Separability of the Lambda Calculus and Term Rewriting Systems

Department of Computer Science
Kyungsung University

Sugwoo Byun

19 August 2003

Contents

1. Head normal form in the λ-Calculus
2. Böhm trees
3. Separability of the λ-calculus
4. Sequentiality of the λ-calculus
5. Böhm-out transformation technique
6. Lambda Definability of Term Rewriting Systems
7. Separable Systems
8. Encoding of Separable Systems
9. Correctness

Reference:

H. Barendregt. *The Lambda Calculus: Its Syntax and Semantics*. North-Holland. 1984.

Head Normal Form in the λ-calculus

- M is a hnf (*head normal form*) if M has the form

$$M \equiv \lambda x_1 \cdots x_n . y N_1 \cdots N_k$$

  $\Omega \equiv (\lambda x.xx)(\lambda x.xx)$. $\lambda x.Ix\Omega$ has no normal form, but has an hnf $\lambda x.x\Omega$. Each normal form is a hnf.

- **Consistent Theory** Let $\mathfrak{T}$ be a formal theory with equations as formulas. Then $\mathfrak{T}$ is *consistent* if $\mathfrak{T}$ does not prove every closed equation.

- $\mathfrak{T} = \{M = N \mid M, N \in \Lambda_K \text{ without normal form}\}$. Then $\mathfrak{T}$ is not consistent. Proof. Let $M \equiv \lambda x.xK\Omega$, $N \equiv \lambda x.xS\Omega$. Then $M = N \in \mathfrak{T}$. Hence

$$\mathfrak{T} \vdash K = MK = NK = S$$

- In λK-calculus

$$\text{meaningless} \Leftrightarrow \text{no hnf} \Rightarrow \text{no nf}$$

- In λI-calculus

$$\text{meaningless} \Leftrightarrow \text{no nf}$$

- Head Reduction

  If M is of the form

$$M \equiv \lambda x_1 \cdots x_n . \underline{(\lambda x.M_0)M_1} \cdots M_k \qquad n \geq 0, k \geq 1$$

  $(\lambda x.M_0)M_1$ is called *head redex*.

  We write $M \xrightarrow[h]{} N$ if a head redex is chosen to be reduced. If the selected redex is not a head redex, it is an *internal redex* and its reduction is written as $M \xrightarrow[i]{} N$.

- Standard Reduction. Let

$$\sigma : M_0 \xrightarrow{\Delta_0} M_1 \xrightarrow{\Delta_1} M_2 \xrightarrow{\Delta_2} \cdots$$

  be a reduction. σ is called a *standard reduction* if $\forall i \; \forall j < i \; \Delta_i$ is not a 'residual' of a redex to the left of Δ_j. We write $M \xrightarrow[s]{} N$ if there is a standard reduction $\sigma : M \longrightarrow N$.

- Example.

$$\lambda a.(\lambda b.\underline{(\lambda c.c)b}b)d \longrightarrow \lambda a.\underline{(\lambda b.bb)d} \longrightarrow \lambda a.dd. \qquad \text{not standard}$$

$$\lambda a.\underline{(\lambda b.(\lambda c.c)bb)d} \longrightarrow \lambda a.\underline{(\lambda c.c)d}d \longrightarrow \lambda a.dd. \qquad \text{standard}$$

- **Standardization Theorem** If $M \longrightarrow^* N$, then $M \underset{s}{\longrightarrow}^* N$ such that

$$\exists Z \quad M \underset{h}{\longrightarrow}^* Z \underset{i}{\longrightarrow}^* N$$

Böhm Trees

- $BT(M)$, the Böhm tree of $M \equiv \lambda\vec{x}.yM_1 \cdots M_k$
  - if M is undefined, $BT(M) = \bot$
  - if $M \equiv$

$$\begin{tikzcd}[column sep=small] & \lambda\vec{x}.y \arrow[dl, no head] \arrow[dr, no head] & \\ BT(M_1) & \cdots & BT(M_k) \end{tikzcd}$$

- Example
  - $BT(S) =$

$$\begin{tikzcd}[column sep=small] & \lambda abc.a \arrow[dl, no head] \arrow[dr, no head] & \\ c & \cdots & b \arrow[d, no head] \\ & & c \end{tikzcd}$$

  - $BT(Sa\Omega) =$

$$\begin{tikzcd}[column sep=small] & \lambda c.a \arrow[dl, no head] \arrow[dr, no head] & \\ c & \cdots & \bot \end{tikzcd} \qquad \lambda c.ac(\Omega c)$$

– $$BT(Y) \;=\; \begin{array}{l} \lambda f.f \\ \quad\;\; | \\ \quad\;\; f \\ \quad\;\; | \\ \quad\;\; \vdots \end{array}$$

$$Y \;\equiv\; \lambda f.\omega_f \omega_f \qquad\qquad \omega_f \;\equiv\; \lambda x.f(xx) \qquad\qquad \omega_f \omega_f \;=\; f(\omega_f \omega_f)$$

Separability of the λ-calculus

- Separability. Let $\mathfrak{T} = \{M_1, \cdots, M_p\}$ be a set of λ-terms.
  1. If $\mathfrak{T} \subseteq \Lambda^0$, then $\mathfrak{T}$ is called *separable* if
     $\forall N_1 \cdots N_p \in \Lambda \qquad \exists F \in \Lambda$
     $$\begin{array}{ccc} FM_1 & = & N_1 \\ FM_2 & = & N_2 \\ \cdots & & \cdots \\ FM_p & = & N_p \end{array}$$
  2. If $\mathfrak{T} \subseteq \Lambda$, then $\mathfrak{T}$ is *separable* if its closure $\lambda\vec{x}.\mathfrak{T} = \{\lambda\vec{x}.M_1, \cdots, \lambda\vec{x}.M_p\}$ is separable.
- Distinct Terms. $\mathfrak{T}$ is *distinct* if $\mathfrak{T}$ consists of one element or some $\alpha \in \mathrm{Seq}$ is 'useful' for $\mathfrak{T}$ and the $\sim_\alpha$ equivalence classes of elements of $\mathfrak{T}$ are all distinct.
- Let $M \equiv \lambda x_1 \cdots x_n.yM_1 \cdots M_m$ and $N \equiv \lambda x_1 \cdots x_{n'}.y'M_1 \cdots M_{m'}$. Then
  $$M \sim N \text{ iff } y \equiv y' \text{ and } n - m = n' - m'$$

$$\begin{array}{ccc} \lambda x.y & \not\sim & \lambda y.y \\ | & & | \\ M & & M \end{array} \qquad\qquad \begin{array}{ccc} \lambda x.x & \sim & \lambda yz.y \\ | & & / \quad \backslash \\ M & & M \quad N \end{array}$$

- **Separability Theorem**.

$$\mathfrak{T} \text{ is separable} \Leftrightarrow \mathfrak{T} \text{ is distinct}$$

Being given a constructive proof, we can find a λ-term F in the separable equations.

Sequentiality of the λ-calculus

- The λ-calculus is *sequential*. "parallel-or" functions cannot be definable in the λ-calculus (by Plotkin 1997).

$$\begin{array}{lcl} por\ x\ T & = & T \\ por\ T\ x & = & T \end{array}$$

- The computation of a definable λ-calculus function is sequential rather than parallel (by Berry 1978.)

- No $F \in \Lambda$ such that

$$\begin{array}{lcll} FMN & = & I & \text{if } M \text{ or } N \text{ is solvable} \\ & = & \text{unsolvable} & \text{else} \end{array}$$

  Such an F is clearly parallel computable (simultaneously try to find the hnf of M and N; if you find one, then give output I, else give no output).

Böhm-out Transformation

Primitive functions

- $<M_0, \ldots, M_n> \equiv \lambda z.zM_0 \cdots M_n$.
- $\mathbf{U}_i^n \equiv \lambda x_1 \ldots x_n.x_i$ (selection).
- $\mathbf{P}_n \equiv \lambda x_1 \ldots x_n. <x_1, \ldots, x_n> = \lambda x_1 \ldots x_n x_{n+1}.x_{n+1} x_1 \ldots x_n$ (permutation).

Algorithm of Böhm-out π_α

$\mathcal{T}$: a set of λ-terms , $\alpha \in BT(M)$

1. π_f – transform $\mathcal{T}$ into λ-free form.

$$(\,)^{\pi_f} = (\,)x_1 \cdots x_n.$$

π_f transformation

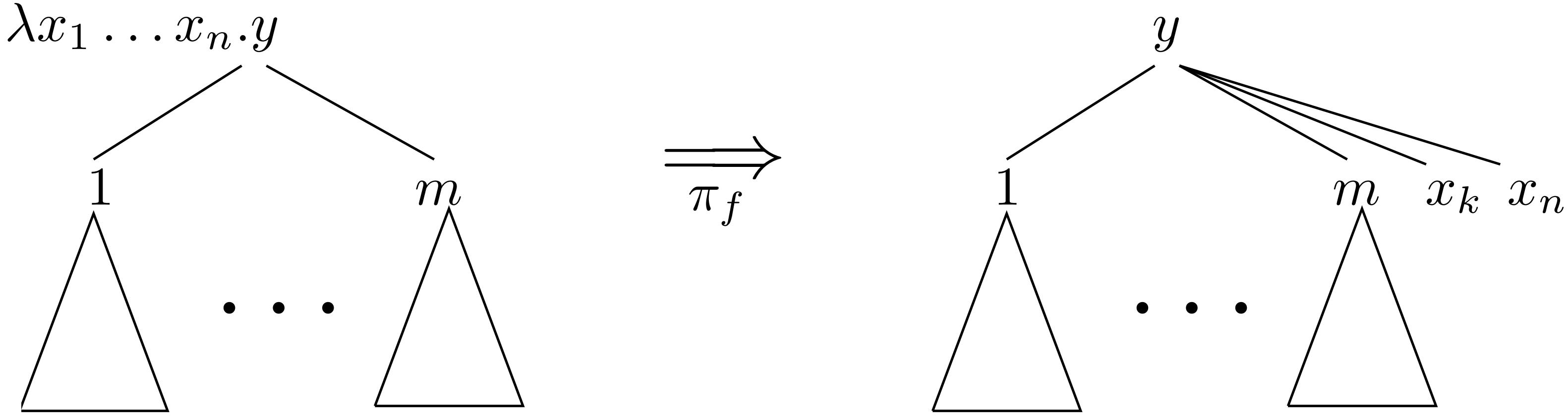


Figure 1: Transformation π_f

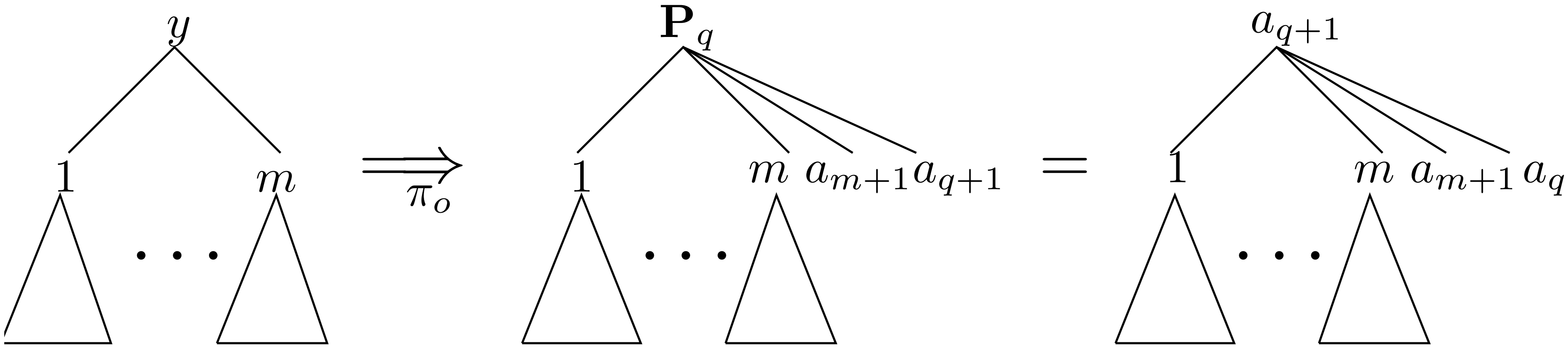
2. π_o – transform $\mathcal{T}^{\pi_f}$ into original form.

$$(\,)^{\pi_o} = (\,)\, a_1 \cdots a_{p+1}\, [y := \mathbf{P}_p],$$

(a) The head variable at the root

π_o transformation

(a) The head variable at the root



(b) Internal node having a head variable

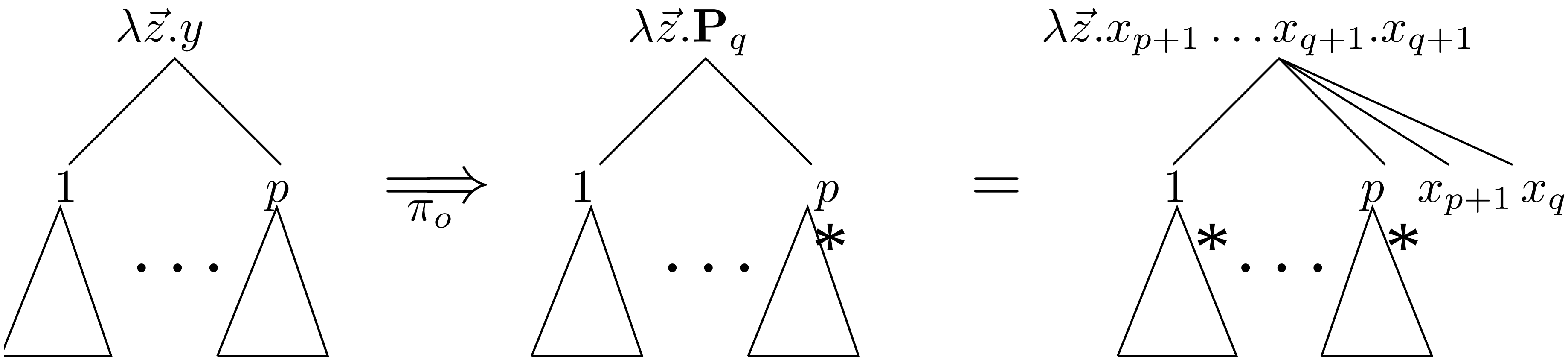


Figure 2: Transformation π_o

3. π_s – select one of following success terms.

$$(\,)^{\pi_s} = (\,)[z := \mathbf{U}_j^n],$$

where $\alpha = j \cdot \alpha'$.

π_s transformation - select a subterm at j

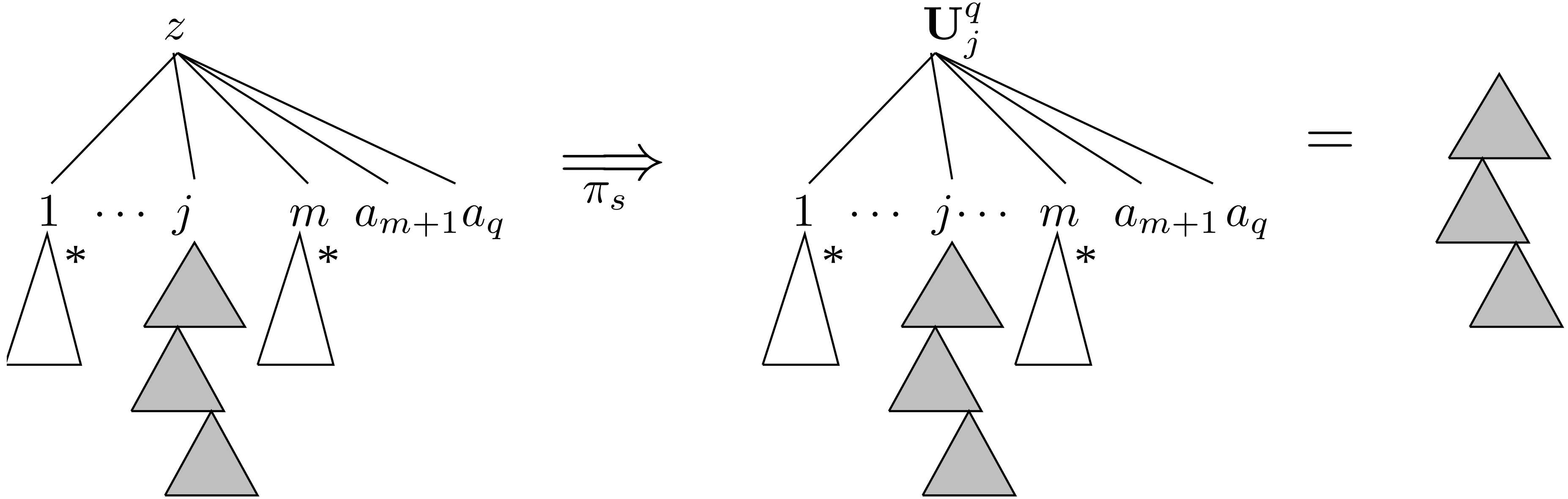


Figure 3: Transformation π_s

4. Repeat the above procedures with $(((\mathcal{T})^{\pi_f})^{\pi_o})^{\pi_s}$ until α' becomes empty.

Lambda Definability of Term Rewriting Systems

- Question [RTA'90]
  "Which rewrite systems can be directly defined in the λ-calculus? One has to find λ-terms representing rewrite system operators such that a rewrite step in TRS translates to a reduction in the λ-calculus."

- Main Idea

  1. Define a *separable system*:

$$\begin{array}{ll} f(a_{11}, a_{12}, \ldots, a_{1n}) \rightarrow b_1 \\ f(a_{21}, a_{22}, \ldots, a_{2n}) \rightarrow b_2 \\ \ldots & \ldots \\ f(a_{m1}, a_{m2}, \ldots, a_{mn}) \rightarrow b_m \end{array}$$

     Let $\mathcal{P}_f = \{ \ (a_{11}, a_{12}, \ldots, a_{1n}), \ (a_{21}, a_{22}, \ldots, a_{2n}), \ \cdots, \ (a_{m1}, a_{m2}, \ldots, a_{mn}) \ \}$.
     The f-rules are separable if $\mathcal{P}_f$ is 'distinct'. Each element of $\mathcal{P}_f$ is in a Böhm tree.

2. Define a homogeneous function $\phi(\mathcal{P}_f)$ such that $\phi(\mathcal{P}_f)$ is a set of distinct λ-terms. Then, by Böhm's separability theorem, $\phi(f)$ is obtained such that $s \to t$ implies $\phi(s) \to^* \phi(t)$.

Separable Systems

E.g.

$$F(x, A, B, C, S(D)) \rightarrow 1$$
$$F(B, x, A, C, S(D)) \rightarrow 2$$
$$F(A, B, x, C, S(E)) \rightarrow 3$$

is separable. Each separable system has a separation tree.

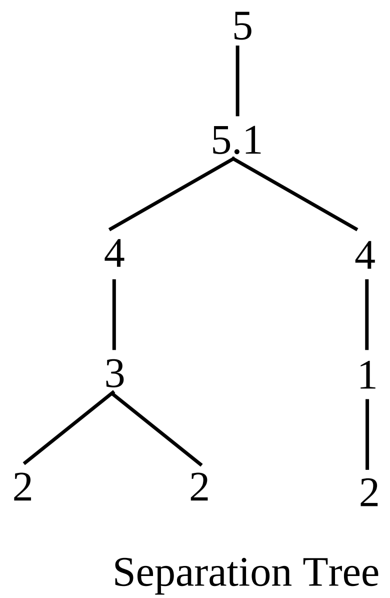


Figure 4: Separation Tree

Encoding of Separable Systems

Example

$$A(0, x) \to x$$
$$A(S(x), y) \to S(A(x, y))$$

A separation tree $U_T \ = \ 1$
Assume $\phi_c(0) \ = <\underline{0}>$, $\phi_c(S) \ = \ \lambda x. <\underline{2}, x>$.
Then, $\phi_p(0, y) \ = <<0>, \square>$, $\phi_p(S(x), y) \ = <<\underline{2}, \square>, \square>$, and $\alpha \ = \ 1 \cdot 1$.

The whole encoding $\pi = \pi_1 \cdot \pi_{\alpha \cdot \alpha'}$.

- $\pi_{\alpha \cdot \alpha'}$: Böhm-out terms at $\alpha \cdot \alpha'$
- π_1 : pass terms matches with variables at LHS to ones at RHS.

Then,
$\phi(A) \ = \mathbf{Y}\ (\lambda a x . x\, \mathbf{U}_1^2\, \mathbf{P}_2\, \mathbf{U}_1^3\, \mathbf{U}_1^2\, (\lambda b.\ < \underline{2},\ (a\ < (x\, \mathbf{U}_1^2\, \mathbf{U}_2^2),\ (x\, \mathbf{U}_2^2) >) >)\ (x\, \mathbf{U}_2^2))$

Reductions $A(S(0),\ 0)\ \rightarrow\ S(A(0,\ 0))\ \rightarrow\ S(0)$ are simulated as follows.

$\phi(A(S(0),\ 0))\ =\ \phi(A)\,\phi_p(S(0),\ 0)\ =\ \phi(A)\ <\!\phi(S(0)),\ \phi(0)\!>$
$=\ \phi(A)\ <<\underline{2},\ <\underline{0}>>,\ <\underline{0}>>$
$\equiv \mathbf{Y}\ (\lambda ax.x\,\mathbf{U}^2_1\,\mathbf{P}_2\,\mathbf{U}^3_1\,\mathbf{U}^2_1\,(\lambda b.\ <\underline{2},\ (a\ <(x\,\mathbf{U}^2_1\,\mathbf{U}^2_2),\ (x\,\mathbf{U}^2_2)>\,)>)\,(x\,\mathbf{U}^2_2))\ \ <<\underline{2},\ <$
$\underline{0}>>,\ <\underline{0}>>$
$\rightarrow^*\ <\underline{2},\ (\phi(A)\ <<\underline{0}>,\ <\underline{0}>>)>$
$\rightarrow^*<\underline{2},\ <\underline{0}>>\qquad\qquad (=\ \phi(S(0))\ )$

Correctness

1. $s \to^* t \Longrightarrow \phi(s) \to^* \phi(t)$.

2. Let s include no operator normal term. Then, $\phi(s) \to^* \phi(t) \Longrightarrow s \to^* t$.

3. If s has a normal form, then $\phi(s)$ has a β-normal form.