M[N/x])^\circ \end{array}$$

$$\begin{array}{rcl} (\boldsymbol{\lambda} x.M \,@\, x)^\circ & \equiv & \lambda y.\mathsf{let}\ !x\ \mathsf{be}\ y\ \mathsf{in}\ M^\circ\,(!x) \\ & = & \lambda y.M^\circ\,(\mathsf{let}\ !x\ \mathsf{be}\ y\ \mathsf{in}\ !x) \\ & = & \lambda y.M^\circ\, y \\ & = & M \end{array}$$

□

Proposition 1.4 (equational completeness) *If* $\Gamma^\circ \ ; \ \Delta^\circ \vdash M^\circ = N^\circ : \sigma^\circ$ *holds in* $\lambda^{!,\multimap}$*, then* $\Gamma \ ; \ \Delta \vdash M = N : \sigma$ *holds in* $\lambda^{\rightarrow,\multimap}$.

Proof: There is an obvious translation $(-)^\bullet$ from $\lambda^{!,\multimap}$ to the simply typed lambda calculus $\lambda^{\rightarrow}$ which forgets all the information regarding the linearlity (see [2]). Since the composition $\lambda^{\rightarrow,\multimap} \xrightarrow{(-)^\circ} \lambda^{!,\multimap} \xrightarrow{(-)^\bullet} \lambda^{\rightarrow}$ obviously reflects the equality, so does $(-)^\circ$. □

2 Dual Logical Predicates

In addition to the case of Girard's translation from the simply typed lambda calculus to $\lambda^{!,\multimap}$, we have to deal with the linearity in the source calculus as well. To this end we add one more parameter for linear contexts to the logical predicates used in [2] for showing the fullness of Girard's translation.

Notations We write $\Lambda^{\rightarrow,\multimap}$ and $\Lambda^{!,\multimap}$ for the sets of well-typed terms of $\lambda^{\rightarrow,\multimap}$ and $\lambda^{!,\multimap}$ respectively, and use the following notations.

$$\begin{array}{rcl} \Lambda_\sigma^{\rightarrow,\multimap}(\Gamma \ ; \ \Delta) & = & \{M \in \Lambda^{\rightarrow,\multimap} \mid \Gamma \ ; \Delta \vdash M : \sigma\} \\ \Lambda_\tau^{!,\multimap}(\Gamma \ ; \ \Delta) & = & \{M \in \Lambda^{!,\multimap} \mid \Gamma \ ; \Delta \vdash M : \tau\} \\ \Gamma_1 \leq \Gamma_2 & \text{iff} & \Gamma_2 = \Gamma, \Gamma_1 \text{ for some } \Gamma \end{array}$$

Definition 2.1 (dual predicate) *Let τ be a type of $\lambda^{!,\multimap}$. A family P of sets indexed by the contexts of $\lambda^{\rightarrow,\multimap}$ is called a* dual predicate on τ *when* $P(\Gamma\ ;\ \Delta) \subseteq \Lambda^{!,\multimap}_{\tau}(\Gamma^\circ\ ;\ \Delta^\circ)$ *and closed under*

- *renaming: $M \in P(\Gamma\ ;\ \Delta)$ implies $M\rho \in P(\Gamma\rho\ ;\ \Delta\rho)$ for any renaming ρ*
- *weakening: $\Gamma \leq \Gamma'$ implies $P(\Gamma\ ;\ \Delta) \subseteq P(\Gamma'\ ;\ \Delta)$*
- *equality: $M \in P(\Gamma\ ;\ \Delta)$ and $M = M'$ imply $M' \in P(\Gamma\ ;\ \Delta)$*

Lemma 2.2 (linear implication on dual predicates) *Let P_1, P_2 be dual predicates on τ_1 and τ_2 respectively. Then there is a dual predicate $P_1 \multimap P_2$ on $\tau_1 \multimap \tau_2$ defined by*

$$(P_1 \multimap P_2)(\Gamma\ ;\ \Delta) = \left\{ M \in \Lambda^{!,\multimap}_{\tau_1\multimap\tau_2}(\Gamma^\circ\ ;\ \Delta^\circ) \;\middle|\; \begin{array}{l} N \in P_1(\Gamma'\ ;\ \Delta') \text{ and } \Gamma \leq \Gamma' \text{ imply} \\ M\,N \in P_2(\Gamma'\ ;\ \Delta\sharp\Delta') \end{array} \right\}$$

Proof: Easy. □

Lemma 2.3 (exponential on dual predicates) *Let P be a dual predicate on τ. Then there is a dual predicate $!P$ on $!\tau$ defined by*

$$!P(\Gamma\ ;\ \Delta) = \begin{cases} \left\{ M \in \Lambda^{!,\multimap}_{!\tau}(\Gamma^\circ\ ;\ \emptyset) \;\middle|\; \begin{array}{l} M = !N \text{ for some} \\ N \in P(\Gamma\ ;\ \emptyset) \end{array} \right\} & \text{if } \Delta = \emptyset \\ \emptyset & \text{otherwise} \end{cases}$$

Proof: Easy. □

Now we are ready to repeat the same story as given in [2]:

Definition 2.4 *A family $\{P_\tau\}$ of dual predicates is called a* logical dual predicate *when each P_τ is a dual predicate on τ, and $P_{\tau_1\multimap\tau_2} = P_{\tau_1} \multimap P_{\tau_2}$ as well as $P_{!\tau} = !P_\tau$ hold.*

Proposition 2.5 (Basic Lemma) *Let $\{P_\tau\}$ be a logical dual predicate. For $M_i \in P_{!\tau_i}(\Gamma\ ;\ \Delta_i)$ $(1 \leq i \leq m)$, $M'_j \in P_{\tau'_j}(\Gamma\ ;\ \Delta'_j)$ $(1 \leq j \leq n)$ and $x_1 : \tau_1, \ldots, x_m : \tau_m\ ;\ y_1 : \tau'_1, \ldots, y_n : \tau'_n \vdash N : \tau$, it follows that*

$$\mathsf{let}\ !x_1\ \mathsf{be}\ M_1\ \mathsf{in}\ \ldots \mathsf{let}\ !x_m\ \mathsf{be}\ M_m\ \mathsf{in}\ N[M'_1/y_1, \ldots, M'_n/y_n] \in P_\tau(\Gamma\ ;\ \Delta)$$

where $\Delta = \Delta_1\sharp\ldots\sharp\Delta_m\sharp\Delta'_1\sharp\ldots\sharp\Delta'_n$.

Note that this can be equally stated as follows:

Proposition 2.6 *Let $\{P_\tau\}$ be a logical dual predicate. For $M_i \in P_{!\tau_i}(\Gamma\ ;\ \emptyset)$ $(1 \leq i \leq m)$, $M'_j \in P_{\tau'_j}(\Gamma\ ;\ \Delta'_j)$ $(1 \leq j \leq n)$ and $x_1 : \tau_1, \ldots, x_m : \tau_m\ ;\ y_1 : \tau'_1, \ldots, y_n : \tau'_n \vdash N : \tau$, it follows that*

$$N[M_1/x_1, \ldots, M_m/x_m, M'_1/y_1, \ldots, M'_n/y_n] \in P_\tau(\Gamma\ ;\ \Delta)$$

where $\Delta = \Delta'_1\sharp\ldots\sharp\Delta'_n$.

Proof: Induction on the derivation of terms (equivalently typing judgements) - actually identical as the proof of Basic Lemma in [2], except that we have to add the parameters for linear contexts. □

Corollary 2.7 (Basic Lemma for closed terms) *Let $\{P_\tau\}$ be a logical dual predicate. Then, for any $\emptyset\ ;\ \emptyset \vdash N : \tau$, it follows that $N \in P_\tau(\emptyset\ ;\ \emptyset)$.*

3 Fullness

Definition 3.1 *For a type σ and context $\Gamma\ ;\ \Delta$ of $\lambda^{\to,\multimap}$, define*

$$\mathbf{P}_\sigma(\Gamma\ ;\ \Delta) = \{N \in \Lambda^{!,\multimap}_{\sigma^\circ}(\Gamma^\circ\ ;\ \Delta^\circ) \mid N = M^\circ \text{ for some } M \in \Lambda^{\to,\multimap}_\sigma(\Gamma\ ;\ \Delta)\}$$

Lemma 3.2 $\mathbf{P}$ *is a dual predicate on σ°.*

Proof: Obvious. □

Lemma 3.3 (crucial lemma) $\mathbf{P}_{\sigma_1\to\sigma_2} = !\mathbf{P}_{\sigma_1} \multimap \mathbf{P}_{\sigma_2}$.

Proof: Suppose that $M \in \mathbf{P}_{\sigma_1\to\sigma_2}(\Gamma\ ;\ \Delta)$, i.e. $M = M'^\circ$ for some $\Gamma\ ;\ \Delta \vdash M' : \sigma_1 \to \sigma_2$. Let $N \in !\mathbf{P}_{\sigma_1}(\Gamma'\ ;\ \Delta')$ – by definition we may safely assume that $\Delta' = \emptyset$. This means that $N = !N'$ for some $N' \in \mathbf{P}_{\sigma_1}(\Gamma'\ ;\ \emptyset)$. Therefore $N = !N''^\circ$ for some $\Gamma'\ ;\ \emptyset \vdash N'' : \sigma_1$. It then follows that

$$\begin{array}{rcl} M\,N & = & M'^\circ\,(!N''^\circ) \\ & = & (M' @ N'')^\circ \end{array}$$

Hence $M\,N \in \mathbf{P}_{\sigma_2}(\Gamma'\ ;\ \Delta\sharp\Delta')$. Therefore $M \in (!\mathbf{P}_{\sigma_1} \multimap \mathbf{P}_{\sigma_2})(\Gamma\ ;\ \Delta)$.

Conversely, suppose that $M \in (!\mathbf{P}_{\sigma_1} \multimap \mathbf{P}_{\sigma_2})(\Gamma\ ;\ \Delta)$. Since $!z \in !\mathbf{P}_{\sigma_1}(z : \sigma_1, \Gamma\ ;\ \emptyset)$, we have $M\,(!z) \in \mathbf{P}_{\sigma_2}(z : \sigma_1, \Gamma\ ;\ \Delta)$. Hence there exists $z : \sigma_1, \Gamma\ ;\ \Delta \vdash N : \sigma_2$ such that $M\,(!z) = N^\circ$.

$$\begin{array}{rcl} M & = & \lambda y^{!\sigma_1^\circ}.M\,y \\ & = & \lambda y^{!\sigma_1^\circ}.M\,(\mathsf{let}\ !z^{\sigma_1^\circ}\ \mathsf{be}\ y\ \mathsf{in}\ !z) \\ & = & \lambda y^{!\sigma_1^\circ}.\mathsf{let}\ !z^{\sigma_1^\circ}\ \mathsf{be}\ y\ \mathsf{in}\ M\,(!z) \\ & = & \lambda y^{!\sigma_1^\circ}.\mathsf{let}\ !z^{\sigma_1^\circ}\ \mathsf{be}\ y\ \mathsf{in}\ N^\circ \\ & = & (\boldsymbol{\lambda} y^{\sigma_1}.N)^\circ \end{array}$$

So we conclude that $M \in \mathbf{P}_{\sigma_1\to\sigma_2}(\Gamma\ ;\ \Delta)$. □

Proposition 3.4 *Consider a family $\{P_\sigma\}$ indexed by the types of $\lambda^{\to,\multimap}$, such that P_σ is a dual predicate on σ° and satisfies $P_{\sigma_1\to\sigma_2} = !P_{\sigma_1} \multimap P_{\sigma_2}$. Then there is a logical dual predicate $\{P^*_\tau\}$ such that $P^*_{\sigma^\circ} = P_\sigma$ holds for any type σ of $\lambda^{\to,\multimap}$ – explicitly, such a $\{P^*_\tau\}$ is given by $P^*_b = P_b$, $P^*_{\tau_1\multimap\tau_2} = P^*_{\tau_1} \multimap P^*_{\tau_2}$ and $P^*_{!\tau} = !P^*_\tau$.*

Proof: Induction on the types of $\lambda^{\rightarrow,\multimap}$. The only nontrivial case is that of arrow types $\sigma_1 \rightarrow \sigma_2$:

$$\begin{array}{rcll} P^*_{(\sigma_1\rightarrow\sigma_2)^\circ} & = & P^*_{!\sigma_1^\circ\multimap\sigma_2^\circ} & \\ & = & !P^*_{\sigma_1^\circ} \multimap P^*_{\sigma_2^\circ} & \text{definition of } P^* \\ & = & !P_{\sigma_1} \multimap P_{\sigma_2} & \text{induction hypothesis} \\ & = & P_{\sigma_1\rightarrow\sigma_2} & \end{array}$$

□

Corollary 3.5 *Define a logical dual predicate* $\{\mathbf{P}^*_\tau\}$ *by* $\mathbf{P}^*_b = \mathbf{P}_b$, $\mathbf{P}^*_{\tau_1\multimap\tau_2} = \mathbf{P}^*_{\tau_1} \multimap \mathbf{P}^*_{\tau_2}$ *and* $\mathbf{P}^*_{!\tau} = !\mathbf{P}^*_\tau$. *Then* $\mathbf{P}^*_{\sigma^\circ} = \mathbf{P}_\sigma$ *holds for any type* σ *of* $\lambda^{\rightarrow,\multimap}$.

Theorem 3.6 (fullness) *Suppose that* $\Gamma^\circ \ ; \ \Delta^\circ \vdash N : \sigma^\circ$ *is derivable in* $\lambda^{!,\multimap}$. *Then there exists* $\Gamma \ ; \ \Delta \vdash M : \sigma$ *derivable in* $\lambda^{\rightarrow,\multimap}$ *such that* $\Gamma^\circ \ ; \ \Delta^\circ \vdash M^\circ = N : \sigma^\circ$ *holds.*

Proof: Apply the Basic Lemma to the logical dual predicate $\{\mathbf{P}^*_\tau\}$. □

References

[1] Barber, A. and Plotkin, G. (1997) Dual intuitionistic linear logic. Submitted.

[2] Hasegawa, M. (2000) Girard translation and logical predicates. *J. Functional Programming* **10**(1), 77–89.

[3] Hasegawa, M. (2001) Linearly used effects: Monadic and CPS transformations into the linear lambda calculus (preliminary version). Presented at *2nd Asian Workshop on Programming Languages and Systems*, December 2001.