A New Type Assignment for Strongly Normalizable Terms

Rick Statman

Carnegie Mellon University
Department of Mathematical Sciences
Pittsburgh, PA 15213
(statman@cs.cmu.edu)

Abstract

We consider an operator definable in the intuitionistic theory of monadic predicates and we axiomatize some of its properties in a definitional extension of that monadic logic. The axiomatization lends itself to a natural deduction formulation to which the Curry-Howard isomorphism can be applied. The resulting Church style type system has the property that an untyped term is typable if and only if it is strongly normalizable.

1998 ACM Subject Classification F.4.1 Mathematical Logic

Keywords and phrases lambda calculus

Digital Object Identifier 10.4230/LIPIcs.CSL.2013.634

1 Introduction

Intersection types [4] are very interesting especially in their use for proving untyped terms to be strongly normalizable [6]. However, we view them only as types, and the Curry-Howard isomorphism does not seem to apply. Here we would like to extend the *formulae as types* direction of Curry-Howard to include all strongly normalizable terms. We shall do this by considering a definitional extension of a very weak version of intuitionistic monadic logic. Our notion of typing appears quite different from the clever application of Curry-Howard to the derivations of intersection types for untyped terms in [2]; we do no linearization of untyped terms.

2 Intuitionistic Monadic Logic

We consider the first-order language of intuitionistic monadic predicate logic in the negative fragment. The language consists of two individual constants

$0, 1$

and an arbitrary selection of monadic predicates R. In addition, we shall have two other distinguished monadic predicates

P, Q

that play a special role and remain mostly hidden. We have the connective, $\rightarrow$, the universal quantifier, $\wedge$, and a symbol for falsehood, @. We shall assume that $P0, Q1$ and P and Q are disjoint; that is, $\wedge x(Px \rightarrow (Qx \rightarrow @))$. As usual, we set $\sim F := F \rightarrow @$.

We define a certain definitional extension of our language as follows. Introduce a new connective/relation symbol D which takes a single individual and two formula arguments, and which is defined by $DxFG := (Px \rightarrow F)\ \&\ (Qx \rightarrow G)$.
Indeed, this is the only way that P and Q enter into our discussion.

Computer Science Logic 2013 (CSL'13).
Editor: Simona Ronchi Della Rocca; pp. 634–652
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

D satisfies many interesting properties. Of these, the equivalences

(a)	$D0FG$	$\leftrightarrow$	F	
(b)	$D1FG$	$\leftrightarrow$	G	
(c)	$Dt(F \to G)(H \to K)$	$\leftrightarrow$	$DtFH \to DtGK$	
(d)	$Dt(\wedge yF)G$	$\leftrightarrow$	$\wedge y\ DtFG$	y not free in G, t
(e)	$DtF(\wedge yG)$	$\leftrightarrow$	$\wedge yDtFG$	y not free in F, t

are the most important. They can be verified as follows

(a) Assume F, then $P0 \to F$ and $Q0 \to G$ since $\sim Q0$.
Conversely, assume $D0FG$. Then $P0 \to F$ so F since $P0$.

(b) Similar to (a)

(c) Assume $Dt(F \to G)(H \to K)$. Now assume $DtFH$. To show $DtGK$ assume Pt. Then, since $DtFH$ we get F and since $Dt(F \to G)(H \to K)$ we get $F \to G$. Thus G. So $Pt \to G$. Now assume Qt. Similarly, we get K. Thus, $Qt \to K$. Conversely, assume $DtFH \to DtGK$. To show $Dt(F \to G)(H \to K)$ assume Pt and F. Then $\sim Qt$ so $DtFH$;. Thus, $DtGK$ and hence since Pt, G. Similarly for $H \to K$.

(d) Assume $Dt(\wedge xF)G$. Let x be given. To show $DtFG$ assume Pt. Then $\wedge xF$, in particular, F. Thus, $Pt \to F$. Now assume Qt. By hypothesis G. Thus, $Qt \to G$. But x was arbitrary; thus, $\wedge x(Pt \to F$ and $Qt \to G)$. Conversely, suppose $\wedge x\ DtFG$. To show $Dt(\wedge xF)G$ assume Pt. Let x be given. We have, by assumption, $Pt \to F$ so F. Thus $\wedge xF$. We already have $Qt \to G$. Similarly for the other half.

The above equivalences would be shared by D with any *discriminator*. Discriminators have been extensively studied, and we refer the reader to Bloom & Tindell [3]. Fortunately, the properties above do not depend on the decidability of P and Q, their coverage, nor on their complementarity. For example, in any (Kripke) model satisfying $\wedge x\ (DxFF \leftrightarrow F)$ if the model fails to satisfy $\wedge x\ (Px \vee Qx)$ then the model satisfies $\sim\sim F$. Unfortunately, the five above are not complete in our context. For example,

$$Dt(DrFG)(DrHK) \leftrightarrow Dr(DtFH)(DtGK)$$

is valid but not derivable from the five. This can be seen as an exercise after the next section. There are more; indeed, the set of all valid equivalences is undecidable. If $A[S]$ is any monadic formula on the monadic predicate S then

$$A[S] \text{ is intuitionistically valid } \Leftrightarrow$$

$A[\lambda xDx(Rx)(Rx)] \leftrightarrow (R0 \to R0)$ is a valid equivalence. In particular, Kripke model M for S can be extended by setting $R = S$, $P = M - \{1\}$ and $Q = \{1\}$. Thus, we can apply the theorem of Maslow, Mints, and Orevkov [5].

3 Natural Deduction and Rewrite Rules

From now on, we consider only the restricted language without @, P, and Q, and with D as a primitive symbol. The above equivalences can be formulated as reduction rules;

$(0) D0FG$	$\rightarrowtail$	F	
$(1) D1FG$	$\rightarrowtail$	G	
$(\rightarrow) Dt(F \rightarrow G)(H \rightarrow K)$	$\rightarrowtail$	$(DtFH) \rightarrow (DtGK)$	
$(\wedge) Dt(\wedge uF)G$	$\rightarrowtail$	$\wedge uDtFG$	u not free in G or t
$(\wedge) DtF(\wedge vG)$	$\rightarrowtail$	$\wedge v\ DtFG$	v not free in F or t
$(\$) \wedge uF$	$\rightarrowtail$	F	u not free in F
$(\$\$) \wedge u \wedge vF$	$\rightarrowtail$	$\wedge v \wedge uF$	

Here ($) and ($$) make for a smoother theory. The congruence generated by $\rightarrowtail$ is called formula conversion (conv.).

$D - \{(0), (1)\}$ is denoted $D-$. The following are easily verified.

Facts:

(i) $D-$ reductions satisfy the weak-diamond property.
(ii) A given formula $D-$ reduces to only finitely many others.
(iii) $D-$ reduction has the strong diamond property, modulo ($$).
There is an obvious notion of residual for reductions. Residuals, if they exist, are unique and the corresponding reductions commute modulo the order of $\wedge$'s.
(iv) $D-$ is Church-Rosser.
(v) $(0) + (1)$ has unique normal forms.
(vi) There is a *strip lemma* for $(0)+(1)$ -reduction over D; If $F \twoheadleftarrow G$ in general, and $G \twoheadrightarrow H$ by $(0) + (1)$ then $H \twoheadrightarrow K$ in general and $F \twoheadrightarrow K$ by $(0) + (1)$.
(vii) $\rightarrowtail$ is Church-Rosser.
(viii) There is a *standardization theorem* for $(\wedge)$;
If $F \twoheadrightarrow \wedge yG$ then there exists H such that $F \twoheadrightarrow \wedge yH$ by $(\wedge)$ and ($$) alone and $H \twoheadrightarrow G$.
(ix) A formula F in $\rightarrowtail$ normal form has the properties

(a) F does not contain $D0HK$ or $D1HK$

(b) If F contains DtF_0F_1 then if F_i is not atomic then F_i has the form $H \rightarrow K$ and F_{1-i} is atomic.

(i) $\rightarrowtail$ normal forms are unique up to ($$).

If we have a formula F with no variable both free and bound and no variable bound twice (alpha normal form) and we require that in ($) expansions u is new, then in any conversion F conv. G each quantifier $\wedge v$ has at most one descendant in G. The *replacement* of $\wedge v$ in F by t in this conversion is the result of substituting t for every occurrence of v and omitting $\wedge v$. The result is a valid conversion when redundant steps are omitted.

Now we have the natural deduction rules for intuitionistic monadic logic with D

$$(\to I)\quad \cfrac{\begin{array}{c}/\\F\\/\\ \vdots\\ G\end{array}}{F \to G}$$

$$(\to E)\quad \cfrac{\begin{array}{c}\vdots\\ F \to G\end{array}\qquad \begin{array}{c}\vdots\\ F\end{array}}{G}$$

$$(\wedge I)\quad \cfrac{\begin{array}{c}\vdots\\ F\end{array}}{\wedge v F}$$

v (eigenvariable) not free in any assumption

$$(\wedge E)\quad \cfrac{\begin{array}{c}\vdots\\ \wedge v F\end{array}}{[t/v]F}$$

t free for v in F (i.e. if t is a variable it should not become bound by substitution)

$$(\text{conv.})\quad \cfrac{\begin{array}{c}\vdots\\ F\end{array}}{G}\qquad F \text{ conv. } G$$

These rules will be transformed into typing rules for untyped lambda terms.

4 Typing Rules

Although we are essentially using Church typing, two deviations from normal conventions will be employed. First, as already mentioned, we adopt Curry style derivations. This is merely a notational convenience. In a second notational convenience, we dispense with lambda abstraction over object variables appearing in types normally used as a coercion into universally quantified types. Instead, we could distinguish the notion of a *typed sub-term* from an untyped one, so an untyped sub-term begins a nested set of typed ones; but, in practice, we shall just refer to a particular point in the typing derivation tree. Nevertheless, each typed sub-term has a unique type. The typing rules are the following.

$$(\to I)\quad \frac{\begin{matrix} / \\ x : F \\ / \\ \vdots \\ X : G \end{matrix}}{\lambda x X \; : \; F \to G}$$

$$(\to E)\quad \frac{\begin{matrix} \vdots & & \vdots \\ X : F \to G & & Y : F \end{matrix}}{(XY) : G}$$

$$(\wedge I)\quad \frac{\begin{matrix} \vdots \\ X : F \end{matrix}}{X : \wedge v F} \qquad v \text{ (eigenvariable) not free in the type of any free variable of } X$$

$$(\wedge E)\quad \frac{\begin{matrix} \vdots \\ X : \wedge v F \end{matrix}}{X : [t/v]F} \qquad t \text{ free for } v \text{ in } F$$

$$(\text{conv.})\quad \frac{\begin{matrix} \vdots \\ X : F \end{matrix}}{X : G} \qquad F \text{ conv. } G$$

where x can occur in an assumption $x : F$ with at most one F. These rules are to be understood in the obvious way. Each derivation of $X : G$, for untyped X, corresponds to a Church typing of X with G, and the free variables of X with the types assigned to them in the assumptions.

5 Reductions of Derivations

With each conversion, F (conv.) G, a pair of reductions

$$F \rightarrowtail\!\!\!\rightarrow H \leftarrow\!\!\!\leftarrowtail G$$

can be associated. We can always assume that H has only 0 and 1 in atomic sub-formulae $R0$ or $R1$, and no $D0KK'$ or $D1KK'$, and by the strip lemma, each reduction begins with $(0), (1)$ reductions and proceeds afterwards with none. Two conversions

$$F \text{ conv. } G \text{ conv. } H$$

in a row can be transformed into a single one

$$F \text{ conv. } H$$

We now define the notion of a "derivation reduction" in 4 parts.

(1) In three successive inferences $(\wedge I)$, (conv.), $(\wedge E)$

$$\begin{array}{c} \vdots \\ F \\ \hline \wedge v \quad F \\ \hline \wedge u \quad G \\ \hline [t/u] \quad G \end{array}$$

either $\wedge u$ is a descendant of $\wedge v$ or it is not. In the first case, omitting trivial pairs of ($$)'s we have

$$\begin{array}{c} [t/v] \quad \vdots \\ [t/v] \quad F \\ \hline [t/u] \quad G \end{array}$$

In the latter case, there is the trivial case that $\wedge v$ is omitted by ($), and we have

$$\begin{array}{c} \vdots \\ F \\ \hline \wedge u\ G \\ \hline [t/u]\ G \end{array} \qquad\qquad \text{(conv.)}$$

Otherwise, in case $\wedge u$ is omitted by ($), we have

$$\dfrac{\dfrac{\begin{matrix}\vdots\\ F\end{matrix}}{\wedge v\ F}}{[t/u]\quad G}\ \text{(conv.)}$$

Finally, we have

$\wedge v\ F$ (conv.) $\wedge v\ \wedge u\ H(v,u)$ by ($\wedge$) and ($$)
$\wedge v\ \wedge u\ H(v,u) \rightarrowtail \wedge v\ \wedge u\ K(v,u)$
$\wedge u\ \wedge v\ K(v,u) \leftarrowtail \wedge u\ \wedge v\ L(u,v)$
$\wedge u\ \wedge v\ L(u,v) \leftarrowtail \wedge u\ G$ by ($\wedge$) and ($)

and so

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\begin{matrix}\vdots\\ F\end{matrix}}{\wedge u\ H(v,u)}}{H(v,t)}}{L(t,v)}}{\wedge v\ L(t,v)}}{[t/u]G}$$

In this manner the three successive inferences are reduced to either

(conv.)
(conv.), ($\wedge E$)
($\wedge I$), (conv.) or
($\wedge E$), (conv.), ($\wedge I$)

(2) In three successive inferences ($\rightarrow I$), (conv.), ($\wedge E$);

$$
\dfrac{\dfrac{\dfrac{\begin{array}{c} /\\ F\\ /\\ \cdot\\ \cdot\\ \cdot\\ G \end{array}}{F \to G}}{\wedge u\ H}}{[t/u]H}
$$

omitting trivial pairs of ($$)'s, there exist H', F', G' such that

$$
\begin{array}{lcccc}
\wedge uH \rightarrowtail\!\!\twoheadrightarrow \wedge uH' \rightarrowtail H' \text{ by } (\$) & & & & \\
\quad H' & \rightarrowtail\!\!\twoheadrightarrow & (F' \to G') & \twoheadleftarrow\!\!\leftarrowtail & (F \to G) \\
\quad H' & \twoheadleftarrow\!\!\leftarrowtail & [t/u]H & &
\end{array}
$$

and so the three successive inferences can be reduced to an $(\to I)$, (conv.).

(3) In three successive inferences $(\wedge I)$, (conv.), $(\to E)$;

$$
\dfrac{\dfrac{\dfrac{\begin{array}{c} /\\ F\\ /\\ \cdot\\ \cdot\\ \cdot\\ H \end{array}}{\wedge u\ H}}{F \to G} \qquad \begin{array}{c} \cdot\\ \cdot\\ \cdot\\ F \end{array}}{G}
$$

omitting trivial pairs of ($$)'s, there exist H', F', G' such that

$$
\begin{array}{lclccl}
\wedge uH & \rightarrowtail\!\!\twoheadrightarrow & \wedge uH' & \rightarrowtail & H' & \text{by } (\$) \\
\quad H' & \rightarrowtail\!\!\twoheadrightarrow & (F' \to G') & \twoheadleftarrow\!\!\leftarrowtail & (F \to G) & \\
\quad H' & \twoheadleftarrow\!\!\leftarrowtail & H & & &
\end{array}
$$

and so the three successive inferences can be reduced to an (conv.) $(\to E)$.

(4) In three successive inferences $(\to I)$, (conv.), $(\to E)$;

$$
\begin{array}{cc}
/ & \\
F & \\
/ & \\
\vdots & \\
G & \\
\hline
F \to G & \vdots \\
\hline
H \to K & H \\
\hline
\multicolumn{2}{c}{K}
\end{array}
$$

we have F conv. H and G conv. K. These three successive inferences can be reduced to

$$
\begin{array}{c}
\vdots \\
H \\
\hline
F \\
\vdots \\
G \\
\hline
K
\end{array}
$$

which uses only (conv.).

A *segment* in a derivation is an alternating sequence of $(\wedge I)$, or $(\wedge E)$ inferences and conversions. Thus, in a segment, we can assume that no $(\wedge I)$ precedes an $(\wedge E)$ by applying suitable reductions (1)-(3). Thus, if a segment begins and ends in a formula beginning with $\to$, it is simply a conversion. When employing the rules as typing rules, applying reductions (1)-(3) to segments does not alter the untyped term being typed.

6 The Main Result

We shall now prove that an untyped term is strongly normalizable if, and only if, it has a type in our system.

▶ **Lemma 1.** *Suppose that we have typings $X : F$ and $X : G$ of the untyped X. Then there exists a typing $X : DvFG$ for v new. Moreover, if, for the free variable x, we have $x : H$ in $X : F$ and $x : K$ in $X : G$ then $x : DvHK$ in $X : DvFG$.*

▶ **Lemma 2.** *A normal untyped term X has a typing $X : F$.*

▶ **Lemma 3.** *Suppose that x occurs in X and $[Y/x]X$ has a typing*

$$[Y/x]X : F.$$

Then there is a typing $(\lambda x X)Y : F$, where the free variables of X may have new types.

▶ Proposition 1. If X is strongly normalizable then for some F we have a typing $X : F$.

Proof. Now suppose that X is strongly normalizable. We show that X has a typing by induction on the reduction tree of X with a subsidiary induction on the length of X. This is really induction on Barendregts's perpetual reduction strategy beginning with X ([1] pg. 334). We can write $X :=$

$$\lambda x_1 \ldots x_r \begin{cases} x_i \\ \\ (\lambda x.X_0) \end{cases} X_1 \ldots X_s$$

Case 1: $r > 0$. Then the induction hypothesis on length can be applied directly.

Case 2: $r = 0$ and there is no head redex. This is just like the case of normal terms.

Case 3: $r = 0$ and X has a head redex. We distinguish two subcases.

Subcase 1: x is not free in X_0. Now both X_1 and $X_0 X_2 \ldots X_s$ have shorter reduction trees than X. Thus, by induction hypothesis, both have typings with $X_1 : G$. We may adjust the typings of the free variables in X_1 and $X_0 X_2 \ldots X_s$ so that they match as in the case of normal terms. Thus, we have a typing of X with $x : G$.

Subcase 2: x appears free in X_0. Now the reduction tree of

$$([X_1/x]X_0)X_2 \ldots X_s$$

is smaller than that of X so the induction hypothesis applies and this term has a typing. Now we can apply Lemma 3 and adjust the types of the free variables in $([X_1/x]X_0)X_2 \ldots X_2$. ◀

▶ **Lemma 4.** *If $X : G$ is strongly normalizable with $y : F$ and $Y : F$ is strongly normalizable then $[Y/y]X : G$ is strongly normalizable.*

▶ Proposition 2. If the untyped term X has a typing $X : F$ then X is strongly normalizable.

Proof. By induction on X where we again write $X :=$

$$\lambda x_1 \ldots x_r \begin{cases} x_i \\ \\ (\lambda x X_0). \end{cases} X_1 \ldots X_s$$

Lemma 4 prevails. ◀

Acknowledgement. The author would like to thank refree 3 (the expert) for his many thoughtful and usefull suggestions. There was neither time nor space to include them all in the version of this paper.

References

1 H. P. Barendegt, "The Lambda Calculus", North Holland, 1984.

2 A. Bucciarelli, Piperno, A., and Salvo, I., Intersection types and lambda definability, MSCS03 **13** (1), 2003, pp. 15-53.

3 S. Bloom and R. Tindell, Varieties of "if-then-else", *SICOMP* **12** (4), 1983.

4 M. Coppo and M. Dezani, A new type assignment for lambda terms, *Archiv fur Math. Logik*, **19**, 1978.

5 S. Ghilezan, Strong normalization and typability with intersection types, *Notre Dame Journal of Formal Logic*, **37** (1), 1996.

6 S. A. Maslow, G. E. Mints, and V. P. Orevkov, Unsolvability in the constructive predicate calculus, *Soviet Math. Doklady* **4** 1963, pp. 1365-1367.

7 G. Pottinger, "A type assignment for strongly normalizable lambda terms", Curry Festschrift, 1980.

8 M. Sorenson and P. Urzyczyn, Lectures on the Curry-Howard Isomorphism Manuscript, 1998.

A Proof of Lemma 1

Proof. Lemma 1. By induction on X.

Basis: $X = x$. The typings $X : F$ and $X : G$ are both segments, which we can assume are the same length by adding trivial conversions. In addition, we can assume that all the variables which occur bound in one typing are distinct from the variables which at some point occur free in the other. Then we can simulate both typings in a typing by $DvFG$ as follows.

From $X : F$ to $X : DvFG$

$$\frac{x \; : \; H}{x \; : \; \wedge u H} \quad \mapsto \quad \cfrac{\cfrac{x : DvHK}{x : \wedge u DvHK}}{x : Dv(\wedge uH)K}$$

$$\frac{x : \wedge uH}{x : [t/u]H} \quad \mapsto \quad \cfrac{\cfrac{x : Dv(\wedge uH)K}{x : \wedge u \; DvHK}}{x : [t/u]DvHK}$$

and similarly for from $X : G$ to $X : DvFG$.

Induction step:

Case 1: $X = (YZ)$

In $X : F$ we have

$$\begin{array}{llll}
 & \vdots & & \vdots \\
Y & : H & Z & : M \\
 & \vdots \text{ (segment)} & & \vdots \text{ (segment)} \\
Y & : K \to L & Z & : K \\
\hline
 & & (YZ) & : L \\
 & & & \vdots \text{ (segment)} \\
 & & (YZ) & : F
\end{array}$$

and in $X : G$ we have

$$
\begin{array}{llll}
 & \vdots & & \vdots \\
Y & : H' & Z & : M' \\
 & \cdot\ \text{(segment)} & & \cdot\ \text{(segment)} \\
Y & : K' \to L' & Z & : K' \\
\hline
 & & (YZ) & : L' \\
 & & & \cdot\ \text{(segment)} \\
 & & (YZ) & : G
\end{array}
$$

The segments can be simulated as in the basis case and this arrives at

$$
\begin{array}{lll}
\vdots & & \vdots \\
Y : DvHH' & & z : DvMM' \\
\cdot\ \text{(segment)} & & \cdot\ \text{(segment)} \\
\underline{Y : Dv(K \to L)(K' \to L')} & \text{(conv.)} & \cdot \\
Y : DvKK' \to DvLL' & & Z : DvKK' \\
\hline
 & (YZ) : DvLL' &
\end{array}
$$

and the final segment can be, again, simulated as in the basis case.

Case 2: $X = \lambda y Y$.

In $X : F$ we have

$$
\begin{array}{c}
/ \\
y : H \\
/ \\
\vdots \\
Y : K \\
\hline
\lambda y Y : H \to K \\
\cdot\ \text{(segment)} \\
X : F
\end{array}
$$

In $X : G$ we have

$$\begin{array}{c} / \\ y : H' \\ / \\ \vdots \\ Y : K' \\ \hline \lambda y Y : H' \to K' \\ \vdots \ \text{(segment)} \\ X : G \end{array}$$

and so we have the simulation

$$\begin{array}{lcl} & / & \\ y & : & DvHH' \\ & / & \\ & \vdots & \\ Y & : & DvKK' \\ \hline \lambda y Y & : & DvHH' \to DvKK' \\ \hline X & : & Dv(H \to K)(H' \to K') \end{array} \quad \text{(conv.)}$$

and the final segment can be, again, simulated as in the basis case. ◀

As usual we say that an untyped term is *strongly normalizable* if every beta reduction sequence terminates.

B Proof of Lemma 4

Proof. Lemma 4: By induction where we let

k = the length of any $\rightarrowtail$ normal form of F,
l = the size of the reduction tree of Y,
m = the size of the reduction tree of X,
n = the length of X

and we order the 4-tuples (k, l, m, n) lexicographically. As before we write $X :=$

$$\lambda x_1 \dots x_r \begin{cases} x_i \\ \\ (\lambda x X_0) \end{cases} X_1 \dots X_2$$

Case 1: $r > 0$. In this case the result follows from the induction hypothesis applied to n.

Case 2: $r = 0$ and x_i is not the free variable y. In this case the result follows by the induction hypothesis applied to the X_j in place of X.

Case 3: $r = 0$ and X has the head redex $(\lambda x X_0)X_1$. In this case the typing of X has the form

$$
\begin{array}{lcl}
\quad / & & \\
x & : & H,\ y : F \\
/ & & \\
 & \vdots & \\
X_0 & : & J \\
\hline
\end{array}
$$

**** $(\to I)$

$$
\begin{array}{lclcccl}
\lambda x.X_0 & : & H \to J & & y & : & F \\
 & \cdot & \text{segment} & & & \cdot & \\
 & \cdot & & & & \cdot & \\
\lambda x.X_0 & : & K \to L & & X_1 & : & K \\
\hline
\end{array}
$$

$$
\begin{array}{lcl}
(\lambda x.X_0)X_1 & : & L \\
 & \cdot & \\
 & \cdot & \text{(segment)} \\
 & \cdot & \\
(\lambda x.X_0)X_1 & : & L \\
 & \vdots & \\
X & : & G \qquad .
\end{array}
$$

By the remark before Lemma 1 we may assume that the segment from $H \to J$ to $K \to L$ is actually a conversion which factors into a conversion of H to K followed by a conversion of J to L. We note here that no eigenvariable in either segment displayed above can occur in F. We distinguish two subcases.

Subcase 1: x is not free in X_0. By induction hypothesis on n, $[Y/y]X_1$ is strongly normalizable. In addition,

$$
\begin{array}{rcl}
y & : & F \\
 & \vdots & \\
X_0 & : & J \\
\hline
\end{array}
$$

(conv.)

$$
\begin{array}{rcl}
X_0 & : & M \\
 & \vdots & \\
X_0 X_2 \dots X_s & : & G.
\end{array}
$$

is a typing of the strongly normalizable $X_0X_2 \dots X_s$ for which the induction hypothesis applies to m. Thus,

$$[Y/y](X_0X_2 \dots X_s)$$

is strongly normalizable. But then Barendregt's perpetual strategy terminates when applied to $[Y/y]X$, so $[Y/y]X$ is strongly normalizable.

Subcase 2: x is free in X_0. Now

$$\begin{array}{rcl} y & : & F \\ & \vdots & \\ X_1 & : & K \\ \hline & & \\ X_1 & : & H, \quad y : F \\ & \vdots & \\ [X_1/x]X_0 & : & J \\ \hline & & \\ [X_1/x]X_0 & : & M \\ & \vdots & \\ \multicolumn{3}{l}{([X_1/x]X_0)X_2 \dots X_s : G} \end{array} \qquad \begin{array}{l} \\ \\ \\ (\text{conv.}) \\ \\ \\ \\ (\text{conv.}) \\ \\ \\ \end{array}$$

is a typing of $([X_1/x]X_0)X_2 \dots X_s$ and the induction hypothesis applies to m. Thus, $([X_1/x]X_0)X_2 \dots X_s$ is strongly normalizable and Barendregt's perpetual strategy terminates when applied to $[Y/y]X$. Thus, $[Y/y]X$ is strongly normalizable.

Case 4: $r = 0$ and $x_i = y$ is free in X. Let

$$Y = \lambda y_1 \dots y_t \begin{cases} y_k \\ \\ \lambda z.Z \end{cases} Y_1 \dots Y_q$$

Subcase i: $t = 0$ and Y has no head redex. Then $[Y/y]X$ is strongly normalizable by the induction hypothesis for n applied to the terms $[Y/y]X_j$ and the assumption that Y is strongly normalizable applied to the terms Y_j.

Subcase ii: $t = 0$ and Y has a head redex. By induction hypothesis for n applied to the $[Y/y]X_j$ we have that

$$x[Y/y]X_1 \dots [Y/y]X_s$$

is strongly normalizable and we can apply the induction hypothesis for ℓ to

$$[(([Y_1/z]Z)Y_2 \dots Y_q/x](x[Y/y]X_1 \dots [Y/y]X_s)$$

where $x : F$.

Subcase iii: $t > 0$. For easier notation let $Y = \lambda z.Z$. In this case the typing of X has the form

$$\dfrac{\dfrac{\begin{array}{ccl} y & : & F \\ & \vdots & \text{(segment)} \\ y & : & H \to K \end{array} \qquad \begin{array}{ccl} y & : & F \\ & \vdots & \\ X_1 & : & H \end{array}}{yX_1 : K}}{\begin{array}{ccl} yX_1 & : & J \\ & \vdots & \\ X & : & G \end{array}} \;\text{(conv.)} \quad , \quad y : F$$

and the typing of Y has the form

$$\dfrac{\begin{array}{c} / \\ Z : L \\ / \\ \vdots \\ Z : M \end{array}}{\begin{array}{ccl} \lambda z.Z & : & L \to M \\ & \vdots & \text{(segment)} \\ \lambda z.Z & : & F \end{array}} \;(\to I)$$

Now the segment

$$\begin{array}{ccl} \lambda z.Z & : & L \to M \\ & \vdots & \text{(segment)} \\ \lambda z.Z & : & F \\ & \vdots & \text{(segment)} \\ \lambda z.Z & : & H \to K \end{array}$$

reduces to a conversion by the remark preceeding Lemma 1 and we have the typings

$$\begin{array}{lcl} Z & : & H \\ \hline & & \\ z & : & L' \\ & \cdot & \\ & \cdot & \\ Z & : & M' \\ \hline & & \\ Z & : & J \end{array} \qquad \text{(conv.)}$$

for suitable instances L' of L and M' of M and

$$\begin{array}{rcl} & / & \\ & Z : L & \\ & / & \\ & \cdot & \\ & \cdot & \\ & \cdot & \\ Z & : & M \\ \hline & & \\ \lambda z.Z & : & L \to M \\ & \cdot & \\ & \cdot & \text{(segment)} \\ & \cdot & \\ \lambda z Z & : & F \\ & \cdot & \\ & \cdot & \\ & \cdot & \\ [Y/y]X_1 & : & H \end{array} \qquad (\to I)$$

and

$$\begin{array}{rcl} & / & \\ Z & : & L \\ & / & \\ & \cdot & \\ & \cdot & \\ & \cdot & \\ Z & : & M \\ \hline & & \\ \lambda z.Z & : & L \to M \\ & \cdot & \\ & \cdot & \text{(segment)} \\ & \cdot & \\ x : J, \quad Y & : & F \\ & \cdot & \\ & \cdot & \\ x([Y/y]X_2)\ldots([Y/y]X_s) & : & G \end{array} \qquad (\to I)$$

Thus, by induction hypothesis for n, $x([Y/y]X_2)\ldots([Y/y]X)s$, is strongly normalizable. By induction hypothesis for n, $[Y/y]X_1$ is strongly normalizable. Now the length of any $\rightarrowtail$ normal form of H is less than that of F since H conv. L' and $L \to M$ conv. F. Thus, by induction hypothesis for k

$$[[Y/y]X_1/z]Z$$

is strongly normalizable. In addition, the length of any $\rightarrowtail$ normal form of J is less than that of F since H conv. L' and $L \to M$ conv. F. Thus, by induction hypothesis for k

$$([[Y/y]X_1/z]Z)([Y/y]X_2)\ldots([Y/y]X_s)$$

is strongly normalizable. Hence, Barendregt's perpetual strategy terminates for $[Y/y]X$ and it is strongly normalizable. ◀