

A NOTE ON SHORTEST DEVELOPMENTS

MORTEN HEINE SØRENSEN

Formalit, Byenden 32, 4660 Store Heddinge
e-mail address: mhs@formalit.dk

ABSTRACT. De Vrijer has presented a proof of the *finite developments* theorem which, in addition to showing that all developments are finite, gives an effective reduction strategy computing longest developments as well as a simple formula computing their length.

We show that by applying a rather simple and intuitive principle of duality to de Vrijer’s approach one arrives at a proof that *some* developments are finite which in addition yields an effective reduction strategy computing shortest developments as well as a simple formula computing their length. The duality fails for general β -reduction.

Our results simplify previous work by Khasidashvili.

1. INTRODUCTION

Let $S = \{ “(”, “)”, “.”, “\underline{\lambda}”, “\lambda” \}$ and V be an infinite alphabet (ranged over by x, y, z) disjoint from S . Then $\underline{\Lambda}_K$ is the set of words over $S \cup V$ defined by:

$$\begin{aligned} x \in V &\Rightarrow x \in \underline{\Lambda}_K \\ P \in \underline{\Lambda}_K &\Rightarrow (\lambda x.P) \in \underline{\Lambda}_K \\ P, Q \in \underline{\Lambda}_K &\Rightarrow (P Q) \in \underline{\Lambda}_K \\ P, Q \in \underline{\Lambda}_K &\Rightarrow ((\lambda x.P)Q) \in \underline{\Lambda}_K \end{aligned}$$

We assume the reader is familiar with the fundamental conventions, definitions, and properties pertaining to $\underline{\Lambda}_K$ —see, *e.g.*, [1]—notably the conventions for omitting parentheses, the notions of free and bound variables, the identification of terms that differ only in the choice of names for bound variables, the conventions for avoiding confusion between free and bound variables, the definition of substitution $M\{x := N\}$, and the set $\text{FV}(M)$ of variables occurring free in M . Also, $M \equiv N$ means that M and N differ only in the choice of names for bound variables.

Let $\rightarrow_{\underline{\beta}}$ be the smallest relation on $\underline{\Lambda}_K$ with $(\lambda x.P)Q \rightarrow_{\underline{\beta}} P\{x := Q\}$ satisfying

$$\begin{aligned} P \rightarrow_{\underline{\beta}} P' &\Rightarrow \lambda x.P \rightarrow_{\underline{\beta}} \lambda x.P' \\ P \rightarrow_{\underline{\beta}} P' &\Rightarrow P Q \rightarrow_{\underline{\beta}} P' Q \\ P \rightarrow_{\underline{\beta}} P' &\Rightarrow Q P \rightarrow_{\underline{\beta}} Q P' \\ P \rightarrow_{\underline{\beta}} P' &\Rightarrow (\lambda x.P) Q \rightarrow_{\underline{\beta}} (\lambda x.P') Q \\ P \rightarrow_{\underline{\beta}} P' &\Rightarrow (\lambda x.Q) P \rightarrow_{\underline{\beta}} (\lambda x.Q) P' \end{aligned}$$

2000 ACM Subject Classification: F.4.1.

Key words and phrases: λ -calculus, developments, shortest reductions, longest reductions.

A development of M_0 is a finite or infinite sequence $M_0 \rightarrow_{\underline{\beta}} M_1 \rightarrow_{\underline{\beta}} \dots$. If the sequence is finite, it ends in the last term M_n and has length n . If it is infinite, it has length ∞ .¹ We write $M \in \text{NF}_{\underline{\beta}}$ and call M a $\underline{\beta}$ -normal form if $M \not\rightarrow_{\underline{\beta}} N$ for all $N \in \underline{\Lambda}_K$. A development is complete if it is infinite or ends in a $\underline{\beta}$ -normal form. By $s_{\underline{\beta}}(M)$ and $l_{\underline{\beta}}(M)$ we denote the length of a shortest complete and longest complete development of M , respectively. The *finite developments* theorem, due to Curry and Feys [3] and later proved by many others, states in its simplest form that all developments are finite.

Lemma 1.1.

- (i) $M, N \in \underline{\Lambda}_K \Rightarrow M\{x := N\} \in \underline{\Lambda}_K$;
- (ii) $M \in \underline{\Lambda}_K \ \& \ M \rightarrow_{\underline{\beta}} N \Rightarrow N \in \underline{\Lambda}_K$.

Proof. (i): By induction on M . (ii): By induction on $M \rightarrow_{\underline{\beta}} N$, using (i). □

2. SHORTEST DEVELOPMENTS

We first present our technique for computing shortest developments and then explain the relation to de Vrijer's [7] technique for computing longest developments in §4.

Definition 2.1.

- (i) For all $x \in V$ define $m_x : \underline{\Lambda}_K \rightarrow \mathbb{N}$ by:²

$$\begin{aligned}
 m_x(x) &= 1 \\
 m_x(y) &= 0 && \text{if } x \neq y \\
 m_x((\underline{\lambda}y.P) Q) &= m_x(P) + m_x(Q) \lfloor m_y(P), 1 \rfloor \\
 m_x(P Q) &= m_x(P) + m_x(Q) && \text{if } P \neq \underline{\lambda}y.R \\
 m_x(\underline{\lambda}y.P) &= m_x(P)
 \end{aligned}$$

- (ii) Define $h : \underline{\Lambda}_K \rightarrow \mathbb{N}$ by:

$$\begin{aligned}
 h(x) &= 0 \\
 h((\underline{\lambda}y.P) Q) &= h(P) + h(Q) \lfloor m_y(P), 1 \rfloor + 1 \\
 h(P Q) &= h(P) + h(Q) && \text{if } P \neq \underline{\lambda}y.R \\
 h(\underline{\lambda}y.P) &= h(P)
 \end{aligned}$$

- (iii) Define $H : \underline{\Lambda}_K \rightarrow \underline{\Lambda}_K$ by:

$$\begin{aligned}
 H(x) &= x \\
 H((\underline{\lambda}y.P) Q) &= \begin{cases} (\underline{\lambda}y.P) H(Q) & \text{if } \lfloor m_y(P), 1 \rfloor = 1 \ \& \ Q \notin \text{NF}_{\underline{\beta}} \\ P\{y := Q\} & \text{otherwise} \end{cases} \\
 H(P Q) &= \begin{cases} H(P) Q & \text{if } P \neq \underline{\lambda}y.R \ \& \ P \notin \text{NF}_{\underline{\beta}} \\ P H(Q) & \text{if } P \neq \underline{\lambda}y.R \ \& \ P \in \text{NF}_{\underline{\beta}} \end{cases} \\
 H(\underline{\lambda}y.P) &= \underline{\lambda}y.H(P)
 \end{aligned}$$

- (iv) Let $L_H(M)$ be the length of the complete development

$$M \rightarrow_{\underline{\beta}} H(M) \rightarrow_{\underline{\beta}} H(H(M)) \rightarrow_{\underline{\beta}} \dots$$

¹We adopt the conventions $n \leq \infty$ and $\infty + n = \infty$ for all $n \in \mathbb{N} \cup \{\infty\}$.

² $\lfloor m, n \rfloor$ and $\lceil m, n \rceil$ denote the minimum and maximum of m and n , respectively.

As will be seen in Corollary 2.8, $M \rightarrow_{\underline{\beta}} H(M) \rightarrow_{\underline{\beta}} H(H(M)) \rightarrow_{\underline{\beta}} \dots$ is a shortest complete development from M , and $h(M)$ is its length. Informally, the auxiliary function $m_x(M)$ calculates the number of copies of N we have to reduce in a shortest complete development of $M\{x := N\}$.

Remark 2.2.

- (i) $x \notin \text{FV}(M) \Rightarrow m_x(M) = 0$.
- (ii) $M \in \text{NF}_{\underline{\beta}} \Leftrightarrow h(M) = 0$.
- (iii) $\lfloor m_y(P), 1 \rfloor \neq 1 \Rightarrow \lfloor m_y(P), 1 \rfloor = m_y(P)$.

Lemma 2.3. *Let $x \neq y$. Then:*

- (i) $m_y(M\{x := N\}) = m_y(M) + m_y(N)m_x(M)$;
- (ii) $h(M\{x := N\}) = h(M) + h(N)m_x(M)$.

Proof. (i) is by induction on M . Let $L^* \equiv L\{x := N\}$.

1. $M \equiv z$.

1.1. $z \equiv x$. Then

$$\begin{aligned} m_y(x^*) &= m_y(N) \\ &= m_y(x) + m_y(N)m_x(x) \end{aligned}$$

1.2. $z \neq x$. Then

$$\begin{aligned} m_y(z^*) &= m_y(z) \\ &= m_y(z) + m_y(N)m_x(z) \end{aligned}$$

2. $M \equiv (\underline{\lambda}z.P) Q$. Since $z \notin \text{FV}(N)$, also $m_z(N) = 0$. By the induction hypothesis,

$$\begin{aligned} &m_y((\underline{\lambda}z.P^*) Q^*) \\ &= m_y(P^*) + m_y(Q^*) \lfloor m_z(P^*), 1 \rfloor \\ &= m_y(P) + m_y(N)m_x(P) + (m_y(Q) + m_y(N)m_x(Q)) \lfloor m_z(P), 1 \rfloor \\ &= m_y(P) + m_y(N)m_x(P) + m_y(Q) \lfloor m_z(P), 1 \rfloor + m_y(N)m_x(Q) \lfloor m_z(P), 1 \rfloor \\ &= m_y(P) + m_y(Q) \lfloor m_z(P), 1 \rfloor + m_y(N)(m_x(P) + m_x(Q) \lfloor m_z(P), 1 \rfloor) \\ &= m_y((\underline{\lambda}z.P) Q) + m_y(N)m_x((\underline{\lambda}z.P) Q) \end{aligned}$$

3. $M \equiv P Q$ where $P \neq \underline{\lambda}y.R$. Then, by the induction hypothesis,

$$\begin{aligned} m_y(P^* Q^*) &= m_y(P^*) + m_y(Q^*) \\ &= m_y(P) + m_y(N)m_x(P) + m_y(Q) + m_y(N)m_x(Q) \\ &= m_y(P Q) + m_y(N)m_x(P Q) \end{aligned}$$

4. $M \equiv \underline{\lambda}y.P$. Similar to Case 3.

This concludes the proof of (i); (ii) is also by induction on M .

1. $M \equiv z$.

1.1. $z \equiv x$. Then

$$\begin{aligned} h(x^*) &= h(N) \\ &= h(x) + h(N)m_x(x) \end{aligned}$$

1.2. $z \neq x$. Then

$$\begin{aligned} h(z^*) &= h(z) \\ &= h(z) + h(N)m_x(z) \end{aligned}$$

2. $M \equiv (\underline{\lambda}z.P) Q$. Since $z \notin \text{FV}(N)$, also $m_z(N) = 0$. Therefore, by the induction hypothesis and (i),

$$\begin{aligned}
& h((\underline{\lambda}z.P^*) Q^*) \\
&= h(P^*) + h(Q^*)[m_z(P^*), 1] + 1 \\
&= h(P) + h(N)m_x(P) + (h(Q) + h(N)m_x(Q))[m_z(P), 1] + 1 \\
&= h(P) + h(N)m_x(P) + h(Q)[m_z(P), 1] + h(N)m_x(Q)[m_z(P), 1] + 1 \\
&= h(P) + h(Q)[m_z(P), 1] + 1 + h(N)(m_x(P) + m_x(Q)[m_z(P), 1]) \\
&= h((\underline{\lambda}z.P) Q) + h(N)m_x((\underline{\lambda}z.P) Q)
\end{aligned}$$

3. $M \equiv P Q$ where $P \not\equiv \underline{\lambda}y.R$. Then, by the induction hypothesis,

$$\begin{aligned}
h(P^* Q^*) &= h(P^*) + h(Q^*) \\
&= h(P) + h(N)m_x(P) + h(Q) + h(N)m_x(Q) \\
&= h(P Q) + h(N)m_x(P Q)
\end{aligned}$$

4. $M \equiv \underline{\lambda}y.P$. Similar to Case 3. □

Lemma 2.4. *Suppose that $M \rightarrow_{\underline{\beta}} N$. Then*

- (i) $m_x(M) \leq m_x(N)$;
(ii) $h(M) \leq h(N) + 1$.

Proof. (i) is by induction on $M \rightarrow_{\underline{\beta}} N$.

1. $M \equiv (\underline{\lambda}y.P) Q \rightarrow_{\underline{\beta}} P\{y := Q\} \equiv N$. By Lemma 2.3,

$$\begin{aligned}
m_x((\underline{\lambda}y.P) Q) &= m_x(P) + m_x(Q)[m_y(P), 1] \\
&\leq m_x(P) + m_x(Q)m_y(P) \\
&= m_x(P\{y := Q\})
\end{aligned}$$

2. $M \equiv (\underline{\lambda}y.P) Q \rightarrow_{\underline{\beta}} (\underline{\lambda}y.P') Q' \equiv N$, where $P \rightarrow_{\underline{\beta}} P'$ and $Q \equiv Q'$, or vice versa. By the induction hypothesis,

$$\begin{aligned}
m_x((\underline{\lambda}y.P) Q) &= m_x(P) + m_x(Q)[m_y(P), 1] \\
&\leq m_x(P') + m_x(Q')[m_y(P'), 1] \\
&= m_x((\underline{\lambda}y.P') Q')
\end{aligned}$$

3. $M \equiv P Q \rightarrow_{\underline{\beta}} P' Q' \equiv N$, where $P \not\equiv \underline{\lambda}y.R$, and where $P \rightarrow_{\underline{\beta}} P'$ and $Q \equiv Q'$, or vice versa. Similar to Case 2.

4. $M \equiv \underline{\lambda}y.P \rightarrow_{\underline{\beta}} \underline{\lambda}y.P' \equiv N$, where $P \rightarrow_{\underline{\beta}} P'$. Similar to Case 2.

This concludes (i); (ii) is also by induction on $M \rightarrow_{\underline{\beta}} N$.

1. $M \equiv (\underline{\lambda}y.P) Q \rightarrow_{\underline{\beta}} P\{y := Q\} \equiv N$. By Lemma 2.3

$$\begin{aligned}
h((\underline{\lambda}y.P) Q) &= h(P) + h(Q)[m_y(P), 1] + 1 \\
&\leq h(P) + h(Q)m_y(P) + 1 \\
&= h(P\{y := Q\}) + 1
\end{aligned}$$

2. $M \equiv (\underline{\lambda}y.P) Q \rightarrow_{\underline{\beta}} (\underline{\lambda}y.P') Q' \equiv N$, where $P \rightarrow_{\underline{\beta}} P'$ and $Q \equiv Q'$, or vice versa. By the induction hypothesis and (i),

$$\begin{aligned}
h((\underline{\lambda}y.P) Q) &= h(P) + h(Q)[m_y(P), 1] + 1 \\
&\leq h(P') + h(Q')[m_y(P'), 1] + 2 \\
&= h((\underline{\lambda}y.P') Q') + 1
\end{aligned}$$

3. $M \equiv P Q \rightarrow_{\underline{\beta}} P' Q' \equiv N$, where $P \not\equiv \underline{\lambda}y.R$, and where $P \rightarrow_{\underline{\beta}} P'$ and $Q \equiv Q'$, or vice versa. Similar to Case 2.
4. $M \equiv \underline{\lambda}y.P \rightarrow_{\underline{\beta}} \underline{\lambda}y.P' \equiv N$, where $P \rightarrow_{\underline{\beta}} P'$. Similar to Case 2. □

Corollary 2.5. *For all $M \in \underline{\Lambda}_K$: $h(M) \leq s_{\underline{\beta}}(M)$.*

Proof. By induction on $h(M)$.

1. $h(M) = 0$. Then $M \in \text{NF}_{\underline{\beta}}$, and then $s_{\underline{\beta}}(M) = 0$.
2. $h(M) \neq 0$. Then $M \notin \text{NF}_{\underline{\beta}}$. Let $M \rightarrow_{\underline{\beta}} N$ be such that $s_{\underline{\beta}}(M) = s_{\underline{\beta}}(N) + 1$. By Lemma 2.4(ii) and the induction hypothesis,

$$\begin{aligned} h(M) &\leq h(N) + 1 \\ &\leq s_{\underline{\beta}}(N) + 1 \\ &= s_{\underline{\beta}}(M) \end{aligned} \quad \square$$

Lemma 2.6. *If $h(M) \neq 0$ then $M \rightarrow_{\underline{\beta}} H(M)$ and $h(M) = h(H(M)) + 1$.*

Proof. By induction on M . Assume $h(M) \neq 0$.

1. $M \equiv x$. This case is impossible since $h(x) = 0$.
2. $M \equiv (\underline{\lambda}y.P) Q$.
 - 2.1. $[m_y(P), 1] = 1$ and $Q \notin \text{NF}_{\underline{\beta}}$. By the induction hypothesis,

$$\begin{aligned} h((\underline{\lambda}y.P) Q) &= h(P) + h(Q)[m_y(P), 1] + 1 \\ &= h(P) + h(Q) + 1 \\ &= h(P) + h(H(Q)) + 2 \\ &= h(P) + h(H(Q))[m_y(P), 1] + 2 \\ &= h((\underline{\lambda}y.P) H(Q)) + 1 \\ &= h(H((\underline{\lambda}y.P) Q)) + 1 \end{aligned}$$

- 2.2. $[m_y(P), 1] \neq 1$ or $Q \in \text{NF}_{\underline{\beta}}$. By Lemma 2.3

$$\begin{aligned} h((\underline{\lambda}y.P) Q) &= h(P) + h(Q)[m_y(P), 1] + 1 \\ &= h(P) + h(Q)m_y(P) + 1 \\ &= h(P\{y := Q\}) + 1 \end{aligned}$$

3. $M \equiv \underline{\lambda}y.P$. Then, by the induction hypothesis,

$$\begin{aligned} h(\underline{\lambda}y.P) &= h(P) \\ &= h(H(P)) + 1 \\ &= h(\underline{\lambda}y.H(P)) + 1 \\ &= h(H(\underline{\lambda}y.P)) + 1 \end{aligned}$$

4. $M \equiv P Q$. Similar to Case 3. □

Corollary 2.7. *For all $M \in \underline{\Lambda}_K$: $h(M) = L_H(M)$.*

Proof. By induction on $h(M)$.

1. $h(M) = 0$. Then $M \in \text{NF}_{\underline{\beta}}$, and then $L_H(M) = 0$.
2. $h(M) \neq 0$. Then $M \notin \text{NF}_{\underline{\beta}}$, and then by Lemma 2.6 and the induction hypothesis,

$$\begin{aligned} h(M) &= h(H(M)) + 1 \\ &= L_H(H(M)) + 1 \\ &= L_H(M) \end{aligned} \quad \square$$

Corollary 2.8. *For all $M \in \underline{\Lambda}_K$: $h(M) = s_{\underline{\beta}}(M) = L_H(M)$.*

Proof. Let $M \in \underline{\Lambda}_K$. Obviously, $s_{\underline{\beta}}(M) \leq L_H(M)$. By Corollary 2.5 and 2.7,

$$s_{\underline{\beta}}(M) \leq L_H(M) = h(M) \leq s_{\underline{\beta}}(M) \quad \square$$

3. RELATION TO KHASIDASHVILI'S TECHNIQUE

Khasidashvili [4] calls a redex Δ in M *essential*, notation $E(\Delta, M)$, if every complete development of M must reduce Δ (or a residual of Δ). He shows that any strategy which reduces in each step an inner-most essential redex yields shortest complete developments, and he gives a formula for the length of such developments: the number of essential redexes in the initial term. He also gives an algorithm to decide whether a redex in a term is essential; this makes the above strategy and formula effective, but the algorithm is—in our opinion—somewhat involved. The algorithm can be simpler formulated in terms of the map m_y as follows:

$$\begin{aligned} E(\Delta, (\underline{\lambda}y.P) Q) &\Leftrightarrow \Delta \equiv (\lambda y.P) Q \text{ or } E(\Delta, P) \text{ or } [E(\Delta, Q) \ \& \ m_y(P) > 0] \\ E(\Delta, P Q) &\Leftrightarrow E(\Delta, P) \text{ or } E(\Delta, Q) \\ E(\Delta, \lambda y.P) &\Leftrightarrow E(\Delta, P) \end{aligned}$$

In this terminology, the map h counts the number of essential redexes in a term, and H reduces *some* essential redex whose argument does not contain another essential redex.

4. RELATION TO DE VRIJER'S TECHNIQUE

De Vrijer [7] studies the following maps n_x , g , and G , which arise from m_x , h , and H by replacing all minimum operators $[\bullet, \bullet]$ by maximum operators $[\bullet, \bullet]$; intuitively this makes sense since we now consider longest instead of shortest developments.

(i) For all $x \in V$ define $n_x : \underline{\Lambda}_K \rightarrow \mathbb{N}$ by:

$$\begin{aligned} n_x(x) &= 1 \\ n_x(y) &= 0 && \text{if } x \neq y \\ n_x((\underline{\lambda}y.P) Q) &= n_x(P) + n_x(Q)[n_y(P), 1] \\ n_x(P Q) &= n_x(P) + n_x(Q) && \text{if } P \neq \underline{\lambda}y.R \\ n_x(\lambda y.P) &= n_x(P) \end{aligned}$$

(ii) Define $g : \underline{\Lambda}_K \rightarrow \mathbb{N}$ by:

$$\begin{aligned} g(x) &= 0 \\ g((\underline{\lambda}y.P) Q) &= g(P) + g(Q)[n_y(P), 1] + 1 \\ g(P Q) &= g(P) + g(Q) && \text{if } P \neq \underline{\lambda}y.R \\ g(\lambda y.P) &= g(P) \end{aligned}$$

(iii) Define $G : \underline{\Lambda}_K \rightarrow \underline{\Lambda}_K$ by:

$$\begin{aligned} G(x) &= x \\ G((\underline{\lambda}y.P) Q) &= \begin{cases} (\underline{\lambda}y.P) G(Q) & \text{if } [n_y(P), 1] = 1 \ \& \ Q \notin \text{NF}_{\underline{\beta}} \\ P\{y := Q\} & \text{otherwise} \end{cases} \\ G(P Q) &= \begin{cases} G(P) Q & \text{if } P \neq \underline{\lambda}y.R \ \& \ P \notin \text{NF}_{\underline{\beta}} \\ P G(Q) & \text{if } P \neq \underline{\lambda}y.R \ \& \ P \in \text{NF}_{\underline{\beta}} \end{cases} \\ G(\lambda y.P) &= \lambda y.G(P) \end{aligned}$$

(iv) Let $L_G(M)$ be the length of the complete development

$$M \rightarrow_{\underline{\beta}} G(M) \rightarrow_{\underline{\beta}} G(G(M)) \rightarrow_{\underline{\beta}} \dots$$

De Vrijer proves that $M \rightarrow_{\underline{\beta}} G(M) \rightarrow_{\underline{\beta}} G(G(M)) \rightarrow_{\underline{\beta}} \dots$ is a longest complete development from M , and that $g(M)$ is the length of this development. This is expressed by the equations: $L_G(M) = l_{\underline{\beta}}(M) = g(M)$. The finite developments theorem is an immediate corollary.

The proof of these equations can be carried out *exactly* as in 2.2–2.8 by replacing $s_{\underline{\beta}}$, $[\bullet, \bullet]$, \leq , m_x , h , and L_H by $l_{\underline{\beta}}$, $[\bullet, \bullet]$, \geq , n_x , g , and L_G , respectively! This works because the properties used in 2.2–2.8 involving $s_{\underline{\beta}}$, m_x , etc. are invariant under the transformation, as the reader is encouraged to check.³ For instance, the property $[m, n] \leq m$ becomes $[m, n] \geq m$.

5. DISCUSSION

Although the general notions of longest and shortest complete β -reduction sequences are intuitively “opposite,” they are, technically speaking, very different. For instance, there is an effective reduction strategy that computes longest complete β -reduction sequences (see [6] among others), but no effective reduction strategy that computes shortest complete β -reduction sequences [1]. In contrast, the above shows that one can effectively compute both shortest and longest complete developments, and the proofs reveal a duality between the two concepts. It is natural to ask why the duality does not carry over to the general case of β -reduction.

The difference between the minimal strategy H and the maximal strategy G is revealed on terms of form $(\lambda y.P) Q$ where $Q \notin \text{NF}_{\underline{\beta}}$. The rationale behind the minimal strategy is that if all reductions of $(\lambda y.P) Q$ to $\underline{\beta}$ -normal form must reduce inside at least one residual of Q , then it is best to perform reductions in Q first, to avoid proliferation. This is decidable for developments, but undecidable for β -reduction [2].

The rationale behind the maximal strategy is that if all reductions of $(\lambda y.P) Q$ to $\underline{\beta}$ -normal form may reduce inside at most one residual of Q , then it is best to perform reductions in Q first, to avoid erasing. An equivalent technique, used by de Vrijer [7], is to test whether reducing $(\lambda y.P) Q$ one step would delete Q , and if so reduce Q to normal form first. This is decidable for developments as well as for β -reduction.

From the point of view of efficiency, a minimal strategy is clearly better than a maximal strategy. It is a remarkable fact that in general β -reductions we can effectively do the worst possible job, but not the best possible job.⁴

REFERENCES

- [1] H.P. Barendregt. *The Lambda Calculus: Its Syntax and Semantics*. North-Holland, second, revised edition, 1984.
- [2] H.P. Barendregt, J.R. Kennaway, J.W. Klop, and M.R. Sleep. Needed reduction and spine strategies for the lambda calculus. *Information and Computation*, 75(3):191–231, 1987.

³To obtain this result, a small change has been made to G as compared to de Vrijer’s formulation; in his formulation the condition $[n_y(P), 1] = 1$ is $n_y(P) = 0$ —see §5.

⁴But see [5] for a technique to establish both longest and shortest reductions, though they may not be effective.

- [3] H.B. Curry and R. Feys. *Combinatory Logic*. North-Holland, 1958.
- [4] Z. Khasidashvili. β -reductions and β -developments with the least number of steps. In P. Martin-Löf and G. Mints, editors, *International Conference on Computer Logic*, volume 417 of *Lecture Notes in Computer Science*, pages 105–111. Springer-Verlag, 1988.
- [5] V. van Oostrom. Random descent. In F. Baader, editor, *Rewriting Techniques and Applications*, volume 4533 of *Lecture Notes in Computer Science*, pages 314–328. Springer-Verlag, 2007.
- [6] M.H. Sørensen. Effective longest and infinite reduction paths in untyped λ -calculi. In H. Kirchner, editor, *Colloquium on Trees in Algebra and Programming*, volume 1059 of *Lecture Notes in Computer Science*, pages 287–301. Springer-Verlag, 1996.
- [7] R.C. de Vrijer. A direct proof of the finite developments theorem. *Journal of Symbolic Logic*, 50:339–343, 1985.