# *Lambda-Cáculo*
# (Aula 6)

Ruy de Queiroz & Anjolina de Oliveira

Centro de Informática, UFPE

2007.2

# Conteúdo

# Representando as funções recursivas
## Funções parciais

### Definição (Função parcial)

*Uma função* $\phi$ *de um subconjunto de* $\mathbb{N}^k$ *para* $\mathbb{N}$ *será chamada de* propriamente parcial *sse* $\phi(m_1, \ldots, m_k)$ *for indefinida para algum* $m_1, \ldots, m_k$*, e* total *caso contrário. Uma* função parcial *pode ser propriamente parcial ou total. A classe de todas as* funções recursivas parciais *(funções parciais que são recursivas) serão definidas adiante. Uma* função recursiva total *será uma função recursiva parcial que é total.*
*Para quaisquer termos X e Y, usaremos a abreviação*

$$\begin{aligned} X^n Y &\equiv \underbrace{X(X(\ldots(X}_{n\ X\text{'s}}\ Y)\ldots)) \qquad \textit{for } n \geq 1 \\ X^0 Y &\equiv Y. \end{aligned}$$

# Representando as funções recursivas
## Numerais de Church

### Definição (Os numerais de Church (1941))

*Para cada número natural n, o* numeral de Church representando *n é (no λ-cálculo) o termo*

$$\overline{n} \equiv \lambda xy.x^n y,$$

*e (em LC) o termo*

$$\overline{n} \equiv (\mathsf{SB})^n(\mathsf{KI})$$

*Às vezes* $\overline{n}$ *é chamado* $\mathsf{Z}_n$*. Tanto no λ-cálculo quanto na LC ele tem a propriedade de que para todos os termos* $F, X$

*(4.3)* $$\overline{n}FX \triangleright_{\beta,w} F^n X.$$

# Representando as funções recursivas

## Definindo funções sobre números naturais

### Definição

*Seja* $\phi$ *uma função parcial de números naturais de n-argumentos. Dizemos que um* $\lambda$*-termo* $X$ $\lambda$-define $\phi$*, ou um LC-termo* $X$ combinatorialmente define $\phi$*, sse para todo* $m_1, \ldots, m_n$,

$$X\overline{m_1}\ldots\overline{m_n} =_{\beta,w} \overline{\phi(m_1,\ldots,m_n)}$$

*quando* $\phi(m_1,\ldots,m_n)$ *for definida, e* $X\overline{m}_1\ldots\overline{m}_n$ *não tem forma normal caso contrário.*

# Representando as funções recursivas
## Definindo funções sobre números naturais

**Representando as funções recursivas:** Provaremos que *toda função recursiva parcial pode ser* $\lambda$- *e combinatorialmente definida.*
A recíproca, i.e. *toda função* $\lambda$- *e combinatorialmente definida é uma função recursiva parcial*, é provada via técnicas padrão de numeração de Gödel.

# Representando as funções recursivas
## Definindo funções sobre números naturais

**Lema**

*Para $\lambda$-termos e $=_\beta$, ou LC-termos e $=_w$: toda função recursiva primitiva $\phi$ pode ser definida por um combinador $\overline{\phi}$.*

# Representando as funções recursivas

## Prova do Lema

### Demonstração.

A classe das *funções recursivas primitivas* é definida por indução (Kleene 1952) da seguinte forma:
(i) (sucessor) A função sucessor $\sigma$ é recursiva primitiva;
(ii) (zero) 0 é uma função recursiva primitiva de 0-argumentos;
(iii) (projeção) para cada $n \geq 1$ e $k \leq n$ a função de projeção
$\phi(m_1, \ldots, m_n) = m_k \qquad$ (para todo $m_1, \ldots, m_n$) é recursiva primitiva;
(iv) (composição) se $\psi$, $\chi_1, \ldots, \chi_p$ forem recursivas primitivas, $n \geq 0$, e $\phi(m_1, \ldots, m_n) = \psi(\chi_1(m_1, \ldots, m_n), \ldots, \chi_p(m_1, \ldots, m_n))$, então $\phi$ é recursiva primitiva;
(v) (definição por recursão sobre $\mathbb{N}$) se $\psi$ e $\chi$ forem recursivas primitivas, $n \geq 0$, e

$$\left\{ \begin{array}{rcl} \phi(0, m_1, \ldots, m_n) & = & \psi(m_1, \ldots, m_n), \\ \phi(k+1, m_1, \ldots, m_n) & = & \chi(k, \phi(k, m_1, \ldots, m_n), m_1, \ldots, m_n), \end{array} \right.$$

então $\phi$ é recursiva primitiva.

$\square$

# Representando as funções recursivas
## Prova do Lema (cont.)

### Demonstração.

(Cont.) O termo $\overline{\phi}$ é definido da seguinte maneira:
(i) (sucessor) $\overline{\sigma} \equiv \lambda uxy.x(uxy)$, que em LC é SB;
(ii) (zero) $\overline{0} \equiv \lambda xy.y$, que em LC é KI;
(iii) (projeção) $\overline{\phi} \equiv \lambda x_1 \ldots x_n.x_k$;
(iv) (composição) dados $\overline{\psi}, \overline{\chi}_1, \ldots, \overline{\chi}_p$, definir $\psi, \chi_1, \ldots, \chi_p$ respectivamente, faça
$\overline{\phi} \equiv \lambda x_1 \ldots x_n.(\overline{\psi}(\overline{\chi}_1 x_1 \ldots x_n) \ldots (\overline{\chi}_p x_1 \ldots x_n))$
(v) (definição por recursão sobre $\mathbb{N}$) dados $\overline{\psi}$ e $\overline{\chi}$ definir $\psi$ e $\chi$ respectivamente, faça
$\overline{\phi} \equiv \lambda ux_1 \ldots x_n.\mathrm{R}(\overline{\psi}x_1 \ldots x_n)(\lambda uv.\overline{\chi}uvx_1 \ldots x_n)u$,
onde R é um combinador chamado de *combinador de recursão primitiva* a ser construído mais adiante, tendo a propriedade de que para todo $X, Y, k$

$$(4.6) \qquad \begin{cases} \mathrm{R}XY\overline{0} & =_{\beta,w} \quad X, \\ \mathrm{R}XY\overline{k+1} & =_{\beta,w} \quad Y\overline{k}(\mathrm{R}XY\overline{k}). \end{cases}$$

□

# Representando as funções recursivas

## Prova do Lema (cont.)

### Demonstração.

Tendo tal combinador R, podemos definir a função $\phi$ por meio do termo $\overline{\phi}$, pois

$$\left\{\begin{array}{rcl} \overline{\phi}\overline{0}x_1\ldots x_n & =_{\beta,w} & \mathrm{R}(\overline{\psi}x_1\ldots x_n)(\lambda uv.\overline{\chi}uvx_1\ldots x_n)\overline{0} \\ & =_{\beta,w} & \overline{\psi}x_1\ldots x_n \\ & & \text{(por (4.6));} \\ \overline{\phi}(\overline{k+1})x_1\ldots x_n & =_{\beta,w} & \mathrm{R}(\overline{\psi}x_1\ldots x_n)(\lambda uv.\overline{\chi}uvx_1\ldots x_n)(\overline{k+1}) \\ & =_{\beta,w} & (\lambda uv.\overline{\chi}uvx_1\ldots x_n)\overline{k}(\mathrm{R}(\overline{\psi}x_1\ldots x_n)(\lambda uv.\overline{\chi}uvx_1\ldots x_n)\overline{k}) \\ & & \text{por (4.6)} \\ & =_{\beta,w} & (\lambda uv.\overline{\chi}uvx_1\ldots x_n)\overline{k}(\overline{\phi}\,\overline{k}x_1\ldots x_n) \\ & & \text{pela def. de } \overline{\phi} \\ & =_{\beta,w} & \overline{\chi}\overline{k}(\overline{\phi}\overline{k}x_1\ldots x_n)x_1\ldots x_n. \end{array}\right.$$

□

# Representando as funções recursivas

## Prova do Lema (cont.)

### Demonstração.

**Construindo o combinador** R. Vamos considerar uma função $\phi$ definida por recursão primitiva:

$$\begin{array}{rcl} \phi(0) & = & m, \\ \phi(k+1) & = & \chi(k, \phi(k)). \end{array}$$

Para calcular $\phi(k)$ podemos começar do par ordenado $\langle 0, m \rangle$ e aí iterar $k$ vezes a operação $f$ t.q.
$f(\langle n, x \rangle) = \langle n+1, \chi(n, x) \rangle$,
e finalmente tomar o segundo componente do último par produzido por esse processo. □

# Representando as funções recursivas
## Lidando com a noção de par ordenado

### Demonstração.

Vamos então definir um combinador para lidar com a noção de par ordenado. Uma possível solução é

(4.7) $$\mathsf{D} \equiv \lambda xyz.z(\mathsf{K}y)x.$$

que tem a seguinte propriedade:

(4.8) $$\begin{cases} \mathsf{D}XY\overline{0} & =_{\beta,w} \quad X, \\ \mathsf{D}XY\overline{k+1} & =_{\beta,w} \quad Y. \end{cases}$$

ou a propriedade de um operador ‘condicional’:

(4.9) $$(\text{Se } Z = 0 \text{ então } X \text{ senão } Y) \equiv \mathsf{D}XYZ.$$

$\square$

# Representando as funções recursivas
## Lidando com a noção de par ordenado

### Demonstração.

Agora, usando tal combinador D, vamos definir:
$Q \equiv \lambda yv.\mathrm{D}(\overline{\sigma}(v\overline{0}))(y(v\overline{0})(v\overline{1}))$,
onde $\overline{\sigma}$ é o combinador 'sucessor' definido em (i) acima. Então, para quaisquer $X$, $Y$, $n$,

$$\begin{array}{rcl} QY(\mathrm{D}\overline{n}X) & =_{\beta,w} & \mathrm{D}(\overline{\sigma}(\mathrm{D}\overline{n}X\overline{0}))(Y(\mathrm{D}\overline{n}X\overline{0})(\mathrm{D}\overline{n}X\overline{1})) \\ & =_{\beta,w} & \mathrm{D}(\overline{\sigma}\overline{n})(Y\overline{n}X) \quad \text{por (4.8)} \\ & =_{\beta,w} & \mathrm{D}(\overline{n+1})(Y\overline{n}X). \end{array}$$

Portanto, $Q$ está fazendo o que desejávamos que $f$ fizesse, i.e.

$$\left\{ \begin{array}{rcll} QY(\mathrm{D}\overline{n}X) & =_{\beta,w} & \mathrm{D}(\overline{n+1})(Y\overline{n}X) & \text{(i)} \\ (QY)^k(\mathrm{D}\overline{0}X) & =_{\beta,w} & \mathrm{D}\overline{k}X_k & \text{(ii)} \end{array} \right.$$

para algum termo $X_k$.
Se $Y$ definisse $\chi$ e $X \equiv \overline{m}$, então $X_k$ corresponderia ao valor de $\phi(k)$ acima. □

# Representando as funções recursivas
## Lidando com a noção de par ordenado

### Demonstração.

Agora, defina

(4.10) $$\mathrm{R}_{\text{Bernays}} \equiv \lambda xyu.u(Qy)(\mathrm{D}\overline{0}x)\overline{1}.$$

Então

$$\begin{array}{llll} \mathrm{R}_{\text{Bernays}} XY\overline{k} & =_{\beta,w} & \overline{k}(QY)(\mathrm{D}\overline{0}X)\overline{1} & \\ & =_{\beta,w} & (QY)^k(\mathrm{D}\overline{0}X)\overline{1} & \text{por } \overline{n}FX \triangleright_{\beta,w} F^nX \text{ (4.3)} \\ & =_{\beta,w} & \mathrm{D}\overline{k}X_k\overline{1} & \text{por (ii) acima} \\ & =_{\beta,w} & X_k & \text{por (4.8)} \qquad \text{(iii)} \end{array}$$

$\square$

# Representando as funções recursivas

## Lidando com a noção de par ordenado

### Demonstração.

Daí podemos verificar que $\mathsf{R}_{\text{Bernays}}$ de fato tem a propriedade que desejávamos (vamos abreviar $\mathsf{R}_{\text{Bernays}}$ por simplesmente R):

$$
\begin{array}{rcll}
\mathsf{R}XY\overline{0} & =_{\beta,w} & (QY)^0(\mathsf{D}\overline{0}X)\overline{1} & \text{por (4.10), (4.3)} \\
 & \equiv & \mathsf{D}\overline{0}X\overline{1} & \text{pela def. de } (QY)^0 \text{ (4.1)} \\
 & =_{\beta,w} & X & \text{por (4.8)} \\
 & & & \\
\mathsf{R}XY(\overline{k+1}) & =_{\beta,w} & (QY)^{k+1}(\mathsf{D}\overline{0}X)\overline{1} & \text{por (4.10), (4.3)} \\
 & =_{\beta,w} & (QY)((QY)^k(\mathsf{D}\overline{0}X))\overline{1} & \text{pela def. de } (QY)^{k+1} \text{ cf. (4.1)} \\
 & =_{\beta,w} & QY(\mathsf{D}\overline{k}X_k)\overline{1} & \text{por (ii) acima} \\
 & =_{\beta,w} & \mathsf{D}(\overline{k+1})(Y\overline{k}X_k)\overline{1} & \text{por (i) acima} \\
 & =_{\beta,w} & Y\overline{k}X_k & \text{por (4.8)} \\
 & =_{\beta,w} & Y\overline{k}(\mathsf{R}XY\overline{k}) & \text{por (iii) acima.}
\end{array}
$$

□

# Representando as funções recursivas
## Lidando com a noção de par ordenado

### Observação

*Para verificar que todas as conversões na prova do Lema 4.5 se verificam para a igualdade fraca* $=_w$ *assim como para a* $=_\beta$ *é simples rotina. Mais adiante (Cap. 6) vai ficar mais claro que basta verificar o seguinte: nunca uma ocorrência de redex é contraída quando ela está no escopo de um* $\lambda$*. Na verdade, com uma exceção, todas as contrações na prova do Lema têm a forma*
$P_1 \dots P_r((\lambda x.M)NQ_1 \dots Q_s) \triangleright P_1 \dots P_r(([N/x]M)Q_1 \dots Q_s)$
$(r, s \geq 0)$*, e essas traduzem para LC como reduções fracas legítimas. O mesmo vai acontecer nas outras provas combinadas para* $=_w$ *e* $=_\beta$ *mais adiante.*
*A única exceção é a* $\beta$*-redução* $\overline{\sigma}\,\overline{n} \triangleright \overline{n+1}$*; mas não precisamos traduzir essa para a LC porque nesse sistema* $\overline{\sigma}\,\overline{n} = \overline{n+1}$ *devido à definição de* $\overline{n}$*.*

# Representando as funções recursivas
## Funções recursivas gerais totais

### Teorema (Kleene)

*Para $\lambda$-termos e $=_\beta$, ou LC-termos e $=_w$: toda função recursiva total $\phi$ pode ser definida por um combinador $\overline{\phi}$ com uma forma normal.*

# Representando as funções recursivas
## Prova do teorema

**Demonstração.**

Pelo Teorema da Forma Normal de Kleene para funções recursivas parciais (1952), para toda função recursiva parcial $\phi$ existem funções recursivas primitivas $\psi$ e $\chi$ tais que $\phi(m_1, \ldots, m_n) = \psi(\mu k[\chi(m_1, \ldots, m_n, k) = 0])$, onde $\mu k[\chi(m_1, \ldots, m_n, k) = 0]$ é o menor $k$, se é que existe algum, para o qual $\chi(m_1, \ldots, m_n, k) = 0$. Se $k$ não existir para alguma $n$-upla $m_1, \ldots, m_n$, então $\phi(m_1, \ldots, m_n)$ é indefinida para aqueles $m_1, \ldots, m_n$. Mas como a hipótese diz que $\phi$ é total, podemos assumir que $k$ sempre existe. $\square$

# Representando as funções recursivas

## Prova do teorema (cont.)

### Continuação.

Uma maneira de computar $\mu k$ é definir um programa $\theta(k)$ que dá como saída $k$ se $\chi(m_1, \dots, m_n, k) = 0$, e segue com $\theta(k+1)$ caso contrário; quando esse programa é inicializado com $k = 0$, produzirá como saída o primeiro $k$ para o qual $\chi(m_1, \dots, m_n, k) = 0$. Vamos definir um análogo formal, que chamamos $H$, de um programa tal como esse. Será a solução da equação recursiva

$Hx_1 \dots x_n y =$ se $\overline{\chi} x_1 \dots x_n y = 0$ então $y$ senão $Hx_1 \dots x_n(\overline{\sigma} y)$.

Uma solução pode ser dada usando-se o combinador Y, mas aí ele não vai ter uma forma normal. Aqui vai uma outra solução que tem uma forma normal. □

# Representando as funções recursivas

## Prova do teorema (cont.)

### Continuação.

Vamos primeiro definir:

(4.16)
$$\begin{cases} T & \equiv & \lambda x.\mathsf{D}\overline{0}(\lambda uv.u(x(\overline{\sigma}v))u(\overline{\sigma}v)), \\ P & \equiv & \lambda xy.Tx(xy)(Tx)y. \end{cases}$$

Então $P$ tem a seguinte propriedade

(4.17)
$$\begin{cases} PXY & =_{\beta,w} & Y & \text{se } XY =_{\beta,w} \overline{0}, \\ PXY & =_{\beta,w} & PX(\overline{\sigma}Y) & \text{se } XY =_{\beta,w} \overline{m+1} \text{ para algum } m. \end{cases}$$

□

# Representando as funções recursivas

## Prova da propriedade (4.17)

### Demonstração.

Sejam $X$, $Y$ termos quaisquer e suponha que $u, v \notin FV(XY)$. Então

$$\begin{array}{rcl} PXY & =_{\beta,w} & TX(XY)(TX)Y \\ & =_{\beta,w} & \mathrm{D}\overline{0}(\lambda uv.u(X(\overline{\sigma}v))u(\overline{\sigma}v))(XY)(TX)Y. \end{array}$$

$$\begin{array}{lrcl} \text{Se} & XY & =_{\beta,w} & \overline{0},\ \text{então por (4.8) (def. de D)}, \\ & PXY & =_{\beta,w} & \overline{0}(TX)Y \\ & & =_{\beta,w} & Y \text{ pois } \overline{0} \equiv \lambda xy.y. \end{array}$$

$$\begin{array}{lrcl} \text{Se} & XY & =_{\beta,w} & \overline{m+1},\ \text{então por (4.8)}, \\ & PXY & =_{\beta,w} & (\lambda uv.u(X(\overline{\sigma}v))u(\overline{\sigma}v))(TX)Y \\ & & =_{\beta,w} & TX(X(\overline{\sigma}Y))(TX)(\overline{\sigma}Y) \\ & & =_{\beta,w} & PX(\overline{\sigma}Y),\ \text{pela def. de } P \text{ onde } \overline{\sigma}Y \text{ substitui } Y. \end{array}$$

□

# Representando as funções recursivas

## Prova do teorema (cont.)

### Demonstração.

Agora defina

$$H \equiv \lambda x_1 \dots x_n y.P(\overline{\chi} x_1 \dots x_n)y.$$

Então, para qualquer $X_1, \dots, X_n$, $Y$, de (4.17) podemos concluir que

$$\begin{array}{lll} HX_1 \dots X_n Y & =_{\beta,w} & P(\overline{\chi} X_1 \dots X_n) Y \\ & =_{\beta,w} & \left\{ \begin{array}{lll} Y & \text{se} & \overline{\chi} X_1 \dots X_n Y =_{\beta,w} \overline{0} \\ HX_1 \dots X_n(\overline{\sigma} Y) & \text{se} & \overline{\chi} X_1 \dots X_n Y =_{\beta,w} \overline{m+1}. \end{array} \right. \end{array}$$

Finalmente, defina

$$\overline{\phi} \equiv \lambda x_1 \dots x_n.\overline{\psi}(Hx_1 \dots x_n \overline{0}).$$

$\overline{\phi}$ define $\phi$, e tem uma forma normal. □

# Representando as funções recursivas
## Funções recursivas parciais

### Teorema

*Para $\lambda$-termos and $=_\beta$, ou LC-termos e $=_w$, toda função recursiva parcial $\phi$ pode ser definida por um combinador $\overline{\phi}$ com uma forma normal.*

# Representando as funções recursivas
## Prova do teorema

### Demonstração.

Como na prova do Teorema 4.15, $\phi$ pode ser expressa como $\phi(m_1, \ldots, m_n) = \psi(\mu k[\chi(m_1, \ldots, m_n, k) = 0])$, mas agora $\mu k$ não está necessariamente definida para toda $n$-upla $m_1, \ldots m_n$. Temos que construir um termo $\overline{\phi}$ de modo que $\overline{\phi}\overline{m_1} \ldots \overline{m_n}$ não tenha forma normal quando não existir $k$ tal que $\chi(m_1, \ldots, m_n, k) = 0$. (A técnica descrita aqui é atribuída a B. Lercher.)

Vamos primeiro tomar $\overline{\phi}$ de 4.15, e chamá-lo '$F$':

$$F \equiv \lambda x_1 \ldots x_n.\overline{\psi}(Hx_1 \ldots x_n\overline{0}).$$

Quando $\phi(m_1, \ldots, m_n)$ estiver definida temos

$$F\overline{m_1} \ldots \overline{m_n} =_{\beta,w} \overline{\phi(m_1, \ldots, m_n)}.$$

Mas ainda precisamos garantir que $F\overline{m_1} \ldots \overline{m_n}$ não tenha forma normal quando $\phi(m_1, \ldots, m_n)$ não estiver definida. $\square$

# Representando as funções recursivas

## Prova do teorema (continuação)

### continuação.

Agora, defina

$$\phi \equiv \lambda x_1 \ldots x_n . P(\overline{\chi} x_1 \ldots x_n)\overline{0}\mathsf{I}(F x_1 \ldots x_n),$$

onde $P$ é definido em

(4.16)
$$\begin{cases} T & \equiv & \lambda x.\mathsf{D}\overline{0}(\lambda uv.u(x(\overline{\sigma}v))u(\overline{\sigma}v)), \\ P & \equiv & \lambda xy.Tx(xy)(Tx)y. \end{cases}$$

Suponha que $m_1, \ldots, m_n$ sejam tais que existe um $k$ tal que

$$\chi(m_1, \ldots, m_n, k) = 0,$$

e assuma que $j$ seja o menor desses $k$'s. Então

$$\begin{array}{llll} \overline{\phi}\overline{m_1} \ldots \overline{m_n} & =_{\beta,w} & \overline{j}\mathsf{I}(F\overline{m_1} \ldots \overline{m_n}) & \text{pela prova de 4.15} \\ & =_{\beta,w} & \mathsf{I}^j(F\overline{m_1} \ldots \overline{m_n}) & \text{por (4.3)} \\ & =_{\beta,w} & F\overline{m_1} \ldots \overline{m_n} & \text{pela def. de I} \\ & =_{\beta,w} & \overline{\phi(m_1, \ldots, m_n)} & \text{pela def. de } F. \end{array}$$

□

# Representando as funções recursivas
## Prova do teorema (continuação)

### continuação.

Suponha agora que $m_1, \ldots, m_n$ sejam tais que não exista $k$ tal que
$$\chi(m_1, \ldots, m_n, k) = 0;$$
então para cada $k$ existe um $p_k$ tal que
$$\chi(m_1, \ldots, m_n, k) = p_k + 1.$$
(Obs.: $\chi$ é total pois é recursiva primitiva.) Seja
$$X \equiv \overline{\chi}\,\overline{m_1} \ldots \overline{m_n}, \qquad G \equiv F\overline{m_1} \ldots \overline{m_n}.$$
Então, para cada $k$, $X\overline{k} =_{\beta,w} \overline{p_k + 1}$. Além do mais, pelo teorema de Church–Rosser temos
$$X\overline{k} \triangleright_{\beta,w} \overline{p_k + 1}$$
porque os numerais de Church estão em forma normal tanto no $\lambda$ quanto na LC.

$\square$

# Representando as funções recursivas
## Prova do teorema (continuação)

### continuação.

Agora temos que mostrar que $\overline{\phi}\,\overline{m_1}\ldots\overline{m_n}$ não tem forma normal. Pelo Corolário 3.19.2, é suficiente dar uma redução infinita quasi-mais-à-esquerda desse termo. Considere a seguinte redução:

$$
\begin{array}{lll}
\overline{\phi}\,\overline{m_1}\ldots\overline{m_n} \triangleright_{\beta,w} & PX\overline{0}\mathrm{I}G & \text{pela def. de } \overline{\phi} \\
\triangleright_{\beta,w} & TX(X\overline{0})(TX)\overline{0}\mathrm{I}G & \text{pela def. de } P \\
\triangleright_{\beta,w} & TX(\overline{p_0+1})(TX)\overline{0}\mathrm{I}G & \text{reduzindo-se } X\overline{0} \\
\triangleright_{\beta,w} & (\lambda uv.u(X(\overline{\sigma}v))u(\overline{\sigma}v))(TX)\overline{0}\mathrm{I}G & \text{pela def. de } T, \mathrm{D} \\
\triangleright_{\beta,w} & TX(X(\overline{\sigma}\overline{0}))(TX)(\overline{\sigma}\overline{0})\mathrm{I}G & \\
\triangleright_{\beta,w} & TX(X\overline{1})(TX)\overline{1}\mathrm{I}G & \text{pela def. de } \overline{\sigma} \\
\triangleright_{\beta,w} & \ldots & \\
\triangleright_{\beta,w} & TX(X\overline{2})(TX)\overline{2}\mathrm{I}G & \text{igualmente} \\
\triangleright_{\beta,w} & \ldots &
\end{array}
$$

Essa redução é infinita, e cada parte

$$TX(X\overline{i})(TX)\overline{i}\mathrm{I}G \triangleright_{\beta,w} TX(X(\overline{i+1}))(TX)(\overline{i+1})\mathrm{I}G$$

contém pelo menos uma contração maximal mais-à-esquerda. □