Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Expansibilidade em Cálculos de Substituições Explícitas

Fábio Henrique Da Silva

Dissertação apresentada como requisito parcial
para conclusão do Mestrado em Informática

Orientador
Prof. Dr. Mauricio Ayala Rincón

Coorientador
Prof. Dr. Daniel Lima Ventura

Brasília
2012

Universidade de Brasília — UnB
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Mestrado em Informática

Coordenadora: Prof. Dr. Myléne Christine Queiroz de Farias

Banca examinadora composta por:

Prof. Dr. Mauricio Ayala Rincón (Orientador) — CIC/UnB
Prof. Dr. Flávio L. C. de Moura — CIC/UnB
Prof. Dr. Edward Hermann Haeusler — (Puc-Rio)

CIP — Catalogação Internacional na Publicação

Da Silva, Fábio Henrique.

Expansibilidade em Cálculos de Substituições Explícitas / Fábio Henrique Da Silva. Brasília : UnB, 2012.
54 p. : il. ; 29,5 cm.

Dissertação (Mestrado) — Universidade de Brasília, Brasília, 2012.

1. Expansibilidade, 2. Cálculos de Substituições Explícitas, 3. Teorema de Scott.

CDU 2011/0009999

Endereço: Universidade de Brasília
Campus Universitário Darcy Ribeiro — Asa Norte
CEP 70910-900
Brasília–DF — Brasil

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Expansibilidade em Cálculos de Substituições Explícitas

Fábio Henrique Da Silva

Dissertação apresentada como requisito parcial
para conclusão do Mestrado em Informática

Prof. Dr. Mauricio Ayala Rincón (Orientador)
CIC/UnB

Prof. Dr. Flávio L. C. de Moura
CIC/UnB

Prof. Dr. Edward Hermann Haeusler
(Puc-Rio)

Prof. Dr. Myléne Christine Queiroz de Farias
Coordenadora do Mestrado em Informática

Brasília, 29 de novembro de 2012

Agradecimentos

A Deus, pela proteção e força. Pois, sem Ele não chegaria aonde cheguei.

A minha família, pelo apoio, em especial aos meus pais Daniel e Antônia.

Aos meus familiares, que sempre acreditaram em mim, em especial ao meu primo Igor e minha tia Isabel.

Aos Professores Mauricio Ayala e Daniel Ventura, pela orientação.

Aos amigos, que foram companheiros de estudo como o Thiago, a Liliane, o Wosley, o Ulisses e o Fernando, dentre outros.

As pessoas do colégio CAIC/UNESCO pelo apoio, em especial as diretoras e ao pessoal com quem cheguei a trabalhar, como a Mazé, a Janaína, a Júlia, a Rosângela, a Cassilda, a Rose, a Mariana, o Sandro e a Claudielly, dentre outros.

A Comunidade Católica Palavra Viva, do qual participo.

Aos professores do Departamento da Matemática, Luciana, Helmar e Liliane.

Aos amigos do Departamento da Matemática e do Departamento de Ciência da Computação.

Ao CNPq e ao Programa Reuni, pelo suporte financeiro.

Resumo

Os cálculos de Substituições Explícitas (CSEs) são variações do cálculo λ que especificam de maneira concreta a operação de substituição, definida de maneira implícita no cálculo λ. Estes cálculos estendem a linguagem do cálculo λ de maneira a atomizar os passos envolvidos numa aplicação concreta da operação de substituição.

Este trabalho abordará a propriedade de expansibilidade em alguns CSEs que podemos dizer que seja uma investigação do passado de um termo. Esta propriedade é interessante quando este passado nos revela um termo *puro*, ou seja, um termo pertencente à linguagem do cálculo λ. Para isso fez-se necessário estudar várias outras propriedades, como a simulação da regra β do cálculo λ, a correção da regra (B) no $\lambda\upsilon$-cálculo e, a propriedade de Projeção do $\lambda\upsilon$-cálculo e do $\lambda\sigma$-cálculo.

O objetivo é estudar o problema de expansão no $\lambda\sigma$-cálculo, e para isso observamos os resultados de Ariel Arbiser no $\lambda\upsilon$-cálculo, em que o Teorema de Scott foi uma ferramenta crucial.

Palavras-chave: Expansibilidade, Cálculos de Substituições Explícitas, Teorema de Scott.

Abstract

Calculi of Explicit Substitutions (CSEs) are variants of the λ calculus which specify concretely the substitution operation, defined implicitly in the λ calculus. These calculi extend the language of λ calculus in order to atomize the steps involved in the practical application of replacement operation.

This work will discuss the expansion property in some CSEs, that it is an investigation of the past of a term. This property is interesting when it discloses a past term *pure*, i.e. a term belonging to the language of λ calculus. For this it was necessary to study several other properties, such as the simulation of rule β of λ calculus, the correction of the rule (B) in $\lambda\upsilon$-calculus and the property projection of $\lambda\upsilon$-calculus and the $\lambda\sigma$-calculus.

The goal is to study the expansion problem in the $\lambda\sigma$-calculus and verify the aplication of the results of Ariel Arbiser in the $\lambda\upsilon$-calculi, in which the Scott Theorem was a crucial tool.

Keywords: Expansion, Explicit Substitutions Calculi, Scott Theorem.

Sumário

1 Introdução **1**

2 Generalidades e notação **3**

2.1 Elementos Básicos . . . 3
2.2 O Cálculo λ . . . 5
2.3 A Notação *de Bruijn* . . . 6
2.4 Simulação, Correção e Projeção . . . 9
2.5 O $\lambda_\emptyset$-cálculo . . . 9
2.5.1 Simulação e correção . . . 10

3 Cálculos de Substituições Explícitas (CSEs) **12**

3.1 O λx-cálculo . . . 12
3.1.1 Simulação e Correção . . . 13
3.2 O $\lambda\upsilon$-cálculo . . . 15
3.2.1 Simulação e Correção de $\lambda\upsilon$-cálculo . . . 17
3.3 O $\lambda\sigma$-cálculo . . . 24
3.3.1 Sintaxe . . . 24
3.3.2 Simulação e Projeção . . . 25

4 Estudo da Expansibilidade nos CSEs **29**

4.1 Sistemas de numerais . . . 29
4.2 Decidibilidade . . . 30
4.3 Expansibilidade . . . 32
4.3.1 Expansibilidade em λx . . . 32
4.3.2 Indecibilidade em $\lambda\upsilon$ . . . 33
4.3.3 Resultados de indecibilidade em $\lambda\sigma$ . . . 43

5 Conclusão **46**

Referências **48**

Capítulo 1

Introdução

O cálculo λ, desenvolvido por Alonzo Church, surgiu na década de 30, definido como um mecanismo algorítmico para computar funções numéricas. Kleene e Rosser demonstraram que todas as funções recursivas podem ser representadas no cálculo λ [8]. E uma das principais operações deste cálculo é a operação de substituição chamada de β-conversão, que consiste em aplicar uma expressão (considerada como uma função) a outra expressão (considerada como um argumento).

Os cálculos de substituições explícitas (CSEs), extensões do cálculo λ, foram criados e estudados com a idéia de concretizar a operação de substituição, que é em geral considerada implícita no processo da β-conversão. Em substituições explícitas esta operação é promovida a uma linguagem de estudo, tornando-a assim um novo operador especificado concretamente sobre uma determinada linguagem, extensão do cálculo λ. Propriedades do cálculo λ devem ser preservadas e propriedades operacionais da substituição são então estudadas matematicamente.

Este trabalho abordará a propriedade de expansibilidade em alguns CSEs, que é dado N, sobre a linguagem estendida de um CSE, obtemos um termo *puro* M, tal que M é "simplificado" em N. Esta propriedade é interessante quando o termo M é *puro*, ou seja, o termo obtido pertence a linguagem do cálculo λ.

Nosso objetivo é estudar a propriedade de expansibilidade no $\lambda\sigma$-cálculo observando alguns resultados de Ariel Arbiser [3] para o $\lambda\upsilon$-cálculo, e tentar obter critérios para um subconjunto do $\lambda\sigma$-cálculo para o qual a propriedade possa ser válida. O Teorema de Scott será a principal ferramenta para atingir os nossos objetivos, assim como foi no caso de $\lambda\upsilon$ feito por Ariel Arbiser.

Este trabalho utiliza como principal fonte de referência o trabalho de Ariel Arbiser, principalmente na estratégia utilizada para provar os resultados de expansibilidade no $\lambda\upsilon$-cálculo. Para o $\lambda\sigma$-cálculo ainda não foi provada a decidibilidade em relação à expansibilidade, possibilidade que conjecturamos que seja indecidível.

Especificamente nossa contribuição foi:

a) provar a propriedade de simulação e correção do λx-cálculo;

b) aprofundar nas provas das propriedades de simulação e correção do $\lambda\upsilon$-cálculo, como também na de indecibilidade da expansão deste cálculo, feitas por Ariel Arbiser;

c) fazer, com um pouco mais de detalhes, a prova da propriedade de simulação da β-contração para o $\lambda\sigma$-cálculo;

d) demonstrar que as relações $\underset{\lambda\sigma}{\twoheadrightarrow}$ e $=_{\lambda\sigma}$ são indecidíveis;

e) provar os Corolários 12 e 13, e a Proposição 12, do $\lambda\sigma$-cálculo, que usam as relações $\underset{\lambda\sigma}{\twoheadrightarrow}$ e $=_{\lambda\sigma}$;

f) e por fim, esquematizar os passos de uma possível prova de indecibilidade da expansão do $\lambda\sigma$-cálculo.

O trabalho está organizado nos seguintes capítulos:

- **Capítulo 2:** As generalidades e as notações iniciais que usaremos durante o trabalho se encontram neste capítulo. O cálculo λ é apresentado no formato de nomes e no de numerais de *de Bruijn*. E o $\lambda_{\emptyset}$-cálculo, também é introduzido neste capítulo, para observamos como funcionam as propriedades de simulação e correção.

- **Capítulo 3:** Os CSEs são divididos em seções diferentes, onde cada seção principal apresenta um CSE. Colocamos a aplicação das propriedades de simulação, de correção e de projeção neste capítulo nos CSEs. Sendo a simulação válida para todos os cálculos apresentados, a correção válida somente para os cálculos λx e $\lambda\upsilon$; e a projeção válida para os cálculos $\lambda\upsilon$ e $\lambda\sigma$.

- **Capitulo 4:** Encontramos aqui o estudo da expansibilidade nos CSEs, como os resultados obtidos por Ariel Arbiser com relação a indecibilidade de $\lambda\upsilon$. Temos aqui também o Teorema de Scott. Dividimos este capítulo nas seções:

    1. **Sistemas de numerais**, onde colocamos conceitos de valores booleanos, de pares e de numerais padrão.
    2. **Decidibilidade**, onde se encontram o Teorema do Ponto Fixo, o Teorema de Scott e resultados de indecibilidade no cálculo λ.
    3. **Expansibilidade**, aqui se encontra o estudo da expansão do λx, a indecibilidade do $\lambda\upsilon$-cálculo e a nossa maior contribuição que são os resultados de indecidibilidade de algumas relações do $\lambda\sigma$-cálculo.

- **Capítulo 5:** A conclusão com respeito ao trabalho é feita falando das dificuldades encontradas para provar a indecibilidade do $\lambda\sigma$-cálculo, que ainda está em aberto. E de possíveis passos para alcançar tal objetivo. Além disso, apresentamos também os resultados obtidos pelos autores das referência estudadas.

Capítulo 2

Generalidades e notação

Introduziremos neste capítulo as notações que iremos usar durante toda a dissertação, sendo importante para consulta em capítulos posteriores.

Veremos, neste capítulo, elementos básicos de sistema abstrato de reescrita, o cálculo λ e o $\lambda_\emptyset$-cálculo. Sendo o $\lambda_\emptyset$-cálculo não tão importante quanto os demais assuntos abordados neste capítulo.

2.1 Elementos Básicos

Introduziremos nesta seção uma noção básica de sistema abstrato de reescrita como também de redução, de confluência e de normalização. Para maiores detalhes veja [7, 10] e [4].

Definição 1 (ARS) *Um "Sistema Abstrato de Reescrita" (ARS) é um par $(R, \rightarrow_R)$ onde R é um conjunto cujos componentes são chamados de objetos ou elementos e $\rightarrow_R$ uma relação binária sobre R, onde $(u, v) \in \rightarrow_R$ é denotado por $u \rightarrow_R v$ e diz-se que u reduz para v. R pode ser eliminado de $\rightarrow_R$ se não houver dificuldade de entendimento.*

Uma sequência de reduções $u_1 \rightarrow u_2 \rightarrow u_3 \rightarrow \ldots$ é chamada de uma ***derivação*** começado com u_1, não sendo necessariamente uma sequência de reduções infinita. Nós chamamos de *grau* da derivação o número de reduções, 0 ou mais, finito ou infinito.

Para qualquer redução $\rightarrow_R$ nós usamos:

1. $\rightarrow_R$ para denotar uma etapa de R-redução, $\stackrel{+}{\rightarrow}_R$ para seu fecho transitivo, $\underset{R}{\twoheadrightarrow}$ para seu fecho reflexivo transitivo, $=_R$ denota o fecho de equivalência de $\rightarrow_R$ e ${}_R\!\leftarrow$ denota a relação inversa de $\rightarrow_R$.

2. Se $M \underset{R}{\twoheadrightarrow} N$ diremos que N R-expande para M, e que M é uma expansão de N.

3. Se $M =_R N$, diz-se que M R-converte ou é R-conversível em N.

Definição 2 *Dado $(R, \rightarrow)$ um sistema de reescrita, um elemento $u \in R$ é denominado **irredutível** se não existe nenhum $v \in R$ tal que $u \rightarrow v$; no caso contrário, é denominado **redutível**.*

Definição 3 *Uma redução $\rightarrow$ é chamada **terminante** se somente se não existe sequência de reduções infinita $u_1 \rightarrow u_2 \rightarrow u_3 \rightarrow \ldots$.*

Exemplo 1 *Considere a relação de reescrita $\rightarrow$ sobre $\mathbb{N}$ definida por $n \rightarrow m$ sse $n = m + 7$. Os inteiros $0 \leq n \leq 6$ são irredutíveis e todo $n \geq 7$ é redutível. $\rightarrow$ é terminante.*

Definição 4 (Normalização) *Dado $(R, \rightarrow)$ temos:*

1. *uma **forma normal** é um elemento de R que é irredutível, i.e. um elemento $u \in R$ tal que não existe $v \in R$ com $u \rightarrow v$;*

2. *a relação de redução $\rightarrow$ é **fortemente normalizante (SN)**, **terminante**, ou **noetheriana**, se para toda sequência de redução começando de $u \in R$ é finita;*

3. *para $u, w, v \in R$ usaremos $(\twoheadrightarrow \circ \twoheadleftarrow)$ para denotar $u \twoheadrightarrow w \twoheadleftarrow v$ e usaremos também $(\twoheadleftarrow \circ \twoheadrightarrow)$ para denotar $u \twoheadleftarrow w \twoheadrightarrow v$;*

4. *dados $u, v \in R$ são denominados **juntáveis** se existe um $w \in R$ tal que $u \twoheadrightarrow w \twoheadleftarrow v$. Para denotar que dois termos u, v são juntáveis usar-se-á a seguinte notação: $u \downarrow v$.*

Como já mencionado na introdução, os CSEs são uma extensão do cálculo λ. E vamos definir a forma normal de um termo que pertence a um CSE.

Definição 5 *Dado λ_ξ-cálculo, um CSE, diremos que a **forma ξ-normal de um termo** $M \in \Lambda_\xi$, representado por $\xi(M)$, significará a forma normal do termo M usando as regras do λ_ξ-cálculo.*

Definição 6 (Confluência) *Um sistema de reescrita $(R, \rightarrow)$ é **confluente**, ou **Church-Rosser (CR)** sse*

$$(\twoheadleftarrow \circ \twoheadrightarrow) \subseteq (\twoheadrightarrow \circ \twoheadleftarrow).$$

Observação 1 *A definição de confluência significa que para todo $u, v, w \in R$ com $v \twoheadleftarrow u \twoheadrightarrow w$, existe algum $r \in R$ com $v \twoheadrightarrow r \twoheadleftarrow w$ (ou simplesmente $v \downarrow w$). Note que $\downarrow = (\twoheadrightarrow \circ \twoheadleftarrow)$; consequentemente, a confluência pode ser expressa da seguinte maneira [4]:*

$$(\twoheadleftarrow \circ \twoheadrightarrow) \subseteq \downarrow$$

.

Definição 7 (Confluência Local) *Um sistema de reescrita $(R, \rightarrow)$ é **localmente confluente** (ou local confluente), ou **fracamente Church-Rosser (WCR)** sse*

$$(\leftarrow \circ \rightarrow) \subseteq \downarrow .$$

*Diz-se que $(R, \rightarrow)$ tem a **propriedade de diamante** se*

$$(\leftarrow \circ \rightarrow) \subseteq (\rightarrow \circ \leftarrow)$$

2.2 O Cálculo λ

Estudaremos nesta seção o cálculo λ, que é importante para os capítulos posteriores, pois este cálculo é a base dos CSEs e pela aplicação do Teorema de Scott, assuntos que veremos em capítulos posteriores. Veremos a sintaxe do cálculo λ na notação de nomes, a regra β, a sintaxe na notação de *de Bruijn* e noção de simulação, correção e o projeção. Nas referências [4, 6, 8] e [11] podem ser encontrados mais detalhes sobre o cálculo λ.

Definição 8 (λ-termos) *O conjunto de termos do cálculo λ clássico, denotado como Λ, é descrito pela sintaxe:*

$$M ::= x \mid (M\ M) \mid (\lambda x.M)$$

onde x varia ao longo de um conjunto enumerável de variáveis $\mathcal{X}$ As letras $M, N, O, P, \ldots$ serão usadas para representar λ-termos.

O termo da forma $(M\ N)$ é chamado de uma *aplicação*, com a interpretação pretendida de "uma função M aplicada ao argumento N"; um termo da forma $(\lambda x.M)$ é chamado de uma *abstração*, com a interpretação pretendida de que "a função susbtitui argumentos para o parâmetro x no objeto M".

Nesta estrutura faz-se necessária a noção de *variáveis livres* (FV) e *variáveis ligadas* (BV).

Definição 9 (Variáveis livres e ligadas) *O conjunto de variáveis livres de M, denotado por $FV(M)$, é definido por:*

$$\begin{array}{lcl} FV(x) & = & \{x\}, \\ FV(MN) & = & FV(M) \cup FV(N), \\ FV(\lambda x.M) & = & FV(M) \setminus \{x\}\,. \end{array}$$

A variável x no termo $\lambda x.M$ é dita ligada. O conjunto de variáveis ligadas de M, denotado por $BV(M)$, é definido por:

$$\begin{array}{lcl} BV(x) & = & \emptyset, \\ BV(MN) & = & BV(M) \cup BV(N), \\ BV(\lambda x.M) & = & BV(M) \cup \{x\}. \end{array}$$

Definição 10 (Sub-termo) *Para todo par de termos M e N, M é um sub-termo de N, denotado por $M \subseteq N$, se somente se $M \in Sub(M)$ onde o conjunto de termos de $Sub(M)$ é definido como abaixo:*

$$\begin{array}{lcl} Sub(x) & = & \{x\} \\ Sub(M\ N) & = & Sub(M) \cup Sub(N) \cup \{(M\ N)\} \\ Sub(\lambda x.M) & = & Sub(M) \cup \{\lambda x.M\} \end{array}$$

A seguinte convenção de variável é adotada: os nomes de variáveis livres e ligadas não coincidem, além do mais, variáveis ligadas distintas têm nomes distintos. Igualdade sintática módulo renomeamento de variáveis ligadas é expressa usando o símbolo "=".

Nomes de variáveis ligadas são irrelevantes. Por exemplo, $\lambda x.x$ e $\lambda y.y$ representam o mesmo λ-termo. Esta correspondência é denominada α-conversão.

Um termo $M\{x \leftarrow N\}$ denota a *substituição atômica* das ocorrências livres da variável x em M por N.

A regra β do cálculo λ está definida como:

$$(\lambda x.M)N \rightarrow_{\beta} M\{x \leftarrow N\}$$

onde $\lambda x.M$ (um λ-termo, formado pelo símbolo λ e pela variável de abstração x) denota uma abstração com corpo M. Essencialmente, a idéia é aplicar um objeto funcional $(\lambda x.M)$, cujo parâmetro formal é x, ao argumento N. A notação $M\{x \leftarrow N\}$ denota o termo resultante da "substituição simultânea de todas as ocorrências livres de x em M por N".

Exemplo 2

$$(\lambda xy.x)N \rightarrow_{\beta} (\lambda y.x)\{x \leftarrow N\} = \lambda y.(x\{x \leftarrow N\}) = \lambda y.N.$$

Se $y \in FV(N)$ então fazemos um renomeamento de variáveis.

O termo $(\lambda x.M)N$ é chamado de β-**redex** e o seu **contractum** é $M\{x \leftarrow N\}$. A aplicação da regra β em um passo é chamado de β-**contração**.

Definição 11 *Sejam M, N λ-termos, e x uma variável. A substituição $M\{x \leftarrow N\}$ é definida indutivamente sobre M da seguinte forma:*

1. $x\{x \leftarrow N\} = N,$

2. $y\{x \leftarrow N\} = y,$ *se* $x \neq y$,

3. $(M_1\ M_2)\{x \leftarrow N\} = M_1\{x \leftarrow N\}\ M_2\{x \leftarrow N\}$ *e*

4. $(\lambda y.M_1)\{x \leftarrow N\} = \lambda y.M_1\{x \leftarrow N\}.$

Quando houver a possibilidade de captura de variável livre pelo λ, sempre podemos fazer o renomeamento de variável por meio da α-conversão.

2.3 A Notação *de Bruijn*

A notação inventada pelo matemático holandês N.G. de Bruijn usa números naturais no lugar de letras. Assim, na definição de λ-termos os parâmetros que ocorrem no corpo de um termo são reduzidos para números naturais que identificam o símbolo λ [11]. Por exemplo:

$$(\lambda(\lambda 2)) \text{ é equivalente a } \lambda xy.x$$

Definição 12 *O conjunto Λ_{dB} de λ-termos na notação de de Bruijn é definido indutivamente como:*

$$M, N ::= \underline{\mathbf{n}} \mid (M\ N) \mid (\lambda M)$$

tal que $\underline{\mathbf{n}} \in \mathbb{N}^*$.

Formalmente, os termos na notação de Bruijn são definidos indutivamente como o conjunto mínimo tal que:

1. *todo número natural (menos o zero) é um termo (variável)*
2. *Se M e N são termos, então $(M\ N)$ é um termo*
3. *Se M é um termo, (λM) é um termo e (β) é definida por:*

$$((\lambda P)\ Q) = P\{\underline{\mathbf{1}} \leftarrow Q\}$$

onde:

$$\underline{\mathbf{n}}\{\underline{\mathbf{m}} \leftarrow N\} = \begin{cases} \underline{\mathbf{n}} & se\ \ n < m \\ \underline{\mathbf{n-1}} & se\ \ n > m \\ renomear_{n,1}(N) & se\ \ n = m \end{cases}$$

$$(M_1\ M_2)\{\underline{\mathbf{m}} \leftarrow N\} = (M_1\{\underline{\mathbf{m}} \leftarrow N\}\ M_2\{\underline{\mathbf{m}} \leftarrow N\})$$

$$(\lambda M)\{\underline{\mathbf{m}} \leftarrow N\} = (\lambda M\{\underline{\mathbf{m+1}} \leftarrow N\})$$

e

$$renomear_{m,i}(\underline{\mathbf{j}}) = \begin{cases} \underline{\mathbf{j}} & se\ \ j < i \\ \underline{\mathbf{j+m-1}} & se\ \ j \geq i \end{cases}$$

$$renomear_{m,i}(N_1\ N_2) = (renomear_{m,i}(N_1)\ renomear_{m,i}(N_2))$$

$$renomear_{m,i}(\lambda N) = (\lambda renomear_{m,i+1}(N))$$

O exemplo a seguir mostra como funciona a notação de Bruijn. E abreviaremos nele a aplicação *renomear* para *ren*.

Exemplo 3

$$
\begin{array}{ccc}
(\lambda r((\lambda x s(u\,x))\,((\lambda v(w\,v))\,r)))\,y & = & ((\lambda((\lambda(\lambda(\underline{\mathbf{4}}\,\underline{\mathbf{2}})))\,((\lambda(\underline{\mathbf{3}}\,\underline{\mathbf{1}}))\,\underline{\mathbf{1}})))\,\underline{\mathbf{1}}) \\
\downarrow\beta & & \downarrow\beta \\
(\lambda r((\lambda s(u\,x))\,\{x \leftarrow ((\lambda v(w\,v))\,r)\}))\,y & = & ((\lambda(\lambda(\underline{\mathbf{4}}\,\underline{\mathbf{2}}))\{\underline{\mathbf{1}} \leftarrow ((\lambda(\underline{\mathbf{3}}\,\underline{\mathbf{1}}))\,\underline{\mathbf{1}})\})\,\underline{\mathbf{1}}) \\
= & & = \\
(\lambda r(\lambda s(u\,x)\,\{x \leftarrow ((\lambda v(w\,v))\,r)\}))\,y & & ((\lambda(\lambda(\underline{\mathbf{4}}\,\underline{\mathbf{2}})\{\underline{\mathbf{2}} \leftarrow ((\lambda(\underline{\mathbf{3}}\,\underline{\mathbf{1}}))\,\underline{\mathbf{1}})\}))\,\underline{\mathbf{1}}) \\
= & & = \\
(\lambda r(\lambda s(u\,\{x \leftarrow ((\lambda v(w\,v))\,r)\} & & ((\lambda(\lambda(\underline{\mathbf{4}}\{\underline{\mathbf{2}} \leftarrow ((\lambda(\underline{\mathbf{3}}\,\underline{\mathbf{1}}))\,\underline{\mathbf{1}})\}\;\underline{\mathbf{2}}\{\underline{\mathbf{2}} \leftarrow ((\lambda(\underline{\mathbf{3}}\,\underline{\mathbf{1}}))\,\underline{\mathbf{1}})\})))\,\underline{\mathbf{1}}) \\
x\,\{x \leftarrow ((\lambda v(w\,v))\,r)\})))\,y & & = \\
= & & \\
 & & ((\lambda(\lambda(\underline{\mathbf{3}}\;ren_{2,1}((\lambda(\underline{\mathbf{3}}\,\underline{\mathbf{1}}))\,\underline{\mathbf{1}}))))\,\underline{\mathbf{1}}) \\
 & & = \\
 & & ((\lambda(\lambda(\underline{\mathbf{3}}\;(ren_{2,1}(\lambda(\underline{\mathbf{3}}\,\underline{\mathbf{1}}))\;ren_{2,1}(\underline{\mathbf{1}})))))\,\underline{\mathbf{1}}) \\
 & & = \\
 & & ((\lambda(\lambda(\underline{\mathbf{3}}\;((\lambda ren_{2,2}(\underline{\mathbf{3}}\,\underline{\mathbf{1}}))\,\underline{\mathbf{2}}))))\,\underline{\mathbf{1}}) \\
 & & = \\
 & & ((\lambda(\lambda(\underline{\mathbf{3}}\;((\lambda(ren_{2,2}(\underline{\mathbf{3}})\;ren_{2,2}(\underline{\mathbf{1}})))\,\underline{\mathbf{2}}))))\,\underline{\mathbf{1}}) \\
 & & = \\
(\lambda r(\lambda s(u\,((\lambda v(w\,v))\,r))))\,y & = & ((\lambda(\lambda(\underline{\mathbf{3}}\;((\lambda(\underline{\mathbf{4}}\,\underline{\mathbf{1}}))\,\underline{\mathbf{2}}))))\,\underline{\mathbf{1}}) \\
\downarrow\beta & & \downarrow\beta \\
(\lambda r(\lambda s(u\,((w\,v)\,\{v \leftarrow r\}))))\,y & = & ((\lambda(\lambda(\underline{\mathbf{3}}\;(\underline{\mathbf{4}}\,\underline{\mathbf{1}})\{\underline{\mathbf{1}} \leftarrow \underline{\mathbf{2}}\})))\,\underline{\mathbf{1}}) \\
= & & = \\
 & & ((\lambda(\lambda(\underline{\mathbf{3}}\;(\underline{\mathbf{4}}\{\underline{\mathbf{1}} \leftarrow \underline{\mathbf{2}}\}\;\underline{\mathbf{1}}\{\underline{\mathbf{1}} \leftarrow \underline{\mathbf{2}}\}))))\,\underline{\mathbf{1}}) \\
 & & = \\
 & & ((\lambda(\lambda(\underline{\mathbf{3}}\;(\underline{\mathbf{3}}\;ren_{1,1}(\underline{\mathbf{2}})))))\,\underline{\mathbf{1}}) \\
 & & = \\
(\lambda r(\lambda s(u\,(w\,\{v \leftarrow r\}\;\;v\,\{v \leftarrow r\}))))\,y & & \\
= & & \\
(\lambda r(\lambda s(u\,(w\,r))))\,y & = & ((\lambda(\lambda(\underline{\mathbf{3}}\;(\underline{\mathbf{3}}\,\underline{\mathbf{2}}))))\,\underline{\mathbf{1}}) \\
\downarrow\beta & & \downarrow\beta \\
(\lambda s(u\,(w\,r)))\,\{r \leftarrow y\} & = & (\lambda(\underline{\mathbf{3}}\;(\underline{\mathbf{3}}\,\underline{\mathbf{2}})))\{\underline{\mathbf{1}} \leftarrow \underline{\mathbf{1}}\} \\
= & & = \\
(\lambda s(u\,(w\,r))\,\{r \leftarrow y\}) & & (\lambda(\underline{\mathbf{3}}\;(\underline{\mathbf{3}}\,\underline{\mathbf{2}}))\{\underline{\mathbf{2}} \leftarrow \underline{\mathbf{1}}\}) \\
= & & = \\
(\lambda s(u\,\{r \leftarrow y\}\;\;(w\,r)\,\{r \leftarrow y\})) & & (\lambda(\underline{\mathbf{3}}\{\underline{\mathbf{2}} \leftarrow \underline{\mathbf{1}}\}\;(\underline{\mathbf{3}}\,\underline{\mathbf{2}})\{\underline{\mathbf{2}} \leftarrow \underline{\mathbf{1}}\})) \\
= & & = \\
(\lambda s(u\,\{r \leftarrow y\}\;\;(w\,\{r \leftarrow y\}\;\;r\,\{r \leftarrow y\}))) & & (\lambda(\underline{\mathbf{2}}\;(\underline{\mathbf{3}}\{\underline{\mathbf{2}} \leftarrow \underline{\mathbf{1}}\}))\underline{\mathbf{2}}\{\underline{\mathbf{2}} \leftarrow \underline{\mathbf{1}}\}) \\
 & & = \\
 & & (\lambda(\underline{\mathbf{2}}\;(\underline{\mathbf{2}}\;ren_{2,1}(\underline{\mathbf{1}})))) \\
= & & = \\
(\lambda s(u\,(w\,y))) & = & (\lambda(\underline{\mathbf{2}}\;(\underline{\mathbf{2}}\,\underline{\mathbf{2}})))
\end{array}
$$

Existe uma simples tradução entre λ-termos e termos na notação de de Bruijn (observe que termos α-congruentes resultam no mesmo termo na notação de Bruijn):

$$
\begin{array}{ll}
DB\;x\;\;(x_1,\ldots,x_n) & = \underline{\mathbf{i}}, \text{ se } i \text{ é o mínimo tal que } x = x_i \\
DB(\lambda x M)\;\;(x_1,\ldots,x_n) & = (\lambda DB\;M\;\;(x, x_1,\ldots,x_n)) \\
DB(M\;N)\;\;(x_1,\ldots,x_n) & = (DB\;M\;\;(x_1,\ldots,x_n)\;DB\;N\;\;(x_1,\ldots,x_n))
\end{array}
$$

Definição 13 (Contexto) *Um contexto C é termo contendo um ou mais buracos em determinadas posições, isto é, gerado pela sintaxe:*

$$C ::= \Box \mid (C\;N) \mid (N\;C) \mid (C\;C) \mid (\lambda C)$$

tal que N é um termo pertencente ao cálculo λ ou a um CSE.

2.4 Simulação, Correção e Projeção

Nesta seção introduziremos as propriedades de simulação, correção e projeção. Estas propriedades serão o ponto chave para provas de lemas e corolários mais a frente.

Para as definições abaixo assumimos que λ_ξ-cálculo seja um CSE.

A propriedade de simulação é a aplicação das regras de um λ_ξ-cálculo a um termo $M \in \Lambda$, reduzindo a um termo $N \in \Lambda$, tal que a regra β é simulada neste cálculo.

Definição 14 (Simulação) *Dados $M, N \in \Lambda$ tal que $M \to_\beta N$ então $M \xrightarrow{+}_{\lambda_\xi} N$.*

A propriedade de correção é a aplicação da regra β, em um ou mais passos, a um termo $M \in \Lambda$, reduzindo a um termo na forma normal $\xi(N')$.

Definição 15 (Correção) *Dado $M \in \Lambda$ e $N \in \Lambda_\xi$ tal que $M \to_{\lambda_\xi} N$ então $M \underset{\beta}{\twoheadrightarrow} \xi(N)$.*

Observação 2 *A propriedade de correção também pode ser expressar como:*

Dado $M \in \Lambda$ e $N \in \Lambda_\xi$ tal que $M \to_{\lambda_\xi} N$ então $M =_\beta \xi(N)$. Pois se vale $M \underset{\beta}{\twoheadrightarrow} \xi(N) \Longrightarrow M =_\beta \xi(N)$[8].

A propriedade de projeção é a aplicação da regra β, em um ou mais passos, de uma forma normal $\xi(M)$, com $M \in \Lambda_\xi$, reduzindo a um termo $N' = \xi(N)$.

Definição 16 (Projeção) *Dados $M, N \in \Lambda_\xi$ tal que $M \to_{\lambda_\xi} N$ então $\xi(M) \underset{\beta}{\twoheadrightarrow} \xi(N)$.*

2.5 O $\lambda_\emptyset$-cálculo

Introduziremos o $\lambda_\emptyset$-cálculo [2], para estudarmos o estilo de prova da propriedade de simulação e correção, e usarmos em cálculos subsequentes. As Proposições 1 e 2 e o Corolário 1 são da referência [2], os quais provamos com mais detalhes nesta seção.

Definição 17 *O $\lambda_\emptyset$-cálculo é definido com a mesma sintaxe do cálculo λ, onde a regra β é substituída pelas quatro regras abaixo:*

$$
\begin{array}{lllll}
(\lambda var1) & (\lambda x.x)M & \longrightarrow & M \\
(\lambda var2) & (\lambda x.y)M & \longrightarrow & y, se\ x \neq y \\
(\lambda app) & (\lambda x.PQ)M & \longrightarrow & (\lambda x.P)M((\lambda x.Q)M) \\
(\lambda\lambda) & (\lambda x.(\lambda y.P))M & \longrightarrow & \lambda y.(\lambda x.P)M se\ x \neq y\ e\ y \notin FV(M)
\end{array}
$$

A convenção usual sobre variáveis aplica; p.ex., na regra $(\lambda\lambda)$, se $y \in FV(M)$, então um renomeamento da variável y em P é aplicado. A compatibilidade sobre $\lambda_\emptyset$ é definida da seguinte maneira: se $M \to_{\lambda_\emptyset} M'$ então $MN \to_{\lambda_\emptyset} M'N, NM \to_{\lambda_\emptyset} NM'$ e $\lambda x.M \to_{\lambda_\emptyset} \lambda x.M'$.

2.5.1 Simulação e correção

Veremos que o $\lambda_\emptyset$-cálculo possui a propriedade da simulação da β-redução, e a propriedade de correção com respeito a $=_\beta$[2]. Apesar do $\lambda_\emptyset$-cálculo não ser um CSE vale a propriedade de simulação e correção.

Proposição 1 (Simulação) *Se $M \to_\beta N$ então $M \xrightarrow{+}_{\lambda_\emptyset} N$.*

Prova. Faremos a prova por indução sobre a estrutura de M. Usaremos as siglas BI, para base de indução, e HI, para hipótese de indução.

Então segue:

- se $M = (\lambda x.U)V \to_\beta U\{x \leftarrow V\}$, então os seguintes casos podem ocorrer:
  - se U é uma variável existem dois casos a considerar (BI):
    - $U = x$, $M \to_\beta x\{x \leftarrow V\} = V$ pela Definição 11, e $M = (\lambda x.x)V \to_{\lambda var1} V$,
    - $U = y$, $M \to_\beta y\{x \leftarrow V\} = y$ pela Definição 11, e $M = (\lambda x.y)V \to_{\lambda var2} y$,
  - $U = PQ$, $M \to_\beta PQ\{x \leftarrow V\} = P\{x \leftarrow V\}\ Q\{x \leftarrow V\}$ pela Definição 11, e $M = (\lambda x.PQ)V \to_{\lambda app} (\lambda x.P)V((\lambda x.Q)V)$

    $$\overset{HI}{\xrightarrow{+}_{\lambda_\emptyset}} P\{x \leftarrow V\}Q\{x \leftarrow V\},$$

  - $U = \lambda y.P$, $M \to_\beta \lambda y.P\{x \leftarrow V\}$ e $M \to_{\lambda\lambda} \lambda y.(\lambda x.P)V \overset{HI}{\xrightarrow{+}_{\lambda_\emptyset}} \lambda y.P\{x \leftarrow V\}$,

- se a redução é interna, então os seguintes casos podem ocorrer:
  - $PQ \to_\beta P'Q$ com $P \to_\beta P'$, então pela HI $P \xrightarrow{+}_{\lambda_\emptyset} P'$ assim $PQ \xrightarrow{+}_{\lambda_\emptyset} P'Q$,
  - $PQ \to_\beta PQ'$ com $Q \to_\beta Q'$, então pela HI $Q \xrightarrow{+}_{\lambda_\emptyset} Q'$ assim $PQ \xrightarrow{+}_{\lambda_\emptyset} PQ'$,
  - $\lambda x.P \to_\beta \lambda x.P'$ com $P \to_\beta P'$, então pela HI $P \xrightarrow{+}_{\lambda_\emptyset} P'$ assim $\lambda x.P \xrightarrow{+}_{\lambda_\emptyset} \lambda x.P'$.

No primeiro caso analisamos U como sendo uma variável, no caso de $U = c$, sendo c uma constante, o caso se reduz ao de tomar $y = c \neq x$. □

Proposição 2 (Correção com respeito a $=_\beta$) *Se $M \underset{\lambda_\emptyset}{\twoheadrightarrow} N$ então $M =_\beta N$.*

Prova. A prova será feita nas regras de $\lambda_\emptyset$. Então o resultado seguirá por indução no tamanho da derivação de $\underset{\lambda_\emptyset}{\twoheadrightarrow}$.

- se a redução ocorre na raiz, então os seguintes casos podem ocorrer:

– se a redução é $M = (\lambda x.x)V \to_{\lambda var1} V$, então pela definição de β temos que $M = (\lambda x.x)V \to_\beta x\{x \leftarrow V\} = V$

– se a redução é $M = (\lambda x.y)V \to_{\lambda var2} y$ com $x \neq y$, então pela definição de β temos que $M = (\lambda x.y)V \to_\beta y\{x \leftarrow V\} = y$ com $x \neq y$

– se a redução é $M = (\lambda x.PQ)V \to_{\lambda app} (\lambda x.P)V((\lambda x.Q)V)$, então pela definição de β o termo $(\lambda x.PQ)V \to_\beta (PQ)\{x \leftarrow V\}$ e pela definição de substituição temos $(PQ)\{x \leftarrow V\} = P\{x \leftarrow V\}Q\{x \leftarrow V\}$. Por fim, pela definição de β o termo $(\lambda x.P)V((\lambda x.Q)V) \underset{\beta}{\twoheadrightarrow} P\{x \leftarrow V\}(Q\{x \leftarrow V\}) = P\{x \leftarrow V\}Q\{x \leftarrow V\}$.

– se a redução é $M = (\lambda x.(\lambda y.P))V \to_{\lambda\lambda} \lambda y.(\lambda x.P)V$, então pela definição de β o termo $M' = (\lambda x.(\lambda y.P))V \to_\beta \lambda y.P\{x \leftarrow V\}$ e temos que $\lambda y.(\lambda x.P)V \to_\beta \lambda y.P\{x \leftarrow V\}$.

- se a redução é interna, então os seguintes casos podem ocorrer:
  – $PQ \to_{\lambda_\emptyset} P'Q$ com $P \to_{\lambda_\emptyset} P'$, então pela HI $P =_\beta P'$ assim $PQ =_\beta P'Q$

  – $PQ \to_{\lambda_\emptyset} PQ'$ com $Q \to_{\lambda_\emptyset} Q'$, então pela HI $Q =_\beta Q'$ assim $PQ =_\beta PQ'$

  – $\lambda x.P \to_{\lambda_\emptyset} \lambda x.P'$ com $P \to_{\lambda_\emptyset} P'$, então pela HI $P =_\beta P'$ assim $\lambda x.P =_\beta x.P'$

- se a relação $=_\beta$ vale para n passos então para $n+1$ passos temos que $M \to_{\lambda\emptyset}^{n+1} N$. E assim existe N' tal que $M \to_{\lambda\emptyset}^{n} N'$ e por HI $M =_\beta N'$ e pelos casos anteriores temos que $N' \to_\beta N$ ou $N' \underset{\beta}{\twoheadrightarrow} N$, logo $N' =_\beta N$ e portanto $M =_\beta N$.

□

Corolário 1 *As relações $=_\beta$ e $=_{\lambda_\emptyset}$ coincidem.*

Prova. Se $M =_\beta N$ então temos que $\exists L \in \Lambda$ tal que $M \underset{\beta}{\twoheadrightarrow} L$ e $N \underset{\beta}{\twoheadrightarrow} L$ [8], e pela proposição 1 obtemos $M \underset{\lambda_\emptyset}{\twoheadrightarrow} L$ e $N \underset{\lambda_\emptyset}{\twoheadrightarrow} L$, que implica que $M =_{\lambda_\emptyset} N$. Por outro lado, $M =_{\lambda_\emptyset} N$ vale porque $M \underset{\lambda_\emptyset}{\twoheadrightarrow} N$, e pela proposição 2 temos $M =_\beta N$. □

Capítulo 3

Cálculos de Substituições Explícitas (CSEs)

Introduziremos três cálculos de substituições explícitas, analisando a formação dos seus termos e algumas das suas características.

Os CSEs são extensões do cálculo λ e foram desenvolvidos com o intuito de expressar mais claramente a operação de substituição. Pelo fato de que os CSEs possuem novas regras e novos elementos em relação ao cálculo λ o seu estudo permiti verificar propriedades que ainda não foram provadas para estes novos cálculos. Assim, como o objetivo do trabalho é estudar a expansibilidade no $\lambda\sigma$-cálculo, é necessário verificar a validade de algumas propriedades, como a simulação, a correção e a projeção nos CSEs e assim aplicá-las mais a frente no estudo da expansibilidade. Para familiarizar com as regras dos CSEs foi utilizado o SUBSEXPL (SUBStituições EXPLícitas) que é uma implementação OCaml de redução através dos CSEs [12].

3.1 O λx-cálculo

O λx-cálculo é uma extensão do cálculo λ. E veremos suas características segundo Ariel Arbiser [2]. A operação de substituição é incorporada no nível-objeto (i.e linguagem de estudo), assim temos uma regra *Beta* que inicia o processo de simulação da β-redução:

$$(\lambda x.M)N \to_{Beta} M \langle x := N \rangle$$

onde $\bullet \langle \bullet := \bullet \rangle$ é um novo operador no cálculo, para executar a substituição de maneira explícita.

Definição 18 (λx-termos) *Dado um conjunto enumerável de variáveis $\mathcal{X}$ o conjunto de termo denotado por* Λx*, são dados pela seguinte sintaxe:*

$$M ::= x \mid (M\ M) \mid (\lambda x.M) \mid M \langle x := M \rangle$$

onde $x \in \mathcal{X}$

O operador $\bullet \langle \bullet := \bullet \rangle$ é chamado de *operador de substituição*. Os termos sem ocorrência de operadores de substituição são chamados *termos puros*. A convenção usual de variáveis é mantida com a observação adicional de que a variável x no termo $M\langle x := N\rangle$ liga as ocorrências livres de x em M; M é chamado o alvo e N o corpo da substituição. Fazemos aqui uma extensão da definição de FV a partir do cálculo λ, tal que $FV(M)$ denota o conjunto de variáveis livres no termo M, definido por $FV(M\langle x := N\rangle) =_{def} (FV(M) \setminus \{x\}) \cup FV(N)$.

As regras do λx-cálculo são:

$$
\begin{array}{lllll}
(Beta) & (\lambda x.M)N & \longrightarrow & M\langle x := N\rangle & \\
(App) & (MN)\langle x := P\rangle & \longrightarrow & M\langle x := P\rangle N\langle x := P\rangle & \\
(Lam) & (\lambda y.M)\langle x := P\rangle & \longrightarrow & \lambda y.M\langle x := P\rangle & x \neq y \\
(Var) & x\langle x := P\rangle & \longrightarrow & P & \\
(Gc) & M\langle x := P\rangle & \longrightarrow & M & \text{se } x \notin FV(M)
\end{array}
$$

A regra Var pode ser usada da forma $Varx$ que significa aplicar a regra Var na variável x. O λx-cálculo é confluente [2]. O λx-cálculo sem a regra $Beta$ é chamado de sub-cálculo de substituição de λx e é denominado x. Este sub-cálculo é SN e confluente, e os termos normais são termos puros [2]. Assim se $M \in \Lambda$x então usamos x(M) para denominar a única forma x-normal de M.

Exemplo 4 *Apresentamos dois exemplos, no primeiro uma redução usando a regra β em um passo, e no segundo usando em vários passos as regras do λx-cálculo, obtendo o mesmo termo.*

$$(\lambda x.x(y\ x))N \quad \rightarrow_\beta N(y\ N)$$

$$
\begin{array}{ll}
(\lambda x.x(y\ x))N & \rightarrow_{Beta} (x(y\ x))\langle x := N\rangle \\
& \rightarrow_{App} x\langle x := N\rangle (y\ x)\langle x := N\rangle \\
& \rightarrow_{App} x\langle x := N\rangle (y\langle x := N\rangle\ x\langle x := N\rangle) \\
& \rightarrow_{Gc} x\langle x := N\rangle (y\ x\langle x := N\rangle) \\
& \rightarrow_{Varx} N(y\ x\langle x := N\rangle) \\
& \rightarrow_{Varx} N(y\ N)
\end{array}
$$

3.1.1 Simulação e Correção

Mostraremos que o λx-cálculo possui a propriedade de simulação da regra β e da correção. As provas são feitas através da indução na estrutura de termos.

Proposição 3 (Simulação de β em λx) *Dados $M, N \in \Lambda$. Se $M \rightarrow_\beta N$ então $M \rightarrow^+_{\lambda x} N$.*

Prova. Faremos a prova por indução sobre a estrutura de M.

- $M = ((\lambda x.P)\ Q)$. Se P é uma variável existem dois casos a considerar (BI):
    1. $P = x$, então $((\lambda x.x)\ Q) \rightarrow_\beta x\{x \leftarrow Q\} = Q$ pela Definição 11, e $(\lambda x.x)Q \rightarrow_{Beta} x\langle x := Q\rangle \rightarrow_{Var} Q$,
    2. $P = y$, $(\lambda x.y)Q \rightarrow_\beta y\{x \leftarrow Q\} = y$ pela Definição 11, e $(\lambda x.y)Q \rightarrow_{Beta} y\langle x := Q\rangle \rightarrow_{Gc} y$,

3. $P = (R\ S)$, $((\lambda x.(R\ S))\ Q) \to_\beta (R\ S)\{x \leftarrow Q\} = (R\{x \leftarrow Q\}\ \ S\{x \leftarrow Q\})$ pela Definição 11, e

$$((\lambda x.(R\ S))\ Q) \to_{Beta} (R\ S)\langle x := Q\rangle \to_{App} (R\langle x := Q\rangle\ \ S\langle x := Q\rangle)$$

$$\overset{+}{\to}{}^{HI}_{\lambda \mathrm{x}}\ (R\{x \leftarrow Q\}\ \ S\{x \leftarrow Q\}),$$

4. $P = \lambda y.R$, $M \to_\beta (\lambda y.R)\{x \leftarrow Q\} = \lambda y.R\{x \leftarrow Q\}$ com $y \neq x$, e $M \to_{Beta}$ $(\lambda y.R)\langle x := Q\rangle \to_{Lam} \lambda y.R\langle x := Q\rangle \overset{+}{\to}{}^{HI}_{\lambda \mathrm{x}} \lambda y.R\{x \leftarrow Q\}$,

- se a redução é interna, então os seguintes casos podem ocorrer:

  - $(R\ S) \to_\beta (R'\ S)$ com $R \to_\beta R'$, então pela HI $R \overset{+}{\to}_{\lambda \mathrm{x}} R'$ assim $(R\ S) \overset{+}{\to}_{\lambda \mathrm{x}}$ $(R'\ S)$,

  - $(R\ S) \to_\beta (R\ S')$ com $S \to_\beta S'$, então pela HI $S \overset{+}{\to}_{\lambda \mathrm{x}} S'$ assim $(R\ S) \overset{+}{\to}_{\lambda \mathrm{x}}$ $(R\ S')$,

  - $\lambda x.P \to_\beta \lambda x.P'$ com $P \to_\beta P'$, então pela HI $P \overset{+}{\to}_{\lambda \mathrm{x}} P'$ assim $\lambda x.P \overset{+}{\to}_{\lambda \mathrm{x}}$ $\lambda x.P'$.

$\square$

Proposição 4 (Correção) *Sejam $M \in \Lambda$ e $N \in \Lambda_{\mathrm{x}}$, se $M \to_{Beta} N$ então $M \underset{\beta}{\twoheadrightarrow} \mathrm{x}(N)$.*

Prova. Faremos a prova por indução sobre a estrutura de M.

- $M = ((\lambda x.P)\ Q)$.

  1. Se P é uma variável existem dois casos a considerar (BI):

     (a) $P = x$, então $((\lambda x.x)\ Q) \to_{Beta} x\langle x := Q\rangle$ e $\mathrm{x}(x\langle x := Q\rangle) = Q$ pois $x\langle x := Q\rangle \to_{Var} Q$ e

     $$(\lambda x.x)Q \underset{\beta}{\to} x\{x \leftarrow Q\} = Q$$

     pela Definição 11,

     (b) $P = y$, então $(\lambda x.y)Q \to_{Beta} y\langle x := Q\rangle \to_{Gc} y$ e $(\lambda x.y)Q \underset{\beta}{\to} y\{x \leftarrow Q\} =$ y pela Definição 11,

  2. $P = (R\ S)$, então $((\lambda x.(R\ S))\ Q) \to_{Beta} (R\ S)\langle x := Q\rangle \to_{App}$

     $$(R\langle x := Q\rangle\ \ S\langle x := Q\rangle)$$

e $((\lambda x.(R\ S))\ Q) \to_\beta (R\ S)\{x \leftarrow Q\} = (R\{x \leftarrow Q\}\ S\{x \leftarrow Q\})$

$$\underset{\beta}{\twoheadrightarrow}^{HI}(R\langle x := Q\rangle\ S\langle x := Q\rangle),$$

3. $P = \lambda y.R$, $M \to_{Beta} (\lambda y.R)\langle x := Q\rangle \to_{Lam} \lambda y.R\langle x := Q\rangle$ com $y \neq x$, e $M \underset{\beta}{\to} (\lambda y.R)\{x \leftarrow Q\} = \lambda y.R\{x \leftarrow Q\} \underset{\beta}{\twoheadrightarrow}^{HI} \lambda y.R\langle x := Q\rangle$,

- se a redução é interna, então os seguintes casos podem ocorrer:

  - $(R\ S) \to_{Beta} (R'\ S)$ com $R \to_{Beta} R'$, então pela HI $R \underset{\beta}{\twoheadrightarrow} R'$ assim

    $$(R\ S) \underset{\beta}{\twoheadrightarrow} (R'\ S),$$

  - $(R\ S) \to_{Beta} (R\ S')$ com $S \to_{Beta} S'$, então pela HI $S \underset{\beta}{\twoheadrightarrow} S'$ assim

    $$(R\ S) \underset{\beta}{\twoheadrightarrow} (R\ S'),$$

  - $\lambda x.P \to_{Beta} \lambda x.P'$ com $P \to_{Beta} P'$, então pela HI $P \underset{\beta}{\twoheadrightarrow} P'$ assim

    $$\lambda x.P \underset{\beta}{\twoheadrightarrow} \lambda x.P'.$$

$\square$

3.2 O $\lambda\upsilon$-cálculo

Nesta seção veremos a definição do $\lambda\upsilon$-cálculo e algumas propriedades. Iremos desenvolver lemas adicionais para provar a correção e a simulação. Nas referências [3] e [9] podem ser encontrados mais detalhes sobre este cálculo. E na referência [9] pode se encontrar a Proposição 5, que é a principal propriedade do $\lambda\upsilon$ que permitiu a Ariel Arbiser provar vários resultados sobre a indecibilidade do $\lambda\upsilon$-cálculo.

Definição 19 ($\lambda\upsilon$-cálculo) *A sintaxe do $\lambda\upsilon$-cálculo é especificada como:*

Termos	$M ::= \underline{\mathbf{n}}\ \mid\ (M\ \ M)\ \mid\ (\lambda M)\ \mid\ M[s]$ *onde* $n \in \mathbb{N}^*$
Substituição	$s ::= M/\ \mid\ \uparrow\ \mid\ \Uparrow(s)$

Chamamos de Λ_υ^t o conjunto de $\lambda\upsilon$-termos, e de Λ_υ^s o conjunto de $\lambda\upsilon$-substituições. E $\Lambda_\upsilon = \Lambda_\upsilon^t \cup \Lambda_\upsilon^s$.

Assim como no λx, termos sem ocorrência de operadores de substituição são chamados termos puros. Um termo da forma $a[s]$ é chamado de *fecho*, onde o sub-termo a é chamado cabeça.

As regras do $\lambda\upsilon$-cálculo são dadas abaixo:

$$
\begin{array}{lll}
(B) & (\lambda M)N & \longrightarrow M[N/] \\
(App) & (MN)[s] & \longrightarrow M[s]N[s] \\
(Lam) & (\lambda M)[s] & \longrightarrow \lambda M[\Uparrow(s)] \\
(Fvar) & \underline{\mathbf{1}}[M/] & \longrightarrow M \\
(Rvar) & (\underline{\mathbf{n}+\mathbf{1}})[M/] & \longrightarrow \underline{\mathbf{n}} \\
(FvarLift) & \underline{\mathbf{1}}[\Uparrow(s)] & \longrightarrow \underline{\mathbf{1}} \\
(RvarLift) & (\underline{\mathbf{n}+\mathbf{1}})[\Uparrow(s)] & \longrightarrow \underline{\mathbf{n}}[s][\uparrow] \\
(VarShift) & \underline{\mathbf{n}}[\uparrow] & \longrightarrow \underline{\mathbf{n}+\mathbf{1}}
\end{array}
$$

O conjunto de regras do $\lambda\upsilon$ sem a regra (B) compõe o denominado υ-cálculo. O $\lambda\upsilon$ é confluente, e υ é confluente e SN [2]. Nós denominamos com $\upsilon(u)$ a única forma υ-normal de um termo ou substituição u.

Representaremos por $Pos(M)$ o conjunto de todas posições do termo M.

Definição 20 (Posição) *Uma posição em um termo $M \in \Lambda_\upsilon^t$ é uma sequência de números do conjunto $\{1,2\}$, tal que*

- $M_{|\epsilon} = M$.
- *Se* $M_{|p} = N[s]$, *então* $M_{|p1} = N$ *e* $M_{|p2} = s$.
- *Se* $M_{|p} = (\lambda N)$, *então* $M_{|p1} = N$.
- *Se* $M_{|p} = (N\ P)$, *então* $M_{|p1} = N$ *e* $M_{|p2} = P$.
- *Se* $M_{|p} = N/$, *então* $M_{|p1} = N$.
- *Se* $M_{|p} = \Uparrow(s)$, *então* $M_{|p1} = s$.

$M_{|p}$ é chamado o sub-termo de M na posição p ou a ocorrência na posição p. As posições são comparadas pela ordem do prefixo. A posição p é um prefixo de q, se existe p' tal que $pp' = q$ [9].

Definição 21 *Um contexto em $\lambda\upsilon$ é um termo contendo um buraco $\Box$ em uma dada posição, isto é gerado pela sintaxe:*

$$
\begin{array}{ll}
\textbf{Contextos-termos} & C\Lambda_\upsilon^t ::= \Box \mid C\Lambda_\upsilon^t\ \Lambda_\upsilon^t \mid \Lambda_\upsilon^t\ C\Lambda_\upsilon^t \mid \lambda\ C\Lambda_\upsilon^t \mid C\Lambda_\upsilon^t[\Lambda_\upsilon^s] \mid \Lambda_\upsilon^t[C\Lambda_\upsilon^s] \\
\textbf{Contextos-substituições} & C\Lambda_\upsilon^s ::= C\Lambda_\upsilon^t/ \ \mid\ \Uparrow(C\Lambda_\upsilon^s)
\end{array}
$$

onde os termos e as substituições são os mesmos definidos no começo desta seção.

Exemplo 5 *Neste exemplo vemos que o termo obtido por uma β-redução, também é obtido por reduções em $\lambda\upsilon$.*

$$(\lambda\underline{\mathbf{1}}(\underline{\mathbf{3}}\ \underline{\mathbf{1}}))N \quad \to_\beta N(\underline{\mathbf{2}}\ N)$$

$$\begin{aligned}(\lambda\underline{\mathbf{1}}(\underline{\mathbf{3}}\ \underline{\mathbf{1}}))N \quad &\to_B \underline{\mathbf{1}}(\underline{\mathbf{3}}\ \underline{\mathbf{1}})[N/] \\ &\to_{App} \underline{\mathbf{1}}[N/](\underline{\mathbf{3}}\ \underline{\mathbf{1}})[N/] \\ &\to_{App} \underline{\mathbf{1}}[N/](\underline{\mathbf{3}}[N/]\ \underline{\mathbf{1}}[N/]) \\ &\to_{Fvar} \underline{\mathbf{1}}[N/]\ (\underline{\mathbf{3}}[N/]\ N) \\ &\to_{Fvar} N\ (\underline{\mathbf{3}}[N/]\ N) \\ &\to_{Rvar} N\ (\underline{\mathbf{2}}\ N)\end{aligned}$$

Definição 22 (Substituição em $\lambda\upsilon$) *Para $s \in \Lambda^s_\upsilon$ definimos as seguintes substituições:*

$$\begin{aligned}\Uparrow^0 (s) \quad &= \quad s \\ \Uparrow^{k+1} (s) \quad &= \quad \Uparrow (\Uparrow^k (s)) \qquad k \geq 0\end{aligned}$$

Toda substituição em Λ^s_υ tem alguma das formas: $\Uparrow^k (M/)$, tal que $M \in \Lambda^t_\upsilon$, ou $\Uparrow^k (\uparrow)$ tal que $k \geq 0$ [2]. Pois se $k = 0$ temos $\Uparrow^0 (M/) = (M/)$, $\Uparrow^0 (\uparrow) = (\uparrow)$ e para $k \neq 0$ temos, pela Definição 19, que as composições de substituições só podem ser feitas com vários $\Uparrow$ e um $(M/)$ ou um $(\uparrow)$.

3.2.1 Simulação e Correção de $\lambda\upsilon$-cálculo

Para mostrar a correção e simulação do $\lambda\upsilon$-cálculo usaremos o artigo [9], no qual são provadas estas propriedades, sendo feito aqui com mais detalhes. E para isso desenvolvemos os Lemas 1 e 2 para provar $((\lambda M)\ N) \underset{\beta}{\to} \sigma_0(M, N)$, que será usado na prova destas propriedades.

Será necessária a introdução de uma nova notação para fazermos a prova, que será apresentada abaixo:

Definição 23 *Dados $M, N, P \in \Lambda$ com $k, i \geq 0$, temos*

$$\begin{aligned}\sigma_k((M\ P), N) &= \sigma_k(M, N)\sigma_k(P, N) \\ \sigma_k(\lambda M, N) &= \lambda(\sigma_{k+1}(M, N))\end{aligned} \qquad \sigma_k(\underline{\mathbf{n}}, N) = \begin{cases} \underline{\mathbf{n-1}} & se\ \ n > k+1 \\ \tau^k_0(N) & se\ \ n = k+1 \\ \underline{\mathbf{n}} & se\ \ n \leq k \end{cases}$$

onde

$$\begin{aligned}\tau^k_i(M\ N) &= \tau^k_i(M)\tau^k_i(N) \\ \tau^k_i(\lambda M) &= \lambda(\tau^k_{i+1}(M))\end{aligned} \qquad \tau^k_i(\underline{\mathbf{n}}) = \begin{cases} \underline{\mathbf{n+k}} & se\ \ n > i \\ \underline{\mathbf{n}} & se\ \ n \leq i \end{cases}$$

Note que $\tau^k_i \circ \tau^m_i = \tau^{k+m}_i$ e $\tau^0_i(M) = M$, para todo $i, k, m \geq 0$ e $M \in \Lambda^t_\upsilon$.

Lema 1 $\tau^{m-1}_{i-1}(P) = renomear_{m,i}(P)$ [1] *para todo $m, i \geq 1$ e $P \in \Lambda$.*

[1] A função *renomear* está descrita na Definição 12

Prova. Por indução na estrutura de P, temos:

- $P = \underline{\mathbf{n}}$, implica que

$$\tau_{i-1}^{m-1}(\underline{\mathbf{n}}) = \begin{cases} \underline{\mathbf{n}} & se \;\; n < i \\ \underline{\mathbf{n+m-1}} & se \;\; n \geq i \end{cases}$$

e

$$renomear_{m,i}(\underline{\mathbf{n}}) = \begin{cases} \underline{\mathbf{n}} & se \;\; n < i \\ \underline{\mathbf{n+m-1}} & se \;\; n \geq i \end{cases}$$

- $P = (M\ N)$, implica que

$$\tau_{i-1}^{m-1}(M\ N) = (\tau_{i-1}^{m-1}(M)\ \tau_{i-1}^{m-1}(N)) =^{HI} (renomear_{m,i}(M)\ renomear_{m,i}(N))$$
$$= renomear_{m,i}(M\ N)$$

- $P = (\lambda N)$, implica que

$$\tau_{i-1}^{m-1}(\lambda N) = (\lambda \tau_{i}^{m-1}(N)) =^{HI} (\lambda renomear_{m,i+1}(N)) = renomear_{m,i}((\lambda N))$$

$\square$

Lema 2 *$\sigma_k(P, Q) = P\{\underline{\mathbf{k+1}} \leftarrow Q\}$ para todo $k \geq 0$ e $P, Q \in \Lambda$.*

Prova. Por indução na estrutura de P, temos:

- $P = \underline{\mathbf{n}}$, então analisaremos três casos que dependem do valor de $\underline{\mathbf{n}}$ em comparação com k:
  - se $n > k+1$ então, pela Definição 23, $\sigma_k(\underline{\mathbf{n}}, Q) = \underline{\mathbf{n-1}}$ e, pela Definição 12, $\underline{\mathbf{n}}\{\underline{\mathbf{k+1}} \leftarrow Q\} = \underline{\mathbf{n-1}}$;
  - se $n = k+1$ então, pela Definição 23, $\sigma_k(\underline{\mathbf{n}}, Q) = \tau_0^k(Q)$ e, pela Definição 12, $\underline{\mathbf{n}}\{\underline{\mathbf{k+1}} \leftarrow Q\} = renomear_{n,1}(Q)$ e, pelo Lema 1, $\tau_0^k(Q) = renomear_{n,1}(Q)$;
  - se $n \leq k$ então, pela Definição 23, $\sigma_k(\underline{\mathbf{n}}, Q) = \underline{\mathbf{n}}$ e, pela Definição 12, $\underline{\mathbf{n}}\{\underline{\mathbf{k+1}} \leftarrow Q\} = \underline{\mathbf{n}}$, para $n < k+1$. Observe que a desigualdade $n < k+1$ implica que $n \leq k$. Portanto temos que $\sigma_k(\underline{\mathbf{n}}, Q) = \underline{\mathbf{n}}\{\underline{\mathbf{k+1}} \leftarrow Q\}$ para $n \leq k$.

Resumindo temos:

$$\sigma_k(\underline{\mathbf{n}}, Q) = \begin{cases} \underline{\mathbf{n-1}} & se \;\; n > k+1 \\ \tau_0^k(Q) & se \;\; n = k+1 \\ \underline{\mathbf{n}} & se \;\; n \leq k \end{cases}$$

e

$$\underline{\mathbf{n}}\{\underline{\mathbf{k+1}} \leftarrow Q\} = \begin{cases} \underline{\mathbf{n-1}} & se \;\; n > k+1 \\ renomear_{n,1}(Q) & se \;\; n = k+1 \\ \underline{\mathbf{n}} & se \;\; n < k+1, \end{cases}$$

- $P = (M\ N)$, então

$$\sigma_k((M\ N), Q) = \sigma_k(M\ Q)\sigma_k(N\ Q) =^{HI} (M\{\underline{\mathbf{k+1}} \leftarrow Q\}\ N\{\underline{\mathbf{k+1}} \leftarrow Q\})$$
$$= (M\ N)\{\underline{\mathbf{k+1}} \leftarrow Q\},$$

- $P = (\lambda N)$, então

$$\sigma_k((\lambda N), Q) = (\lambda \sigma_{k+1}(N, Q)) =^{HI} (\lambda N\{\underline{\mathbf{k+2}} \leftarrow Q\}) = (\lambda N)\{\underline{\mathbf{k+1}} \leftarrow Q\}.$$

□

Assim usaremos a notação de substituição σ_0, onde a definição clássica da regra β do cálculo λ, é

$$((\lambda M)\ N) \underset{\beta}{\rightarrow} \sigma_0(M, N)$$

onde σ_0 é a ocorrência em 0 da aplicação σ_k definida acima.

Introduzimos também a aplicação μ que trata termos impuros com σ_k e τ_k^i.

Definição 24 *[9] Para todo $M, N \in \Lambda_v^t$ com $k \geq 0$ e $n \geq 1$ temos que:*

$$\begin{array}{lcl} \mu(M[\Uparrow^k (N/)]) & = & \sigma_k(\mu(M), \mu(N)) \\ \mu(M[\Uparrow^k (\uparrow)]) & = & \tau_k^1(\mu(M)) \\ \mu(\underline{\mathbf{n}}) & = & \underline{\mathbf{n}} \\ \mu(M\ N) & = & \mu(M)\mu(N) \\ \mu(\lambda M) & = & (\lambda \mu(M)) \end{array}$$

Observe que se M é um termo puro, então $\mu(M) = M$, em particular, $\mu(\upsilon(M)) = \upsilon(M)$.

Lema 3 *[9] Sejam $M, N \in \Lambda_v^t$ tal que $M \underset{\upsilon}{\rightarrow} N \;\; \Rightarrow \;\; \mu(M) = \mu(N)$.*

Prova. Para a prova é suficiente considerar reduções na raiz de M. Assim foi observado os casos das regras de υ.

1. se $M = (P\ Q)[s] \rightarrow_{App} (P[s]\ Q[s])$ e

   (a) $s = \Uparrow^k (R/)$, então

   $$\mu((P\ Q)[\Uparrow^k (R/)]) = \sigma_k(\mu((P\ Q)), \mu(R)) = \sigma_k(\mu(P)\mu(Q), \mu(R)),$$

   e

   $$\mu((P[\Uparrow^k (R/)]\ Q[\Uparrow^k (R/)])) = \mu(P[\Uparrow^k (R/)])\mu(Q[\Uparrow^k (R/)])$$
   $$= \sigma_k(\mu(P), \mu(R))\sigma_k(\mu(Q), \mu(R)) = \sigma_k((\mu(P)\ \mu(Q)), \mu(R))$$

   (b) $s = \Uparrow^k (\uparrow)$, então

   $$\mu((P\ Q)[\Uparrow^k (\uparrow)]) = \tau_k^1(\mu((P\ Q))) = \tau_k^1(\mu(P)\mu(Q)),$$

   e

   $$\mu((P[\Uparrow^k (\uparrow)]\ Q[\Uparrow^k (\uparrow)])) = (\mu(P[\Uparrow^k (\uparrow)])\ \mu(Q[\Uparrow^k (\uparrow)])) = \tau_k^1(\mu(P))\tau_k^1(\mu(Q))$$
   $$= \tau_k^1(\mu(P)\mu(Q))$$

2. se $M = (\lambda P)[s] \rightarrow_{Lam} (\lambda P[\Uparrow (s)])$ e

   (a) $s = \Uparrow^k (R/)$, então

   $$\mu((\lambda P)[\Uparrow^k (R/)]) = \sigma_k(\mu(\lambda P), \mu(R))$$
   $$= \sigma_k((\lambda\mu(P)), \mu(R)) = (\lambda\sigma_{k+1}(\mu(P), \mu(R)))$$

   e

   $$\mu(\lambda P[\Uparrow^{k+1} (R/)]) = (\lambda\mu(P[\Uparrow^{k+1} (R/)])) = (\lambda\sigma_{k+1}(\mu(P), \mu(R)))$$

   (b) $s = \Uparrow^k (\uparrow)$, então

   $$\mu((\lambda P)[\Uparrow^k (\uparrow)]) = \tau_k^1(\mu(\lambda P)) = \tau_k^1(\lambda\mu(P)) = (\lambda\tau_{k+1}^1(\mu(P)))$$

   e

   $$\mu(\lambda P[\Uparrow^{k+1} (\uparrow)]) = (\lambda\mu(P[\Uparrow^{k+1} (\uparrow)])) = (\lambda\tau_{k+1}^1(\mu(P)))$$

3. se $M = \underline{\mathbf{1}}[N/] \rightarrow_{FVar} N$ então

   $$\mu(\underline{\mathbf{1}}[N/]) = \sigma_0(\mu(\underline{\mathbf{1}}), \mu(N)) = \sigma_0(\underline{\mathbf{1}}, \mu(N)) = \tau_0^0(\mu(N)) = \mu(N)$$

4. se $M = \underline{\mathbf{n+1}}[R/] \to_{RVar} \underline{\mathbf{n}}$ então

$$\mu(\underline{\mathbf{n+1}}[R/]) = \sigma_0(\mu(\underline{\mathbf{n+1}}), \mu(R)) = \sigma_0(\underline{\mathbf{n+1}}, \mu(R)) = \underline{\mathbf{n}}$$

e $\mu(\underline{\mathbf{n}}) = \underline{\mathbf{n}}$

5. se $M = \underline{\mathbf{1}}[\Uparrow (s)] \to_{FVarLift} \underline{\mathbf{1}}$ e
   (a) $s = \Uparrow^k (P/)$, então

$$\mu(\underline{\mathbf{1}}[\Uparrow^{k+1} (P/)]) = \sigma_{k+1}(\mu(\underline{\mathbf{1}}), \mu(P)) = \sigma_{k+1}(\underline{\mathbf{1}}, \mu(P)) = \underline{\mathbf{1}}$$

   e $\mu(\underline{\mathbf{1}}) = \underline{\mathbf{1}}$

   (b) $s = \Uparrow^k (\uparrow)$, então

$$\mu(\underline{\mathbf{1}}[\Uparrow^{k+1} (\uparrow)]) = \tau^1_{k+1}(\mu(\underline{\mathbf{1}})) = \tau^1_{k+1}(\underline{\mathbf{1}}) = \underline{\mathbf{1}}$$

e $\mu(\underline{\mathbf{1}}) = \underline{\mathbf{1}}$

6. se $M = \underline{\mathbf{n+1}}[\Uparrow (s)] \to_{RVarLift} \underline{\mathbf{n}}[s][\uparrow]$ e
   (a) $s = \Uparrow^k (R/)$, então

$$\mu(\underline{\mathbf{n+1}}[\Uparrow^{k+1} (R/)]) = \sigma_{k+1}(\mu(\underline{\mathbf{n+1}}), \mu(R)) = \sigma_{k+1}(\underline{\mathbf{n+1}}, \mu(R))$$

   assim $\sigma_{k+1}(\underline{\mathbf{n+1}}, \mu(R)) = \begin{cases} \underline{\mathbf{n}} \text{ se } n > k+1 \\ \tau_0^{k+1}(\mu(R)) \text{ se } n = k+1 \\ \underline{\mathbf{n+1}} \text{ se } n \leq k \end{cases}$

   e

$$\begin{aligned}\mu(\underline{\mathbf{n}}[\Uparrow^k (R/)][\uparrow]) &= \mu(\underline{\mathbf{n}}[\Uparrow^k (R/)][\Uparrow^0 (\uparrow)]) = \tau^1_0(\mu(\underline{\mathbf{n}}[\Uparrow^k (R/)])\\ &= \tau^1_0(\sigma_k(\mu(\underline{\mathbf{n}}), \mu(R))) = \tau^1_0(\sigma_k(\underline{\mathbf{n}}, \mu(R)))\end{aligned}$$

   como

$$\sigma_k(\underline{\mathbf{n}}, \mu(R)) = \begin{cases} \underline{\mathbf{n-1}} \text{ se } n > k+1 \\ \tau_0^k(\mu(R)) \text{ se } n = k+1 \\ \underline{\mathbf{n}} \text{ se } n \leq k \end{cases}$$

   temos que $\tau^1_0(\underline{\mathbf{n-1}}) = \begin{cases} \underline{\mathbf{n}} & se\ n-1 > 0 \\ \underline{\mathbf{n-1}} & se\ n-1 \leq 0 \end{cases}$
   como $\underline{\mathbf{n-1}} \geq 1$, por se tratar de um numeral de De Bruijn, temos que $n \geq 2$ assim temos que $n > 1$, portanto $\tau^1_0(\underline{\mathbf{n-1}}) = \underline{\mathbf{n}}$. No segundo caso obtemos

$$\tau^1_0(\tau^k_0(\mu(R))) = \tau^{k+1}_0(\mu(R)).$$

E por fim $\tau_0^1(\underline{\mathbf{n}}) = \begin{cases} \underline{\mathbf{n+1}} & se\ \ n > 0 \\ \\ \underline{\mathbf{n}} & se\ \ n \leq 0 \end{cases}$

obtendo-se $\tau_0^1(\underline{\mathbf{n}}) = \underline{\mathbf{n+1}}$ pela condição de $n \geq 1$, pois temos que $\underline{\mathbf{n}} > 0$.

(b) $s = \Uparrow^k (\uparrow)$, então

$$\mu(\underline{\mathbf{n+1}}[\Uparrow^{k+1} (\uparrow)]) = \tau_{k+1}^1(\mu(\underline{\mathbf{n+1}})) = \tau_{k+1}^1(\underline{\mathbf{n+1}}) = \begin{cases} \underline{\mathbf{n+2}} & se\ \ n > k \\ \\ \underline{\mathbf{n+1}} & se\ \ n \leq k \end{cases}$$

e temos que

$$\mu(\underline{\mathbf{n}}[\Uparrow^k (\uparrow)][\uparrow]) = \mu(\underline{\mathbf{n}}[\Uparrow^k (\uparrow)][\Uparrow^0 (\uparrow)]) = \tau_0^1(\mu(\underline{\mathbf{n}}[\Uparrow^k (\uparrow)])) = \tau_0^1(\tau_k^1(\mu(\underline{\mathbf{n}}))) = \tau_0^1(\tau_k^1(\underline{\mathbf{n}}))$$

e assim resolvendo $\tau_k^1(\underline{\mathbf{n}})$ temos,

$$\tau_k^1(\underline{\mathbf{n}}) = \begin{cases} \underline{\mathbf{n+1}} & se\ \ n > k \\ \\ \underline{\mathbf{n}} & se\ \ n \leq k \end{cases}$$

e aplicando τ_0^1 obtemos, se $n > k$

$$\tau_0^1(\underline{\mathbf{n+1}}) = \begin{cases} \underline{\mathbf{n+2}} & se\ \ n+1 > 0 \\ \\ \underline{\mathbf{n+1}} & se\ \ n \leq 0 \end{cases}$$

como $\underline{\mathbf{n+1}} > 0$, por se um numeral de de Bruijn, temos que $\tau_0^1(\underline{\mathbf{n+1}}) = \underline{\mathbf{n+2}}$. Para o caso de $n \leq k$, temos que

$$\tau_0^1(\underline{\mathbf{n}}) = \begin{cases} \underline{\mathbf{n+1}} & se\ \ n > 0 \\ \\ \underline{\mathbf{n}} & se\ \ n \leq 0 \end{cases}$$

e assim obtemos que $\tau_0^1(\underline{\mathbf{n}}) = \underline{\mathbf{n+1}}$, pois $n > 0$, por ser um numeral de de Bruijn.

7. se $M = \underline{\mathbf{n}}[\uparrow] \rightarrow_{VarShift} \underline{\mathbf{n+1}}$ então

$$\mu(\underline{\mathbf{n}}[\uparrow]) = \tau_0^1(\mu(\underline{\mathbf{n}})) = \tau_0^1(\underline{\mathbf{n}}) = \underline{\mathbf{n+1}}$$

e

$$\mu(\underline{\mathbf{n+1}}) = \underline{\mathbf{n+1}}$$

8. se $M = P[s_2] \underset{v}{\rightarrow} P'[s_2]$ (observe primeiramente que $P \underset{v}{\rightarrow} P'$ tal que $\mu(P) = \mu(P')$ é válido, bastando aplicar os casos anteriores) então

   (a) se $s_2 = \Uparrow^k (R/)$ temos

$$\mu(P[\Uparrow^k (R/)]) = \sigma_k(\mu(P), \mu(R)) =^{HI} \sigma_k(\mu(P'), \mu(R))$$

e

$$\mu(P'[\Uparrow^k (R/)]) = \sigma_k(\mu(P'), \mu(R))$$

(b) se $s_2 = \Uparrow^k (\uparrow)$ e assim

$$\mu(P[\Uparrow^k (\uparrow)]) = \tau_k^1(\mu(P)) =^{HI} \tau_k^1(\mu(P'))$$

e

$$\mu(P'[\Uparrow^k (\uparrow)]) = \tau_k^1(\mu(P'))$$

□

Proposição 5 (Simulação e correção) *Sejam $P, Q \in \Lambda, M, N, Q' \in \Lambda_\upsilon^t$, assim se:*

1. $M \underset{\upsilon}{\twoheadrightarrow} N \quad \Rightarrow \quad \mu(M) = \mu(N)$,
2. $\upsilon(M) \;=\; \mu(M)$,
3. $\upsilon(M[N/]) \;=\; \sigma_0(\mu(M), \mu(N))$,
4. $P \underset{\beta}{\rightarrow} Q$ *se somente se* $P \underset{B}{\rightarrow} Q'$ *tal que* $Q = \upsilon(Q')$.

Prova.

1. A prova será feita por indução no número n de passos de $M \overset{n}{\underset{\upsilon}{\rightarrow}} N$. Para isso observe pelo lema 3 que se $M \underset{\upsilon}{\rightarrow} N'$ vale, então temos que $\mu(M) = \mu(N')$, e assim temos que $M \overset{n+1}{\underset{\upsilon}{\rightarrow}} N = M \underset{\upsilon}{\rightarrow} N' \overset{n}{\underset{\upsilon}{\rightarrow}} N$, assim $\mu(M) =^{\text{lema } 3} \mu(N') =^{HI} \mu(N)$. Portanto temos $\mu(M) = \mu(N)$,
2. Temos $\upsilon(M) = M'$ tal que $M \underset{\upsilon}{\twoheadrightarrow} M'$ e assim pelo item 1 temos que $\mu(M) = \mu(M')$ e segue que $\mu(M') = \mu(\upsilon(M)) = \upsilon(M)$, pois $\upsilon(M)$ é puro. Portanto $\mu(M) = \upsilon(M)$.
3. Pelo item 2 temos que $\upsilon(M[N/]) = \mu(M[N/])$ então segue que

$$\mu(M[N/]) = \mu(M[\Uparrow^0 (N/)]) = \sigma_0(\mu(M), \mu(N)).$$

4. Seja $P = ((\lambda R)\ S) \underset{\beta}{\rightarrow} Q = \sigma_0(R, S)$ então, aplicando a regra B em P, obtemos

$$P = ((\lambda R)\ S) \underset{B}{\rightarrow} R[S/] = Q'$$

e aplicando o item 3, temos que $\upsilon(R[S/]) = \sigma_0(\mu(R), \mu(S)) = \sigma_0(R, S)$, pois $R, S \in \Lambda$, já que, são componentes de P. A outra parte da prova segue o mesmo argumento anterior, bastando aplicar o item 3. Assim concluimos a prova.

□

3.3 O $\lambda\sigma$-cálculo

Nesta seção estudaremos o $\lambda\sigma$-cálculo [1, 5], no qual iremos ver as propriedades como a simulação e a projeção. Estas que serão utilizadas na prova de resultados a respeito do estudo da expansibilidade deste cálculo. Aqui aprofundaremos a prova da Proposição 7, que é a simulação da regra β, para isso utilizamos a Proposição 6 e a definição da redução β que se encontram na referência [1]. Para maiores informações sobre o $\lambda\sigma$-cálculo consulte a referência [1].

3.3.1 Sintaxe

O $\lambda\sigma$-cálculo trabalha sobre dois tipos de expressões, os *termos (próprios)* e as *substituições*; como pode-se ver a seguir:

Termos	M, N	$::=$	$\underline{\mathbf{1}} \mid (M\ N) \mid \lambda M \mid M[s]$
Substituições	s, s'	$::=$	$id \mid \uparrow \mid M \cdot s \mid s \circ s'$

A operação de substituição no $\lambda\sigma$-cálculo é "construída"sob a forma de *fecho* de termos (i.e., termos da forma $M[s]$).

Chamaremos de Λ^t_σ o conjunto de $\lambda\sigma$-termos, e Λ^s_σ o conjunto de $\lambda\sigma$-substituições. Para toda substituição s definimos a *interação de composição de* s indutivamente como $s^1 = s$ e $s^{n+1} = s \circ s^n$, e usamos s^0 para denominar id. Observe que somente o índice $\underline{\mathbf{1}}$ de de Bruijn é usado, mas podemos codificar $\underline{\mathbf{n}}$ por $\underline{\mathbf{1}}[\uparrow^{n-1}]$. Assim podemos também codificar os índices como: $1, 1[\uparrow], 1[\uparrow \circ \uparrow], 1[\uparrow \circ(\uparrow \circ \uparrow)], \ldots$, e então termos puros incluem clausuras, composições e $\uparrow$'s.

O sistema de reescrita $\lambda\sigma$ é dado por [6]:

(Beta)	$(\lambda M)\ N \to M[N \cdot id]$
(App)	$(M\ N)[s] \to (M[s]\ N[s])$
(Abs)	$(\lambda M)[s] \to \lambda M[\underline{\mathbf{1}} \cdot (s \circ \uparrow)]$
(Clos)	$M[s][t] \to M[s \circ t]$
(VarCons)	$\underline{\mathbf{1}}[M.s] \to M$
(Id)	$M[id] \to M$
(Assoc)	$(s_1 \circ s_2) \circ t \to s_1 \circ (s_2 \circ t)$
(Map)	$(M \cdot s) \circ t \to M[t] \cdot (s \circ t)$
(IdL)	$id \circ s \to s$
(IdR)	$s \circ id \to s$
(ShiftCons)	$\uparrow \circ (M \cdot s) \to s$
(VarShift)	$\underline{\mathbf{1}} \cdot \uparrow \to id$
(SCons)	$\underline{\mathbf{1}} \cdot [s](\uparrow \circ s) \to s$

Nós usamos o $\lambda\sigma$ para designar o conjunto dessas regras. E assim, utilizaremos $\underset{\lambda\sigma}{\twoheadrightarrow}$ para representar o fecho transitivo-reflexivo com aplicação do conjunto de regras do $\lambda\sigma$-cálculo.

O sistema de reescrita $\lambda\sigma$-cálculo é confluente [1]. O σ-cálculo, i.e., $\lambda\sigma$ sem (Beta), é confluente e terminante [1]. Nós denominamos com $\sigma(u)$ a única forma σ-normal de um termo ou substituição.

Exemplo 6 *Damos um exemplo de uma redução utilizando as regras de $\lambda\sigma$.*

$$\begin{array}{lll}
\underline{(\lambda\underline{\mathbf{1}}(\underline{\mathbf{2}}\ \underline{\mathbf{1}})N)} & \rightarrow_{Beta} & \underline{(\underline{\mathbf{1}}\ (\underline{\mathbf{2}}\ \underline{\mathbf{1}}))[N\cdot id]} \\
 & \rightarrow_{App} & \underline{\mathbf{1}}[N\cdot id]\underline{(\underline{\mathbf{2}}\ \underline{\mathbf{1}})[N\cdot id]} \\
 & \rightarrow_{App} & \underline{\mathbf{1}}[N\cdot id](\underline{\mathbf{2}}[N\cdot id]\ \underline{\mathbf{1}}[N\cdot id]) \\
 & = & \underline{\mathbf{1}}[N\cdot id](\underline{\mathbf{1}}\underline{[\uparrow][N\cdot id]}\ \underline{\mathbf{1}}[N\cdot id]) \\
 & \rightarrow_{Clos} & \underline{\mathbf{1}}[N\cdot id](\underline{\mathbf{1}}[\underline{\uparrow\circ(N\cdot id)}]\ \underline{\mathbf{1}}[N\cdot id]) \\
 & \rightarrow_{ShiftCons} & \underline{\mathbf{1}}[N\cdot id](\underline{\underline{\mathbf{1}}[id]}\ \underline{\mathbf{1}}[N\cdot id]) \\
 & \rightarrow_{Id} & \underline{\underline{\mathbf{1}}[N\cdot id]}(\underline{\mathbf{1}}\ \underline{\mathbf{1}}[N\cdot id]) \\
 & \rightarrow_{VarCons} & N(\underline{\mathbf{1}}\ \underline{\underline{\mathbf{1}}[N\cdot id]}) \\
 & \rightarrow_{VarCons} & N(\underline{\mathbf{1}}\ N)
\end{array}$$

3.3.2 Simulação e Projeção

A propriedade de simulação é feita em cima do sistema definito abaixo da redução β, o qual é utilizado na Proposição 6. E temos nesta seção a propriedade de projeção, que apresentamos a referência da prova da mesma.

A definição precisa da redução β, no estilo de De Bruijn, é como a seguir [1]:

$$((\lambda a)\ b) \rightarrow_\beta a\,\langle \underline{\mathbf{1}} \leftarrow b, \underline{\mathbf{2}} \leftarrow \underline{\mathbf{1}}, \ldots \underline{\mathbf{n+1}} \leftarrow \underline{\mathbf{n}}, \ldots\rangle$$

onde a substituição meta-nível $\langle\ldots\rangle$ é definida indutivamente pelo uso das regras:

$$\underline{\mathbf{n}}\,\langle \underline{\mathbf{1}} \leftarrow a_1, \ldots, \underline{\mathbf{n}} \leftarrow a_n, \ldots\rangle = a_n,$$

$$\frac{a\,\langle \underline{\mathbf{1}} \leftarrow a_1, \ldots, \underline{\mathbf{n}} \leftarrow a_n, \ldots\rangle = a' \quad b\,\langle \underline{\mathbf{1}} \leftarrow a_1, \ldots, \underline{\mathbf{n}} \leftarrow a_n, \ldots\rangle = b'}{(a\ b)\,\langle \underline{\mathbf{1}} \leftarrow a_1, \ldots, \underline{\mathbf{n}} \leftarrow a_n, \ldots\rangle = (a'\ b')}$$

$$\frac{a_i\,\langle \underline{\mathbf{1}} \leftarrow \underline{\mathbf{2}}, \ldots, \underline{\mathbf{n}} \leftarrow \underline{\mathbf{n+1}}, \ldots\rangle = a'_i \quad a\,\langle \underline{\mathbf{1}} \leftarrow \underline{\mathbf{1}}, \underline{\mathbf{2}} \leftarrow a'_1, \ldots, \underline{\mathbf{n+1}} \leftarrow a'_n, \ldots\rangle = a'}{(\lambda a)\,\langle \underline{\mathbf{1}} \leftarrow a_1, \ldots, \underline{\mathbf{n}} \leftarrow a_n, \ldots\rangle = (\lambda a')}$$

Proposição 6 *Se existem m e p tal que $a_{m+q} = \underline{\mathbf{p+q}}$, $\forall q \geq 1$, e $a\,\langle \underline{\mathbf{1}} \leftarrow a_1, \ldots, \underline{\mathbf{n}} \leftarrow a_n, \ldots\rangle = b$ é obtido utilizando o sistema acima, então $\sigma(a[a_1 \cdot a_2 \cdot \ldots \cdot a_m \cdot \uparrow^p]) = b$.*

Prova. A prova será feita por meio de indução sobre a estrutura do termo a.

1. se $a = \underline{\mathbf{n}}$ então temos a seguinte estrutura, de acordo com o sistema acima:

$$\underline{\mathbf{n}}\,\langle \underline{\mathbf{1}} \leftarrow a_1, \ldots, \underline{\mathbf{n}} \leftarrow a_n, \ldots\rangle = a_n,$$

se $n \leq m$ então
$$\sigma(\underline{\mathbf{n}}[a_1 \cdot a_2 \cdot \ldots \cdot a_m \cdot \uparrow^p]) = a_n.$$
Mas se $n > m$ então temos pela redução σ que
$$\underline{\mathbf{n}}[a_1 \cdot a_2 \cdot \ldots \cdot a_m \cdot \uparrow^p] \underset{\sigma}{\overset{+}{\rightarrow}} \underline{\mathbf{n-m+p}}$$
e por HI temos que $a_n = a_{n-m+m}$ assim tomando $q = n - m$ temos $a_n = a_{q+m} = \underline{\mathbf{n-m+p}}$.

2. se $a = (N\ M)$ então temos: $(N\ M)\langle \underline{\mathbf{1}} \leftarrow a_1, \ldots, \underline{\mathbf{n}} \leftarrow a_n, \ldots\rangle$
$$= (N\langle \underline{\mathbf{1}} \leftarrow a_1, \ldots, \underline{\mathbf{n}} \leftarrow a_n, \ldots\rangle\ M\langle \underline{\mathbf{1}} \leftarrow a_1, \ldots, \underline{\mathbf{n}} \leftarrow a_n, \ldots\rangle) = (N'\ M')$$
por HI temos que se
$$N\langle \underline{\mathbf{1}} \leftarrow a_1, \ldots, \underline{\mathbf{n}} \leftarrow a_n, \ldots\rangle = N',\ \text{então}\ \sigma(N[a_1 \cdot a_2 \cdot \ldots \cdot a_m \cdot \uparrow^p]) = N'$$
$$\text{e se}\quad M\langle \underline{\mathbf{1}} \leftarrow a_1, \ldots, \underline{\mathbf{n}} \leftarrow a_n, \ldots\rangle = M'\ \text{ então }\ \sigma(M[a_1 \cdot a_2 \cdot \ldots \cdot a_m \cdot \uparrow^p]) = M'$$
e assim
$$(N\ M)[a_1 \cdot a_2 \cdot \ldots \cdot a_m \cdot \uparrow^p] \underset{\sigma}{\overset{+}{\rightarrow}}$$
$$\underset{\sigma}{\overset{+}{\rightarrow}} (N[a_1 \cdot a_2 \cdot \ldots \cdot a_m \cdot \uparrow^p]\ M[a_1 \cdot a_2 \cdot \ldots \cdot a_m \cdot \uparrow^p]) \underset{\sigma}{\overset{+}{\rightarrow}}{}^{HI} (N'\ M')$$

3. se $a = (\lambda a)$ então $(\lambda a)\langle \underline{\mathbf{1}} \leftarrow a_1, \ldots, \underline{\mathbf{n}} \leftarrow a_n, \ldots\rangle$
$$= (\lambda a\langle \underline{\mathbf{1}} \leftarrow \underline{\mathbf{1}}, \underline{\mathbf{2}} \leftarrow a_1\langle \underline{\mathbf{1}} \leftarrow \underline{\mathbf{2}}, \ldots, \underline{\mathbf{n}} \leftarrow \underline{\mathbf{n+1}}, \ldots\rangle, \ldots$$
$$\ldots, \underline{\mathbf{n+1}} \leftarrow a_n\langle \underline{\mathbf{1}} \leftarrow \underline{\mathbf{2}}, \ldots, \underline{\mathbf{n}} \leftarrow \underline{\mathbf{n+1}}, \ldots\rangle, \ldots\rangle) = (\lambda a')$$
por HI temos que se
$$a_i\langle \underline{\mathbf{1}} \leftarrow \underline{\mathbf{2}}, \ldots, \underline{\mathbf{n}} \leftarrow \underline{\mathbf{n+1}}, \ldots\rangle = a'_i \text{ então } \sigma(a_i[\underline{\mathbf{2}} \cdot \underline{\mathbf{3}} \cdot \ldots \cdot \underline{\mathbf{m}} \cdot \uparrow^p]) = a'_i$$
e se $a\langle \underline{\mathbf{1}} \leftarrow \underline{\mathbf{1}}, \underline{\mathbf{2}} \leftarrow a'_1, \ldots, \underline{\mathbf{n+1}} \leftarrow a'_n, \ldots\rangle = a'$ então $\sigma(a[\underline{\mathbf{1}} \cdot a'_1 \cdot a'_2 \cdot \ldots \cdot a'_m \cdot \uparrow^{p+1}]) = a'$
logo fazendo indução sobre os a_i's temos por HI para $m = 0$, $p = 1$ e $q = j$ com $j \geq 1$ que $a_j = \underline{\mathbf{j+1}}$ e assim
$$\sigma(a_j[\uparrow]) = a'_j$$
e aplicando a redução σ obtemos
$$(\lambda a)[a_1 \cdot a_2 \cdot \ldots \cdot a_m \cdot \uparrow^p] \underset{\sigma}{\overset{+}{\rightarrow}} (\lambda a[(\underline{\mathbf{1}} \cdot (a_1 \cdot a_2 \cdot \ldots \cdot a_m \cdot \uparrow^p \circ \uparrow))])$$
$$(\lambda a[\underline{\mathbf{1}} \cdot a_1[\uparrow] \cdot a_2[\uparrow] \cdot \ldots \cdot a_m[\uparrow] \cdot \uparrow^{p+1}]) \underset{\sigma}{\overset{+}{\rightarrow}}{}^{HI} (\lambda a[\underline{\mathbf{1}} \cdot a'_1 \cdot a'_2 \cdot \ldots \cdot a'_m \cdot \uparrow^{p+1}])$$
$$\underset{\sigma}{\overset{+}{\rightarrow}}{}^{HI} (\lambda a')$$
$\square$

Proposição 7 (Simulação) *Se* $M \to_\beta N$ *então* $M \xrightarrow{+}_{\lambda\sigma} N$.

Prova. Faremos a prova por indução sobre a estrutura de M. Então segue:

- se $M = ((\lambda U)\ V) \to_\beta U\,\langle \underline{\mathbf{1}} \leftarrow V, \underline{\mathbf{2}} \leftarrow \underline{\mathbf{1}}, \ldots \underline{\mathbf{n+1}} \leftarrow \underline{\mathbf{n}}, \ldots\rangle$, então os seguintes casos podem ocorrer:

  – se U é uma variável existem dois casos a considerar (BI):

    * $U = \underline{\mathbf{1}}$,

    $$M = ((\lambda\underline{\mathbf{1}})\ V) \to_\beta \underline{\mathbf{1}}\,\langle \underline{\mathbf{1}} \leftarrow V, \underline{\mathbf{2}} \leftarrow \underline{\mathbf{1}}, \ldots \underline{\mathbf{n+1}} \leftarrow \underline{\mathbf{n}}, \ldots\rangle = V.$$

    E aplicando as regras de $\lambda\sigma$ temos $M = ((\lambda\underline{\mathbf{1}})\ V) \to_{Beta} \underline{\mathbf{1}}[V \cdot id] \to_{VarCons} V$,

    * $U = \underline{\mathbf{n}}$, com $\underline{\mathbf{n}} \geq 2$,

    $$M = ((\lambda\underline{\mathbf{n}})\ V) \to_\beta \underline{\mathbf{n}}\,\langle \underline{\mathbf{1}} \leftarrow V, \underline{\mathbf{2}} \leftarrow \underline{\mathbf{1}}, \ldots, \underline{\mathbf{n}} \leftarrow \underline{\mathbf{n-1}}, \underline{\mathbf{n+1}} \leftarrow \underline{\mathbf{n}}, \ldots\rangle = \underline{\mathbf{n-1}},$$

    e no $\lambda\sigma$ temos

    $$(\lambda\underline{\mathbf{n}})V \to_{Beta} \underline{\mathbf{n}}[(V \cdot id)] = \underline{\mathbf{1}}[\uparrow^{n-1}][(V \cdot id)] \to_{Clos}$$
    $$\underline{\mathbf{1}}[(\uparrow^{n-1} \circ (V \cdot id))] = \underline{\mathbf{1}}[((\uparrow^{n-2} \circ \uparrow) \circ (V \cdot id))] \to_{Assoc}$$
    $$\underline{\mathbf{1}}[(\uparrow^{n-2} \circ (\uparrow \circ (V \cdot id)))] \to_{ShiftCons} \underline{\mathbf{1}}[(\uparrow^{n-2} \circ id)] \to_{IdR} \underline{\mathbf{1}}[\uparrow^{n-2}] = \underline{\mathbf{n-1}},$$

  – $U = (P\ Q)$, então

  $$M = ((\lambda(P\ Q))\ V) \to_\beta (P\ Q)\,\langle \underline{\mathbf{1}} \leftarrow V, \underline{\mathbf{2}} \leftarrow \underline{\mathbf{1}}, \ldots, \underline{\mathbf{n+1}} \leftarrow \underline{\mathbf{n}}, \ldots\rangle = (P'\ Q')$$

  e no λ_σ temos

  $$M = ((\lambda(P\ Q))\ V) \to_{Beta} (P\ Q)[(V \cdot id)]$$

  e pela proposição 6 com $m = 1$ e $p = 0$ temos que

  $$(P\ Q)[(V \cdot id)] \underset{\sigma}{\xrightarrow{+}} (P'\ Q')$$

  – $U = (\lambda P)$, então

  $$M = ((\lambda(\lambda P))\ V) \to_\beta (\lambda P)\,\langle \underline{\mathbf{1}} \leftarrow V, \underline{\mathbf{2}} \leftarrow \underline{\mathbf{1}}, \ldots, \underline{\mathbf{n+1}} \leftarrow \underline{\mathbf{n}}, \ldots\rangle = (\lambda P')$$

  e assim temos $(\lambda(\lambda P))V \to_{Beta} (\lambda P)[(V \cdot id)]$ tomando $m = 1$ e $p = 0$ aplicando a proposição 6 obtemos, na redução σ, que $(\lambda P)[(V \cdot id)] \underset{\sigma}{\xrightarrow{+}} (\lambda P')$

- se a redução é interna, então os seguintes casos podem ocorrer:

- $PQ \to_\beta P'Q$ com $P \to_\beta P'$, então pela HI $P \xrightarrow{+}_{\lambda\sigma} P'$ assim $PQ \xrightarrow{+}_{\lambda\sigma} P'Q$,
- $PQ \to_\beta PQ'$ com $Q \to_\beta Q'$, então pela HI $Q \xrightarrow{+}_{\lambda\sigma} Q'$ assim $PQ \xrightarrow{+}_{\lambda\sigma} PQ'$,
- $(\lambda P) \to_\beta (\lambda P')$ com $P \to_\beta P'$, então pela HI $P \xrightarrow{+}_{\lambda\sigma} P'$ assim $(\lambda P) \xrightarrow{+}_{\lambda\sigma} (\lambda P')$.

Assim, o $\lambda\sigma$-cálculo simula a regra β. □

Lema 4 *[1][Projeção]Se* $\exists m\ M, N \in \Lambda_\sigma^t$ *tal que* $M \to_{Beta} N$ *então* $\sigma(M) \underset{\beta}{\twoheadrightarrow} \sigma(N)$.

Prova. Veja a referência [1] □

Capítulo 4

Estudo da Expansibilidade nos CSEs

No estudo da expansibilidade veremos os resultados de Ariel Arbiser a respeito dos cálculos λx e $\lambda\upsilon$. Apresentaremos os nossos resultados obtidos do $\lambda\sigma$, onde fazemos a aplicação de alguns resultados de Ariel Arbiser para o $\lambda\sigma$-cálculo. E indicaremos possíveis caminhos para provar que a expansão a termos puro não é recursiva, no $\lambda\sigma$-cálculo.

Neste capítulo introduziremos os sistemas de numerais, que serão usados na demonstração do Teorema de Scott e a partir daí veremos sua utilização nos resultados de indecidibilidade.

Dividimos este capítulo em várias seções, onde trataremos sobre assuntos preliminares em preparação ao estudo da expansibilidade.

4.1 Sistemas de numerais

Como sistemas de numerais veremos a definição de valores booleanos, de pares e de numerais padrão, que se encontram na referência [11], que auxiliarão no Teorema do Ponto Fixo e no Teorema de Scott.

Os valores *booleanos* true e false são definidos por termos **T** e **F** respectivamente:

Definição 25 (True e False)

$$\boldsymbol{T} = \lambda xy.x \quad \boldsymbol{F} = \lambda xy.y$$

Definição 26 (Pares) *Definimos a operação de pares como um operador da forma,* [_ , _] *tal que*

$$[M, N] = \lambda z.zMN$$

Definição 27 (Numerais padrão)

$$\begin{array}{ll} \ulcorner 0 \urcorner & = \boldsymbol{I} = \lambda x.x \\ \ulcorner n+1 \urcorner & = [\boldsymbol{F}, \ulcorner n \urcorner] \end{array}$$

Assim por exemplo:

$$\ulcorner 3 \urcorner = [\mathbf{F}, [\mathbf{F}, [\mathbf{F}, \mathbf{I}]]]$$

Definiremos um predicado unário, **Zero**, que retorna **T** se o argumento é $\ulcorner 0 \urcorner$ e **F** caso contrário:

$$\mathbf{Zero} = \lambda x.x\mathbf{T}$$

Uma vez que:

$$\begin{array}{ll} \mathrm{I}\mathbf{T} & = \mathbf{T} \\ [\mathbf{F}, \ulcorner n \urcorner]\mathbf{T} & = \mathbf{F} \end{array}$$

4.2 Decidibilidade

Daremos início a esta seção com algumas noções, como o número de Gödel, e apresentando também o Teorema do Ponto Fixo representado nos numerais padrões e no número de Gödel. Depois veremos o teorema de D. Scott que permite identificar alguns conjuntos de termos não-recursivos (isto é, indecidíveis) no cálculo λ. Esta seção tem por base as referências [11] e [3].

Na prova do Teorema da Incompletude de Gödel, Gödel introduz um conceito de numeração, a numeração que usa como técnica a codificação de cada sentença de uma dada teoria a um único número natural. Traduzindo este resultado para o cálculo λ, nós temos que existe uma aplicação injetiva $\# : \Lambda \to \mathbb{N}$ tal que $\#M$ é o número Gödel de M.

Teorema 1 (Teorema do Ponto Fixo)

$$\forall F \in \Lambda, \exists X \in \Lambda. F\ulcorner \#X \urcorner = X$$

Prova. Definindo

$$\begin{array}{ll} \mathbf{Ap}\ulcorner \#M \urcorner(\ulcorner \#N \urcorner) & = \ulcorner \#(MN) \urcorner \\ \mathbf{Num}\ulcorner \#n \urcorner & = \ulcorner \#(\ulcorner \#n \urcorner) \urcorner \end{array}$$

Agora tomando $W = \lambda x.F(\mathbf{Ap}\ x(\mathbf{Num}\ x))$ e $X = W\ulcorner \#W \urcorner$; então:

$$\begin{array}{ll} X & \to F(\mathbf{Ap}\ulcorner \#W \urcorner(\mathbf{Num}\ulcorner \#W \urcorner)) \\ & = F(\mathbf{Ap}\ulcorner \#W \urcorner(\ulcorner \#\ulcorner \#W \urcorner \urcorner)) \\ & = F(\ulcorner \#(\ulcorner W\ulcorner \#W \urcorner)\urcorner) \\ & = F\ulcorner \#X \urcorner \text{ conforme requerido.} \end{array}$$

□

Definição 28 *Se A é um subconjunto de Λ então: $\#A = \{\#M \mid M \in A\}$*

Definição 29 *A é não trivial se $A \neq \emptyset$ e $A \neq \Lambda$.*

Definição 30 *A é fechado sobre o fecho de equivalencia β se*

$$\forall M, N \in \Lambda[M \in A \wedge M =_\beta N \Rightarrow N \in A]$$

Definição 31 (Conjuntos recursivamente separáveis) *A e B são recursivamente separáveis se e somente se existe um conjunto recursivo C tal que:*

$$(A \subseteq C) \wedge (B \cap C = \emptyset).$$

Teorema 2 ***Teorema de Scott** [11]*

(1) *Sejam A e B, subconjuntos de Λ, conjuntos não-vazios, diferentes e fechados sobre a $=_\beta$. Então A e B não são recursivamente separáveis.*

(2) *Seja A um subconjunto de Λ não trivial e fechado sobre a $=_\beta$. Então A não é recursivo.*

Prova. *(1)* Sejam $M_0 \in A$ e $M_1 \in B$ e C um conjunto recursivo contendo A e tal que $B \cap C = \emptyset$. A função característica de $\#C$ é recursiva e denotada por F. Visto que:

$$M \in C \Rightarrow F\ulcorner \#M \urcorner = \ulcorner 0 \urcorner$$

$$M \notin C \Rightarrow F\ulcorner \#M \urcorner = \ulcorner 1 \urcorner$$

Definimos:

$$G = \lambda x.(\mathbf{Zero}(Fx))M_1 M_0$$

então:

$$M \in C \Rightarrow G\ulcorner \#M \urcorner = \ulcorner M_1 \urcorner$$
$$M \notin C \Rightarrow G\ulcorner \#M \urcorner = \ulcorner M_0 \urcorner$$

mas, pelo Teorema do Ponto Fixo:

$$G\ulcorner \#X \urcorner = X \text{ para algum } X$$

e assim:

$$X \in C \Rightarrow X = G\ulcorner \#X \urcorner = M_1 \in B \Rightarrow X \notin C$$
$$X \notin C \Rightarrow X = G\ulcorner \#X \urcorner = M_0 \in A \Rightarrow X \in C$$

que é uma contradição.

Logo não existe C, e assim A e B não são recursivamente separáveis, de acordo com a Definição 31.

(2) Se A é um conjunto não trivial e fechado sobre a igualdade, então basta aplicar *(1)* a A e seu complemento. Suponha que A seja recursivo e pela Definição 31 obtemos

$$(A \subseteq A) \wedge (A^c \cap A = \emptyset)$$

e assim A e A^c são recursivamente separáveis. Logo, uma contradição pelo item *(1)*. Assim A não pode ser recursivo. □

Com o Teorema de Scott veremos que o cálculo λ não é recursivo e assim não decidível.

Proposição 8 *Seja $C \subset \Lambda$ um sub-conjunto próprio de λ-termos (isto é não vazio e nem todo o conjunto Λ). Se C é fechado sobre a relação $=_\beta$ (isto é, se $M \in C$, $N \in \Lambda$ e $M =_\beta N$ então $N \in C$), então C é não-recursivo.*

Prova. Basta tomar $C = A$ no item *(2)* no Teorema de Scott. □

Lema 5 *Sejam $N \in \Lambda$, um termo fixo dado, e $C = \{M \in \Lambda \mid M =_\beta N\}$, isto é, o conjunto de todos os termos β-equivalentes a N. Então C é não-recursivo.*

Prova. O conjunto C é não trivial, pois $N \in C$ e $C \neq \Lambda$ senão teríamos $\Lambda = N$, que não é verdade, e fechado sob $=_\beta$. Então usando a Proposição 8, temos que C é não-recursivo. $\square$

Corolário 2 *A relação $=_\beta$ é indecidível.*

Prova. Seja a relação $=_\beta$ decidível, então dados dois termos $M, N \in \Lambda$ conseguimos determinar $M =_\beta N$. Assim, tomando $N = \underline{\mathbf{1}}$ e o conjunto $C = \{M \in \Lambda \mid M =_\beta \underline{\mathbf{1}}\}$ que é recursivo, chegamos a uma contradição pelo Lema 5. Logo esse conjunto é não-recursivo, e $=_\beta$ é indecidível $\square$

Além disso temos:

Corolário 3 *A relação $\underset{\beta}{\twoheadrightarrow}$ é indecidível.*

Prova. Seja a relação $\underset{\beta}{\twoheadrightarrow}$ decidível, assim dados $M, N \in \Lambda$ tal que $M \underset{\beta}{\twoheadrightarrow} N$ implica que $M =_\beta N$ é decidível. Isto é um absurdo pelo Lema 5. $\square$

4.3 Expansibilidade

O objetivo nesta seção é estudar a expansibilidade, e para isso veremos como se comporta esta propriedade nos CSEs, analisando se é possível decidir a expansão dos termos. Somente os resultados obtidos para os cálculos λx e $\lambda\upsilon$, nesta seção, foram obtidos por A. Abirser [3].

Corolário 4 *No cálculo λ o problema da expansibilidade comum é indecidível, isto é, dados $M, N \in \Lambda$, o problema de decidir se existe $P \in \Lambda$ tal que $P \underset{\beta}{\twoheadrightarrow} M$ e $P \underset{\beta}{\twoheadrightarrow} N$ é indecidível.*

Prova. Seja o problema acima decidível. Então dados $M, N \in \Lambda$ é possível determinar $P \in \Lambda$ tais que $P \underset{\beta}{\twoheadrightarrow} M$ e $P \underset{\beta}{\twoheadrightarrow} N$. Então dado $M = \underline{\mathbf{1}}$ temos que $P \underset{\beta}{\twoheadrightarrow} \underline{\mathbf{1}}$ e $N \underset{\beta}{\twoheadrightarrow} \underline{\mathbf{1}}$,pela confluência e pela fato que $\underline{\mathbf{1}}$ é uma forma β-normal. Então $C = \{M \in \Lambda \mid M \underset{\beta}{\twoheadrightarrow} \underline{\mathbf{1}}\}$ é recursivo. Mas pelo Lema 5 obtemos uma contradição. $\square$

4.3.1 Expansibilidade em λx

No λx a propriedade de expansão é trivial, ou seja, todos os termos expandem para termos puros (em zero ou mais etapas) vemos a prova de [3].

Proposição 9 *Para todo $M \in \Lambda$x, existe $M' \in \Lambda$ tal que $M' \underset{\lambda x}{\twoheadrightarrow} M$.*

Prova. Por indução sobre a estrutura de M. Se $M = x$, então como $x \in \Lambda$ basta tomar a redução $x \underset{\lambda x}{\twoheadrightarrow} x$. Se $M = PQ$, então por HI existe $P' \in \Lambda$ tal que $P' \underset{\lambda x}{\twoheadrightarrow} P$, e existe $Q' \in \Lambda$ tal que $Q' \underset{\lambda x}{\twoheadrightarrow} Q$. Tome $M' = P'Q' \underset{\lambda x}{\twoheadrightarrow} M$. Se $M = \lambda x.P$, então por HI existe $P' \in \Lambda$ tal que $P' \underset{\lambda x}{\twoheadrightarrow} P$. Tome $M' = \lambda x.P' \underset{\lambda x}{\twoheadrightarrow} M$. Se $M = P\langle x := Q\rangle$, por HI existe $P' \in \Lambda$ tal que $P' \underset{\lambda x}{\twoheadrightarrow} P$, e existe $Q' \in \Lambda$ tal que $Q' \underset{\lambda x}{\twoheadrightarrow} Q$. Tome $M' = (\lambda x.P')Q' \rightarrow_{Beta} P'\langle x := Q'\rangle \underset{\lambda x}{\twoheadrightarrow} M$. $\square$

4.3.2 Indecibilidade em $\lambda\upsilon$

Agora avançaremos para o cálculo $\lambda\upsilon$ para vermos alguns resultados de indecibilidade feitos por Ariel Arbiser [3]. E apresentaremos estes resultados com mais detalhes nesta seção. E para isso usaremos dentre outras relações, a simulação, a correção e a projeção.

Definição 32 *[$\lambda\upsilon$-termos com expansão pura] Dado $M \in \Lambda^t_\upsilon$, M tem uma expansão pura se existe $N \in \Lambda$ tal que $N \underset{\lambda\upsilon}{\twoheadrightarrow} M$. Dado pelo conjunto $\Lambda^P_\upsilon = S(\Lambda) = \{M \in \Lambda^t_\upsilon | M$ tem uma expansão pura$\}$.*

Corolário 5 *A relação $\underset{\lambda\upsilon}{\twoheadrightarrow}$ é indecidível.*

Prova. Suponha que $\underset{\lambda\upsilon}{\twoheadrightarrow}$ é decidível, logo o conjunto $C = \{M \in \Lambda \mid M \to_B N \underset{\lambda\upsilon}{\twoheadrightarrow} \underline{\mathbf{1}}\}$ é recursivo. Então, pela propriedade de correção, temos que o conjunto $D = \{M \in \Lambda \mid M \to_\beta \underline{\mathbf{1}}\}$ é recursivo, que é um absurdo pelo Corolário 3. □

Para determinar a indecibilidade de $=_{\lambda\upsilon}$ precisamos do lema abaixo:

Lema 6 *Dados $M, N \in \Lambda$. Então $M =_{\lambda\upsilon} N$ se somente se $M =_\beta N$.*

Prova. Se $M =_\beta N$, então pela confluência do cálculo λ existe um termo $U \in \Lambda$ tal que $M \underset{\beta}{\twoheadrightarrow} U$ e $N \underset{\beta}{\twoheadrightarrow} U$ [8]. Então pela simulação[9], $M \underset{\lambda\upsilon}{\twoheadrightarrow} U$ e $N \underset{\lambda\upsilon}{\twoheadrightarrow} U$, assim $M =_{\lambda\upsilon} N$.Para outra implicação, se $M =_{\lambda\upsilon} N$, então pela confluência de $\lambda\upsilon$ existe um termo $U' \in \Lambda^t_\upsilon$ tal que $M \underset{\lambda\upsilon}{\twoheadrightarrow} U'$ e $N \underset{\lambda\upsilon}{\twoheadrightarrow} U'$. U' pode não ser puro, assim tomando $U = \upsilon(U')$ a forma υ-normal de U', pela terminação e confluência de υ [9], então temos $M \underset{\lambda\upsilon}{\twoheadrightarrow} U$ e $N \underset{\lambda\upsilon}{\twoheadrightarrow} U$ e então pela propriedade de correção $M \underset{\beta}{\twoheadrightarrow} U$ e $N \underset{\beta}{\twoheadrightarrow} U$ assim $M =_\beta N$. □

Corolário 6 *A relação $=_{\lambda\upsilon}$ é indecidível.*

Prova. Suponha que seja decidível, então pelo Lema 6 existe um algorítmo que prova $M =_\beta N$ para todo par $M, N \in \Lambda$, que é absurdo pelo Corolário 2. □

Corolário 7 *Dados $M, N \in \Lambda$, o problema de decidir se existe $P \in \Lambda$ tal que $P \underset{\lambda\upsilon}{\twoheadrightarrow} M$ e $P \underset{\lambda\upsilon}{\twoheadrightarrow} N$ é indecidível.*

Prova. $\lambda\upsilon$ satisfaz as propriedades de correção e de simulação da β-redução, então se este problema for decidível, ele irá contradizer o Corolário 4, pois teríamos que é possível decidir se existe $P \in \Lambda$ tal que $P \underset{\beta}{\twoheadrightarrow} M$ e $P \underset{\beta}{\twoheadrightarrow} N$. □

Corolário 8 *Dados $M, N \in \Lambda^t_\upsilon$, o problema de decidir se existe $P \in \Lambda$ tal que $P \underset{\lambda\upsilon}{\twoheadrightarrow} M$ e $P \underset{\lambda\upsilon}{\twoheadrightarrow} N$ é indecidível.*

Prova. Reduz-se este problema ao do Corolário 7. Pois, se existe um algorítmo Γ para decidir a existência de um termo $P \in \Lambda$ tal que $P \underset{\lambda\upsilon}{\twoheadrightarrow} M$ e $P \underset{\lambda\upsilon}{\twoheadrightarrow} N$ com $M, N \in \Lambda^t_\upsilon$, então poderíamos usar Γ no Corolário 7, o que seria uma contradição, visto que $\Lambda \subseteq \Lambda^t_\upsilon$. □

Lema 7 *Dados $M, N \in \Lambda$. Então existe $P' \in \Lambda^t_\upsilon$ tal que $P' \underset{\lambda\upsilon}{\twoheadrightarrow} M$ e $P' \underset{\lambda\upsilon}{\twoheadrightarrow} N$ se somente se existe $P \in \Lambda$ tal que $P \underset{\beta}{\twoheadrightarrow} M$ e $P \underset{\beta}{\twoheadrightarrow} N$.*

Prova. A implicação ($\Leftarrow$) é provada pela simulação da β-redução. Portanto para cada redução $P_1 \to_\beta P_2$ obtemos pela simulação $P_1 \to_B P_2'$, depois usando as regras de υ temos a forma normal $\upsilon(P_2') = P_2$. Para verificar a implicação ($\Rightarrow$), usa-se a Lema de Projeção [9]. Suponha que $P' \in \Lambda_\upsilon^t$, $P' \underset{\lambda\upsilon}{\twoheadrightarrow} M$ e $P' \underset{\lambda\upsilon}{\twoheadrightarrow} N$, com $M, N \in \Lambda$, para casos $P_1 \underset{\upsilon}{\twoheadrightarrow} P_2$ temos que $\upsilon(P_1) \to_\beta^0 \upsilon(P_2)$, pois $\upsilon(P_1) = \upsilon(P_2)$, e para $P_1 \to_B P_2$ temos $\upsilon(P_1) \to_\beta \upsilon(P_2)$. Então tem-se $P = \upsilon(P')$, assim pela Lema de Projeção $P \underset{\beta}{\twoheadrightarrow} \upsilon(M) = M$ e $P \underset{\beta}{\twoheadrightarrow} \upsilon(N) = N$. $\square$

Análise de Contexto

Nesta secão observa-se-á quais condições são necessárias para que a propriedade de expansibilidade seja válida para contextos de termos do $\lambda\upsilon$ feitas por Arbiser [3].

Definição 33 *Um contexto será chamado direito se não sastisfaz as definições logo abaixo.*

1. *Um contexto $C = C\{\Box\}$ é do tipo-λ se $C\{\Box\} = D\{\lambda D'\{\Box\}\}$ com D, D' contextos.*

2. *Um contexto $C = C\{\Box\}$ é do tipo-fecho se $C\{\Box\} = D\{(D'\{\Box\})[s]\}$ com D, D' contextos e $s \in \Lambda_\upsilon^s$.*

3. *Um contexto $C = C\{\Box\}$ é do tipo-aplicação se $C\{\Box\} = D\{(D'\{\Box\})M\}$ com D, D' contextos e $M \in \Lambda_\upsilon^t$.*

4. *Um contexto $C = C\{\Box\}$ é do tipo-$\Uparrow$ se $C\{\Box\} = D\{M[\Uparrow (D'\{\Box\})]\}$ com D um contexto-termo, D' um contexto qualquer e $M \in \Lambda_\upsilon^t$. Equivalentemente, se $\exists k \geq 1$ tal que $C\{\Box\} = D\{M[\Uparrow^k (D'\{\Box\}/)]\}$ com D, D' contextos-termos e $M \in \Lambda_\upsilon^t$.*

Dizendo de outra forma, o buraco não está no corpo de λ nem de $\Uparrow$, nem dentro da cabeça de um fecho, nem dentro do lado esquerdo de uma aplicação.

Observação 3 *Dado um contexto direito $C\{\Box\}$ e $a \in \Lambda_\upsilon^t$, então:*

- *$aC\{\Box\}$ é um contexto direito,*
- *$a[C\{\Box\}/]$ é um contexto direito.*

Definição 34 *Dado um termo $M \in \Lambda_\upsilon^t$, uma posição $q \in Pos(M)$ está a direita da posição $p \in Pos(M)$ se e somente se*

- *existe um sub-termo de M que tem a forma $(P\ Q)$, e p é uma posição de M dentro de P e q é uma posição de M dentro de Q*

- *ou existe um sub-termo de M que tem a forma $P[s]$, e p é uma posição de M dentro de P e q é uma posição de s.*

Lema 8 *Dado $M \in \Lambda_\upsilon^t$, com p uma posição de um sub-termo P de M e q uma posição de um sub-termo Q de M. Então somente uma das seguintes afirmações é verdadeira:*

1. $p = q$,
2. q *está a direita de* p *(equivalentemente,* $p = \alpha.1.\beta$ *e* $q = \alpha.2.\gamma$ *para alguns valores de* α, β, γ*),*
3. p *está a direita de* q *(equivalentemente,* $p = \alpha.2.\beta$ *e* $q = \alpha.1.\gamma$ *para alguns valores de* α, β, γ*),*
4. p *é um prefixo próprio de* q *(equivalentemente,* $q = p.\alpha$ *para alguns valores não-vazio de* α*),*
5. q *é um prefixo próprio de* p *(equivalentemente,* $p = q.\alpha$ *para alguns valores não-vazio de* α*).*

No caso do item 2 ou 3 ser verdadeiro, diz-se que p e q são *disjuntos*.

Para $M \in \Lambda_v^t$, chamam-se *posições de termos* aquelas posições de $Pos(M)$ de subtermos válidos de um termo (mas não termo de substituição). Por exemplo, para um termo igual a $M[\Uparrow (N/)]$, 1 e 2.1.1 são posições de termos (os subtermos são M e N respectivamente), mas 2 e 2.1 não são. Note que um contexto irá ser um contexto-termo se e somente se o buraco está em uma posição de termo.

Lema 9 *Se no contexto* C *o buraco tem alguma posição de termo à direita, então* C *não é direito.*

Prova. Por indução sobre o contexto C.

- Se $C\{\Box\} = \Box$, o lema vale apesar de não haver nenhum conteúdo.
- Se $C\{\Box\} = C'\{\Box\}N$, pela Definição 33, C não é direito.
- Se $C\{\Box\} = MC'\{\Box\}$, pela HI C' não é direito (porque em C' o buraco também tem alguma posição de termo à direita), assim C não é direito.
- Se $C\{\Box\} = \lambda C'\{\Box\}$, pela Definição 33, C não é direito.
- Se $C\{\Box\} = C'\{\Box\}[s]$, pela Definição 33, C não é direito.
- Se $C\{\Box\} = M[C'\{\Box\}]$, pela HI C' não é direito (porque em C' o buraco também tem alguma posição de termo à direita), assim C não é direito.

$\Box$

Lema 10 (Invariantes $\Uparrow -/$ em um contexto direito) *Se* C *é um contexto direito, com* $B, M, P \in \Lambda_v^t$*, e* $k \geq 1$*, tal que* $B \underset{v}{\rightarrow} C\{M[\Uparrow^k (P/)]\}$*.*

Então existe um contexto direito C'*, termos* $M', P' \in \Lambda_v^t$ *e* $k' \geq 1$ *tal que* $B = C'\{M'[\Uparrow^{k'} (P'/)]\}$ *e que* $P' \underset{\lambda v}{\twoheadrightarrow} P$*.*

Prova. Por indução sobre o contexto C. Tome $e = M[\Uparrow^k (P/)]$. Não especificaremos M', P', k' quando a escolha for nítida tendo C' escolhido.

- Se $C\{\Box\} = \Box$, então:

- Se a redução está na raiz de B, então analisaremos cada $\lambda\upsilon$-regra:
  * (Beta), é impossível visto que $a[b/]$ não é igual a e porque $k \geq 1$.
  * (App), é impossível visto que $(a[s]\ b[s])$ não é igual a e.
  * (Lam), é impossível visto que $\lambda(a[\Uparrow(s)])$ não é igual a e.
  * (Fvar), então tem-se $C'\{\Box\} = \underline{\mathbf{1}}[\Box/]$ que é um contexto direito pela Observação 3, assim segue os resultados conseqüentes.
  * (Rvar), é impossível visto que $\underline{\mathbf{n}}$ não é igual a e.
  * (FvarLift), é impossível visto que $\underline{\mathbf{1}}$ não é igual a e.
  * (RvarLift), é impossível visto que $\underline{\mathbf{n}}[s][\uparrow]$ não é igual a e.
  * (VarShift), é impossível visto que $\underline{\mathbf{n+1}}$ não é igual a e.
- Se a redução é interna em B:
  * $B = (a'\ b) \to (a\ b) = e$ com $a' \to a$, não acontece visto que $(a\ b)$ não é igual a um fecho.
  * $B = (a\ b') \to (a\ b) = e$ com $b' \to b$, não acontece também pela mesma razão.
  * $B = \lambda a' \to \lambda a = e$ com $a' \to a$, não acontece visto que λa não é igual a um fecho.
  * $B = a'[s] \to a[s] = e$ com $a' \to a$, então $a = M$ e $s = \Uparrow^k(P/)$, assim temos $C'\{\Box\} = \Box$ que é direito.
  * $B = a[s'] \to a[s] = e$ com $s' \to s$, então $a = M$ e $s' = \Uparrow^k(P'/)$, com $P' \to P$, assim temos novamente $C'\{\Box\} = \Box$ que é direito.

- Se $C\{\Box\} \neq \Box$, então:
  - Se a redução está na raiz de B, então analisaremos cada $\lambda\upsilon$-regra:
    * para a regra $((\lambda a)\ b) \to_{Beta} a[b/]$, o buraco precisaria estar em b visto que C é direito, assim $b = D\{e\}$, para D um contexto direito; temos $C'\{\Box\} = (\lambda a)D\{\Box\}$ que é direito pela Observação 3, assim segue os resultados conseqüentes.
    * para a regra $(a\ b)[s] \to_{App} (a[s]\ b[s])$, o buraco precisaria estar na extrema direita de s (de outra forma o contexto poderia não ser direito), e então a única possibilidade é que existe N tal que $s = N/$ com o buraco em N (de outra forma, se $s = \Uparrow^d(N/)$ com $d \geq 1$, $C\{\Box\}$ poderia não ser direito), assim temos $C'\{\Box\} = (a\ b)[D\{\Box\}/]$ onde D é um contexto direito tal que $N = D\{e\}$, então C' é direito pela Observação 3, assim segue os resultados conseqüentes.
    * para a regra $(\lambda a)[s] \to_{Lam} (\lambda a[\Uparrow(s)])$, qualquer posição que colocar o buraco, C não será direito. Assim este caso é descartado.
    * para a regra $\underline{\mathbf{1}}[a/] \to_{Fvar} a$, temos $C'\{\Box\} = \underline{\mathbf{1}}[C\{\Box\}/]$ que é direito pelo Observação 3.
    * para a regra $\underline{\mathbf{n+1}}[a/] \to_{Rvar} \underline{\mathbf{n}}$, é descartado, visto que $\underline{\mathbf{n}}$ não é igual a $C\{e\}$.
    * para a regra $\underline{\mathbf{1}}[\Uparrow(s)] \to_{FvarLift} \underline{\mathbf{1}}$, é descartado, visto que $\underline{\mathbf{1}}$ não é igual a $C\{e\}$.

* para a regra $\underline{\mathbf{n+1}}[\Uparrow (s)] \to_{RvarLift} \underline{\mathbf{n}}[s][\uparrow]$, o buraco não pode ser situado em $\underline{\mathbf{n}}$ visto que $\underline{\mathbf{n}}$ não é igual a $C\{e\}$, nem em s porque C poderia não ser direito, e nem na posição de $\underline{\mathbf{n}}[s]$ visto que novamente C poderia não ser direito, assim este caso é descartado.
* para a regra $\underline{\mathbf{n}}[\uparrow] \to_{VarShift} \underline{\mathbf{n+1}}$, é descartado, visto que $\underline{\mathbf{n+1}}$ não é igual a $C\{e\}$.

- Se a redução é interna em B:
  * para $B = (a'\ b) \underset{v}{\to} (a\ b) = C\{e\}$ onde $a' \underset{v}{\to} a$, então o buraco deve estar situado em b (de outra forma C poderia não ser direito). Então existe um contexto direito D tal que $b = D\{e\}$. Temos $C'\{\Box\} = (a'\ D\{\Box\})$ que é direito pela Observação 3, assim segue os resultados conseqüentes.
  * para $B = (a\ b') \underset{v}{\to} (a\ b) = C\{e\}$ onde $b' \underset{v}{\to} b$, então como antes o buraco deve estar situado em b (de outra forma C poderia não ser direito). Então existe um contexto direito D tal que $b = D\{e\}$. Pela HI, existe um contexto direito D' tal que $b' = D'\{M'[\Uparrow^{k'} (P'/)]\}$ para $M', P' \in \Lambda_v^t$ e $k' \geq 1$ com $P' \underset{\lambda v}{\twoheadrightarrow} P$. Temos $C'\{\Box\} = (a\ D'\{\Box\})$ que é direito pela Observação 3, assim segue os resultados conseqüentes.
  * para $B = (\lambda a') \underset{v}{\to} (\lambda a) = C\{e\}$ onde $a' \underset{v}{\to} a$, não pode acontecer porque C poderia não ser direito.
  * para $B = a'[s] \underset{v}{\to} a[s] = C\{e\}$ onde $a' \underset{v}{\to} a$, então o buraco deve estar situado em s (de outra forma C poderia não ser direito). Então s não pode ser $\Uparrow (s')$ para qualquer substituição s' visto que C é direito, assim $s = R/$ para algum termo R dessa forma $R = D\{e\}$ para algum contexto direito D. Tomando $C'\{\Box\} = a'[D\{\Box/\}]$ que é direito pela Observação 3 assim segue os resultados conseqüentes.
  * para $B = a[s'] \underset{v}{\to} a[s] = C\{e\}$ onde $s' \underset{v}{\to} s$, então como antes o buraco deve estar situado em s (de outra forma C poderia não ser direito), e s não pode ser $\Uparrow (s'')$ para qualquer substituição s'', visto que C não seria direito. Assim $s = R/$ para algum termo R, e dessa forma $s' \underset{v}{\to} s$, existe R' tal que $s' = R'/$ com $R' \underset{v}{\to} R$, assim pela HI $R' = D'\{M'[\Uparrow^{k'} (P'/)]\}$ com D' direito e $P' \underset{\lambda v}{\twoheadrightarrow} P$, temos $C'\{\Box\} = a[D'\{\Box\}/]$ que é direito pela Observação 3 assim segue os resultados conseqüentes.

$\Box$

Lema 11 ***[Invariantes $\Uparrow -/$ em uma aplicação de contexto direito]*** *Dado C um contexto qualquer, e C_1, C_2 contextos direitos, $B, M, N, P, Q \in \Lambda_v^t$ e $k, r \geq 1$, tal que $B \underset{v}{\to} C\{C_1\{M[\Uparrow^k (P/)]\}C_2\{N[\Uparrow^r (Q/)]\}\}$, onde a redução não é um etapa-(App) na posição de $\Box \in C\{\Box\}$. Em outras palavras, suponha-se que não é o caso de $B = C\{(R\ T)[s]\}$ para alguns termos R, T e uma substituição s tal que $(R\ T)[s] \to_{App} C_1\{M[\Uparrow^k (P/)]\}C_2\{N[\Uparrow^r (Q/)]\}\}$ sendo esta uma redução na raiz.*

Então existe um contexto C', contextos direitos C_1', C_2', termos $M', N', P', Q' \in \Lambda_v^t$ e $k', r' \geq 1$ tal que $B = C'\{C_1'\{M'[\Uparrow^{k'} (P'/)]\}C_2'\{N'[\Uparrow^{r'} (Q'/)]\}\}$ e tal que $P' \underset{\lambda v}{\twoheadrightarrow} P$ e $Q' \underset{\lambda v}{\twoheadrightarrow} Q$.

Prova. Pela indução sobre o contexto C. Chame $e = C_1\{M[\Uparrow^k (P/)]\}C_2\{N[\Uparrow^r (Q/)]\}$. Não daremos $M', N', P', Q', k', r', C_1', C_2'$ quando a escolha é nítida uma vez que C' tenha sido escolhido.

- Se $C\{\Box\} = \Box$, então:
  - Se a redução está na raiz de B, então analisa-se cada $\lambda\upsilon$-regra e obtém somente uma possibilidade:
    * (Fvar), então tem-se $C'\{\Box\} = \underline{\mathbf{1}}[C\{\Box\}/]$ assim segue os resultados conseqüentes.
  - Se a redução é interna em B:
    * caso $B = (a'\ b) \underset{\upsilon}{\rightarrow} (a\ b)$, com $a' \underset{\upsilon}{\rightarrow} a$, então $(a\ b) = e$ assim $a = C_1\{M[\Uparrow^k (P/)]\}$ e $b = C_2\{N[\Uparrow^r (Q/)]\}$. Pelo Lema 10, existem C_1' um contexto direito, $M', k' \geq 1$ e P' tal que $a' = C_1'\{M'[\Uparrow^{k'} (P'/)]\}$ e $P' \underset{\lambda\upsilon}{\twoheadrightarrow} P$, assim temos $C'\{\Box\} = \Box$.
    * caso $B = (a\ b') \underset{\upsilon}{\rightarrow} (a\ b)$, com $b' \underset{\upsilon}{\rightarrow} b$, então de forma análoga $(a\ b) = e$ assim $a = C_1\{M[\Uparrow^k (P/)]\}$ e $b = C_2\{N[\Uparrow^r (Q/)]\}$. Pelo Lema 10, existem C_2' um contexto direito, $N', r' \geq 1$ e Q' tal que $b' = C_2'\{N'[\Uparrow^{r'} (Q'/)]\}$ e $Q' \underset{\lambda\upsilon}{\twoheadrightarrow} Q$, assim temos $C'\{\Box\} = \Box$.
    * caso $B = (\lambda a') \underset{\upsilon}{\rightarrow} (\lambda a)$ com $a' \underset{\upsilon}{\rightarrow} a$, é descartado, visto que λa não é igual a uma aplicação.
    * caso $B = a'[s] \underset{\upsilon}{\rightarrow} a[s]$ com $a' \underset{\upsilon}{\rightarrow} a$, é descartado, visto que $a[s]$ não é igual a uma aplicação.
    * caso $B = a[s'] \underset{\upsilon}{\rightarrow} a[s]$ com $s' \underset{\upsilon}{\rightarrow} s$, é descartado, visto que $a[s]$ não é igual a uma aplicação.
- Se $C\{\Box\} \neq \Box$, então:
  - Se a redução está na raiz de B, então analisaremos cada $\lambda\upsilon$-regra:
    * para a regra $((\lambda a)\ b) \rightarrow_{Beta} a[b/]$, então onde quer que o buraco esteja situado segue os resultados do Lema. Por exemplo, se o buraco está em b, então $b = D\{e\}$, para D um contexto; temos $C'\{\Box\} = ((\lambda a)\ D\{\Box\})$ assim segue os resultados conseqüentes.
    * para a regra $(a\ b)[s] \rightarrow_{App} (a[s]\ b[s])$, sempre o buraco é situado, assim segue os resultados conseqüentes para um cotexto C' apropriado, ou seja:
      1. se o buraco está no primeiro s, temos $C'\{\Box\} = (\lambda a)b[D\{\Box\}]$ onde $s = D\{e\}$;
      2. se o buraco está no segundo s, temos C' como acima;
      3. se o buraco está em a, temos $C'\{\Box\} = (D\{\Box\}\ b)[s]$ onde $a = D\{e\}$;
      4. se o buraco está em b, temos $C'\{\Box\} = (a\ D\{\Box\})[s]$ onde $b = D\{e\}$;
      5. o buraco não pode estar na posição de $a[s]$ visto que $a[s]$ não é igual a uma aplicação;
      6. o buraco não pode estar na posição de $b[s]$ visto que $b[s]$ não é igual a uma aplicação também;

7. o buraco não pode estar na posição de $(a[s]\ b[s])$ se não $C\{\square\} = \square$ no qual corresponde a um caso analizado anteriormente.

* para a regra $(\lambda a)[s] \to_{Lam} (\lambda a[\Uparrow(s)])$, se $\square \in a$, tome $C'\{\square\} = (\lambda D\{\square\})[s]$, para $a = D\{e\}$ e se $\square \in s$, tome $s = N/$ com $\square \in N$ e $N = D\{e\}$, logo tome $C' = (\lambda a)[D\{e\}/]$.
* para a regra $\underline{\mathbf{1}}[a/] \to_{Fvar} a$, temos $C'\{\square\} = \underline{\mathbf{1}}[C\{\square\}/]$.
* para a regra $\underline{\mathbf{n+1}}[a/] \to_{Rvar} \underline{\mathbf{n}}$, descartamos, visto que $\underline{\mathbf{n}}$ não é igual a $C\{e\}$.
* para a regra $\underline{\mathbf{1}}[\Uparrow(s)] \to_{FvarLift} \underline{\mathbf{1}}$, descartamos, visto que $\underline{\mathbf{1}}$ não é igual a $C\{e\}$.
* para a regra $\underline{\mathbf{n+1}}[\Uparrow(s)] \to_{RvarLift} \underline{\mathbf{n}}[s][\uparrow]$, então $\square \in s$, visto que $\underline{\mathbf{n}}$ não pode conter $C\{e\}$ assim tome $C' = \{\square\} = (\underline{\mathbf{n+1}})[\Uparrow(D\{\square\})]$ para $s = D\{e\}$.
* para a regra $\underline{\mathbf{n}}[\uparrow] \to_{VarShift} \underline{\mathbf{n+1}}$, descartamos, visto que $\underline{\mathbf{n+1}}$ não é igual a $C\{e\}$.

– Se a redução é interna em B:

* para $B = (a'\ b) \underset{v}{\to} (a\ b) = C\{e\}$ onde $a' \underset{v}{\to} a$, então temos:
  1. se o buraco é situado em b então existe um contexto D tal que $b = D\{e\}$. Temos $C'\{\square\} = (a'\ D\{\square\})$ e assim segue os resultados conseqüentes.
  2. se o buraco é situado em a então existe um contexto D tal que $a = D\{e\}$. Pela HI, existe um contexto D', contextos direitos C'_1 e C'_2, termos M', N', P', Q' e $k', r' \geq 1$ tal que $a' = D'\{C'_1\{M'[\Uparrow^{k'}(P'/)]\}C'_2\{N'[\Uparrow^{r'}(Q'/)]\}\}$, $P' \underset{\lambda v}{\twoheadrightarrow} P$ e $Q' \underset{\lambda v}{\twoheadrightarrow} Q$.
     Temos $C'\{\square\} = D'\{\square\}b$ e assim segue os resultados conseqüentes.
* para $B = (a\ b') \underset{v}{\to} (a\ b) = C\{e\}$ onde $b' \underset{v}{\to} b$, é análogo ao caso anterior.
* para $B = \lambda a' \underset{v}{\to} \lambda a = C\{e\}$ onde $a' \underset{v}{\to} a$, então $\square \in a$ e pela HI, existe um contexto D tal que $a' = D'\{M'[\Uparrow^{k'}(P'/)]N'[\Uparrow^{r'}(Q'/)]\}$ para $M', N', P', Q' \in \Lambda^t_v$ e $k', r' \geq 1$ com $P' \underset{\lambda v}{\twoheadrightarrow} P$ e $Q' \underset{\lambda v}{\twoheadrightarrow} Q$. Temos $C'\{\square\} = \lambda D'\{\square\}$ e assim segue os resultados.
* para $B = a'[s] \underset{v}{\to} a[s] = C\{e\}$ onde $a' \underset{v}{\to} a$, então temos:
  1. se o buraco está situado em s então s não pode ser $\Uparrow^n(\uparrow)$ para algum $n \geq 0$, assim $s = \Uparrow^n(R/)$ para algum termo R e $n \geq 0$, assim $R = D\{e\}$ para algum contexto D. Temos $C'\{\square\} = a'[\Uparrow^n(D\{\square\}/)]$ e assim segue os resultados conseqüentes.
  2. se o buraco é situado em a então $a = D\{e\}$ para algum contexto D. Pela HI existe um contexto D', contextos direitos C'_1 e C'_2, termos M', N', P', Q' e $k', r' \geq 1$ tal que $a' = D'\{C'_1\{M'[\Uparrow^{k'}(P'/)]\}C'_2\{N'[\Uparrow^{r'}(Q'/)]\}\}$, $P' \underset{\lambda v}{\twoheadrightarrow} P$ e $Q' \underset{\lambda v}{\twoheadrightarrow} Q$.
     Temos $C'\{\square\} = D'\{\square\}[s]$ e assim segue os resultados conseqüentes.
* para $B = a[s'] \underset{v}{\to} a[s] = C\{e\}$ onde $s' \underset{v}{\to} s$, então s não pode ser $\Uparrow^n(\uparrow)$ para algum $n \geq 0$, assim $s = \Uparrow^r(R/)$ para algum termo R e $n \geq 0$, e visto que $s' \underset{v}{\to} s$ então existe R' tal que $s' = \Uparrow^n(R'/)$ com $R' \underset{v}{\to} R$, assim:

1. se o buraco é situado em s então, visto que está em uma posição de termo, precisaria estar situado em R, então pela HI $R' = D'\{C_1'\{M'[\Uparrow^{k'} (P'/)]\}C_2'\{N'[\Uparrow^{r'} (Q'/)]\}\}$ com C_1' e C_2' contextos direitos, $P' \underset{\lambda\upsilon}{\twoheadrightarrow} P$ e $Q' \underset{\lambda\upsilon}{\twoheadrightarrow} Q$, assim temos $C'\{\Box\} = a[\Uparrow^n (D'\{\Box\}/)]$ e assim segue os resultados conseqüentes.
2. se o buraco é situado em a então existe um contexto D tal que $a = D\{e\}$. Temos $C'\{\Box\} = D'\{\Box\}[s']$ e assim segue os resultados conseqüentes.

$\Box$

Veremos agora uma propriedade importante na qual relaciona o problema de *expansibilidade comum*, conforme o Corolário 4, com o problema de expansibilidade para um termo puro:

Proposição 10 *Dados $P, Q \in \Lambda$. Então o termo $(\lambda(\underline{\mathbf{1}}[\Uparrow (P/)]\ \underline{\mathbf{1}}[\Uparrow (Q/)]))$ é expansível a um termo puro se e somente se $\exists R \in \Lambda\ R \underset{\beta}{\twoheadrightarrow} P$ e $R \underset{\beta}{\twoheadrightarrow} Q$.*

Prova.

- $(\Leftarrow)$ Suponha que existe um termo R. Então pela simulação de $\lambda\upsilon$ temos que $R \underset{\lambda\upsilon}{\twoheadrightarrow} P$ e $R \underset{\lambda\upsilon}{\twoheadrightarrow} Q$, logo

$$\begin{tikzcd}
(\lambda(\underline{\mathbf{1}}[\Uparrow (P/)]\ \underline{\mathbf{1}}[\Uparrow (Q/)])) \\
(\lambda(\underline{\mathbf{1}}[\Uparrow (R/)]\ \underline{\mathbf{1}}[\Uparrow (R/)])) \arrow[u, two heads, "\lambda\upsilon"'] \\
(\lambda(\underline{\mathbf{1}}\ \underline{\mathbf{1}})[\Uparrow (R/)]) \arrow[u, two heads, "App"'] \\
(\lambda(\underline{\mathbf{1}}\ \underline{\mathbf{1}}))[R/] \arrow[u, two heads, "Lam"'] \\
((\lambda\lambda(\underline{\mathbf{1}}\ \underline{\mathbf{1}}))\ R) \arrow[u, two heads, "Beta"']
\end{tikzcd}$$

  que é um termo puro, $((\lambda\lambda(\underline{\mathbf{1}}\ \underline{\mathbf{1}}))\ R)$.

- $(\Rightarrow)$ Dado $t = \lambda(\underline{\mathbf{1}}[\Uparrow (P/)]\underline{\mathbf{1}}[\Uparrow (Q/)])$. Este termo satisaz as condições do Lema 11, isto é, $t = C\{C_1\{M[\Uparrow^k (P)]\}C_2\{N[\Uparrow^r (Q)]\}\}$ onde $C_1\{\Box\} = C_2\{\Box\} = \Box$ e $C\{\Box\} = \lambda\Box$ um contexto qualquer, e $k = r = 1$, assim C_1 e C_2 são nitidamente contextos direitos.

  Pela hipótese existe um termo puro B tal que $B \underset{\lambda\upsilon}{\twoheadrightarrow} t$, assim existe $t_1 \in \Lambda, t_2, \ldots, t_m \in \Lambda_\upsilon^t$ tal que $B = t_1 \underset{\upsilon}{\rightarrow} t_2 \underset{\upsilon}{\rightarrow} \ldots \underset{\upsilon}{\rightarrow} t_m = t$.

  Mas visto que B é puro, a quantidade de $\Uparrow$ em B representado por $n_\Uparrow$ é igual a zero ($n_\Uparrow(B) = 0$), assim repetindo o Lema 11 existiria $1 \leq i < m$ tal que $t_i \rightarrow_{App} t_{i+1}$

onde esta redução está na posição de $\Box$ em $C^{(i)}\{\Box\}$, o contexto correspondente ao termo t_i (de outra forma a conclusão do Lema poderia implicar que $n_{\Uparrow}(B) > 0$). Tomando $i =$ o máximo que possa assumir, que é o primeiro i satisfazendo esta condição para o termo extremamente à direita na derivação (isto é, t).

Isto significa que (dentro desta derivação) existem termos U, V, contextos direitos D, E e termos R', R'' tal que

$$t_{i+1} = C^{(i+1)}\{D\{U[\Uparrow^k (R'/)]\}E\{V[\Uparrow^r (R''/)]\}\}$$

onde $R' \underset{\lambda v}{\twoheadrightarrow} P$ e $R'' \underset{\lambda v}{\twoheadrightarrow} Q$, e $C^{(i+1)} = C^{(i)}$ é o mesmo contexto do termo $t_i = C^{(i)}\{(ab)[s]\}$.

Correspondente a regra-(App) padrão, existem termos a, b e uma substituição s tal que $a[s] = D\{U[\Uparrow^k (R'/)]\}$ e $b[s] = E\{V[\Uparrow^r (R''/)]\}$.

Iremos mostrar que $R' = R''$, razão para analizar D e E e, em alguns casos, a posição dos buracos com respeito aos termos $a[s]$ e $b[s]$:

- $D = \Box$ e $E = \Box$: então $\Uparrow^k (R'/) = s = \Uparrow^r (R''/)$ assim $U = V, k = r$ e $R' = R''$.
- $D \neq \Box$ e $E \neq \Box$, e o buraco de D é situado dentro do fecho de $a[s]$ e o buraco de E é situado dentro do fecho de $b[s]$, em outras palavras ambos os buracos são situados dentro da substituição s. Lembramos que $a[s] = D\{U[\Uparrow^k (R'/)]\}$ e $b[s] = E\{V[\Uparrow^r (R''/)]\}$. Mostrar-se-à que a posição destes dois buracos é necessariamente a mesma. De acordo com Lema 8 os seguintes casos podem ocorrer:
  * ambos os buracos estão exatamente na mesma posição em s, então não há nada a fazer;
  * ou ambos os buracos estão em posições diferentes em s, então a posição de um dos buracos tem uma posição de termo à direita (a posição de outro buraco), portanto pelo Lema 9 o correspondente contexto não pode ser direito, que é absurdo;
  * ou a posição dos buracos de E é um prefixo próprio da posição do buraco de D, isto é, o buraco de D está localizado dentro da posição do buraco de E. Isto significa que o buraco de D está dentro do sub-termo $V[\Uparrow^r (R''/)]$, então existem três casos para verificar:
    1. a posição do buraco de D é a posição de $V[\Uparrow^r (R''/)]$, mas este caso não poderá acontecer, visto que, era suposto ser prefixo próprio (em outras palavras, este caso corresponde a um analisado antes em que ambas as posições dos buracos coincidem);
    2. a posição do buraco de D está dentro de V, mas isto é um fecho e então D não poderia ser direito, assim isto não acontece;
    3. a posição do buraco de D está dentro de $\Uparrow^r (R''/)$. Visto que D é um contexto-termo, a posição do buraco necessariamente está em R''. Esta posição está abaixo de $\Uparrow$ (visto que $r \geq 1$), que é absurdo visto que D é direito.
  * ou a posição do buraco de D é um prefixo próprio da posição do buraco de E, isto é, o buraco de E está localizado dentro da posição dos buracos

de D. Este caso é análogo ao anterior unicamente trocando as funções de D e E.
Então a única possibilidade é que os dois buracos têm a mesma posição em s. Isto exigi $U = V, k = r$ e $R' = R''$.

Agora temos que $R' = R'', R' \underset{\lambda\upsilon}{\twoheadrightarrow} P$ e $R' \underset{\lambda\upsilon}{\twoheadrightarrow} Q$. Então, visto que P e Q são puros, pelo Lema 7 existe um termo puro R tal que $R \underset{\beta}{\twoheadrightarrow} P$ e $R \underset{\beta}{\twoheadrightarrow} Q$.

$\square$

Corolário 9 *O conjunto Λ_υ^p é recursivamente enumerável mas não-recursivo. Assim, não existe algoritmo com a capacidade de decidir, dado um termo $M \in \Lambda_\upsilon^t$, se ele é expansível para um termo puro.*

Prova. Ele é nitidamente recursivamente enumerável visto que um algoritmo não-terminante pode, pela sintaxe, enumerar todos termos puros e delas reduzir sistematicamente pela aplicação todas etapas de reduções desses termos.

Se fosse recursivo, pela Proposição 10 poderíamos testar, dados termos $M, N \in \Lambda$, se existe $P \in \Lambda$ tal que $P \underset{\beta}{\twoheadrightarrow} M$ e $P \underset{\beta}{\twoheadrightarrow} N$, que é indecidível pelo Corolário 4. $\square$

Proposição 11 *Seja $C \subseteq \Lambda_\upsilon^t$ não trivial e fechado sobre $=_{\lambda\upsilon}$. Então C é não-recursivo.*

Prova. Seja $D = C \cap \Lambda$, assim tomando $N \in D$ e $P \in \Lambda$ tal que $N =_\beta P$ temos, pelo Lema 6, que $N =_{\lambda\upsilon} P$. Como C é fechado sobre $=_{\lambda\upsilon}$ então $P \in C$ e assim $P \in D$, pois $P \in \Lambda$ e $P \in \Lambda_\upsilon^t$. Portanto D é fechado sobre $=_\beta$. Como $D \subset \Lambda$ temos pela Proposição 8 que D é não-recursivo. Logo se C for recursivo, o conjunto D também deveria ser recursivo, já que $D \subset C$, que é uma contradição, pois D é não-recursivo. $\square$

4.3.3 Resultados de indecibilidade em $\lambda\sigma$

Mostraremos alguns resultados de indecibilidade no $\lambda\sigma$-cálculo, sendo lemas e proposições que foram aplicados no cálculo $\lambda\upsilon$ e agora estamos modicando-os para aplicarmos neste novo cálculo. Para isso usaremos, dentre outras relações, a simulação, e a projeção. Observe que o Lema 12, o Corolário 11 e a Proposição 12 utilizam-se do sistema definito na redução β do cálculo λ, usado na Proposição 7.

Corolário 10 *A relação* $\underset{\lambda\sigma}{\twoheadrightarrow}$ *é indecidível.*

Prova. Suponha que $\underset{\lambda\sigma}{\twoheadrightarrow}$ é decidível, logo o conjunto $C = \{M \in \Lambda \mid M \rightarrow_{Beta} N \underset{\sigma}{\overset{+}{\rightarrow}} \underline{\mathbf{1}}\}$ é recursivo. Então, pelo Lema 4, temos que o conjunto $D = \{M \in \Lambda \mid M \rightarrow_{\beta} \underline{\mathbf{1}}\}$ é recursivo, que é um absurdo pelo Corolário 3. $\square$

Lema 12 *Dados* $M, N \in \Lambda$. *Então* $M =_{\lambda\sigma} N$ *se somente se* $M =_{\beta} N$.

Prova. Se $M =_{\beta} N$, então pela confluência do cálculo λ existe um termo $U \in \Lambda$ tal que $M \underset{\beta}{\twoheadrightarrow} U$ e $N \underset{\beta}{\twoheadrightarrow} U$ [8]. Então pela Proposição 7, $M \underset{\lambda\sigma}{\twoheadrightarrow} U$ e $N \underset{\lambda\sigma}{\twoheadrightarrow} U$, assim $M =_{\lambda\sigma} N$. Para outra implicação, se $M =_{\lambda\sigma} N$, então pela confluência de $\lambda\sigma$ existe um termo $U' \in \Lambda^t_\sigma$ tal que $M \underset{\lambda\sigma}{\twoheadrightarrow} U'$ e $N \underset{\lambda\sigma}{\twoheadrightarrow} U'$. U' pode não ser puro, assim tomando $U = \sigma(U')$ a forma υ-normal de U', pela terminação e confluência de σ [1], então temos $M \underset{\lambda\sigma}{\twoheadrightarrow} U$ e $N \underset{\lambda\sigma}{\twoheadrightarrow} U$ e então pelo Lema 4 $M \underset{\beta}{\twoheadrightarrow} U$ e $N \underset{\beta}{\twoheadrightarrow} U$ assim $M =_{\beta} N$. $\square$

Corolário 11 *A relação* $=_{\lambda\sigma}$ *é indecidível.*

Prova. Suponha que seja decidível, então pelo Lema 12 é possível determinar que $M =_{\beta} N$ para todo par $M, N \in \Lambda$, que é absurdo pelo Corolário 2. $\square$

Corolário 12 *Dados* $M, N \in \Lambda$, *o problema de decidir se existe* $P \in \Lambda$ *tal que* $P \underset{\lambda\sigma}{\twoheadrightarrow} M$ *e* $P \underset{\lambda\sigma}{\twoheadrightarrow} N$ *é indecidível.*

Prova. $\lambda\sigma$ satisfaz o Lema 4 e de simulação da β-redução (Proposição 7), então se este problema for decidível, ele irá contradizer o Corolário 4, pois teríamos que é possível decidir se existe $P \in \Lambda$ tal que $P \underset{\beta}{\twoheadrightarrow} M$ e $P \underset{\beta}{\twoheadrightarrow} N$. $\square$

Corolário 13 *Dados* $M, N \in \Lambda^t_\sigma$, *o problema de decidir se existe* $P \in \Lambda$ *tal que* $P \underset{\lambda\sigma}{\twoheadrightarrow} M$ *e* $P \underset{\lambda\sigma}{\twoheadrightarrow} N$ *é indecidível.*

Prova. Reduz-se este problema ao do Corolário 12. Pois, se existe um algoritmo que decida a existência de um termo $P \in \Lambda$ tal que $P \underset{\lambda\sigma}{\twoheadrightarrow} M$ e $P \underset{\lambda\sigma}{\twoheadrightarrow} N$ com $M, N \in \Lambda^t_\sigma$, então pode-se usar este algoritmo no Corolário 12, o que seria uma contradição, visto que $\Lambda \subseteq \Lambda^t_\sigma$. $\square$

Lema 13 *Dados* $M, N \in \Lambda$. *Então existe* $P' \in \Lambda^t_\sigma$ *tal que* $P' \underset{\lambda\sigma}{\twoheadrightarrow} M$ *e* $P' \underset{\lambda\sigma}{\twoheadrightarrow} N$ *se somente se existe* $P \in \Lambda$ *tal que* $P \underset{\beta}{\twoheadrightarrow} M$ *e* $P \underset{\beta}{\twoheadrightarrow} N$.

Prova. A implicação ($\Leftarrow$) é provada pela simulação da β-redução. Portanto para cada redução $P_1 \underset{\beta}{\rightarrow} P_2$ obtemos pela simulação $P_1 \xrightarrow{+}_{\lambda\sigma} P_2'$, depois usando as regras de σ temos a forma normal $\sigma(P_2') = P_2$.

Para verificar a implicação ($\Rightarrow$), usamos o Lema 4 (o Lema de Projeção). Suponha que $P' \in \Lambda^t_\sigma$, $P' \underset{\lambda\sigma}{\twoheadrightarrow} M$ e $P' \underset{\lambda\sigma}{\twoheadrightarrow} N$, com $M, N \in \Lambda$, para casos $P_1 \underset{\sigma}{\xrightarrow{+}} P_2$ temos que $\sigma(P_1) \rightarrow^0_\beta \sigma(P_2)$, pois $\sigma(P_1) = \sigma(P_2)$, já que σ é confluente e terminante, e para $P_1 \rightarrow_{Beta} P_2$ temos $\sigma(P_1) \rightarrow_\beta \sigma(P_2)$. Então tem-se $P = \sigma(P')$, assim pela Lema de Projeção $P \underset{\beta}{\twoheadrightarrow} \sigma(M) = M$ e $P \underset{\beta}{\twoheadrightarrow} \sigma(N) = N$. $\square$

Proposição 12 *Seja $C \subseteq \Lambda^t_\sigma$ não trivial e fechado sobre $=_{\lambda\sigma}$. Então C é não-recursivo.*

Prova. Seja $D = C \cap \Lambda$, assim tomando $N \in D$ e $P \in \Lambda$ tal que $N =_\beta P$. Assim, temos que $\exists L \in \Lambda$ tal que $N \underset{\beta}{\twoheadrightarrow} L$ e $P \underset{\beta}{\twoheadrightarrow} L$ e pela Proposição 7 $N \underset{\lambda\sigma}{\twoheadrightarrow} P$ e $M \underset{\lambda\sigma}{\twoheadrightarrow} P$. Logo $N =_{\lambda\sigma} P$. Como C é fechado sobre $=_{\lambda\sigma}$ então $P \in C$, e assim $P \in D$, pois $P \in \Lambda$ e $P \in C$. Portanto D é fechado sobre $=_\beta$. Como $D \subset \Lambda$ temos pela Proposição 8 que D é não-recursivo. Logo se C for recursivo, o conjunto D também deveria ser recursivo, já que $D \subset C$, que é uma contradição, pois D é não-recursivo. $\square$

Possíveis aplicações no $\lambda\sigma$-cálculo

Nesta seção mostraremos possível uma aplicação da Proposição 10 no $\lambda\sigma$-cálculo, fazendo algumas modificações para adequar ao novo cálculo. Destacamos que a aplicação desses resultados ainda não é possível, necessitando-se ainda de algumas provas para a sua validade.

Proposição 13 *Dados $P, Q \in \Lambda$. Então o termo*

$$(\lambda(\underline{\mathbf{1}}[(\underline{\mathbf{1}} \cdot ((P \cdot id) \circ \uparrow))]\ \underline{\mathbf{1}}[(\underline{\mathbf{1}} \cdot ((Q \cdot id) \circ \uparrow))]))$$

é expansível a um termo puro se e somente se $\exists R \in \Lambda$ $R \underset{\beta}{\twoheadrightarrow} P$ e $R \underset{\beta}{\twoheadrightarrow} Q$.

Prova.

- ($\Leftarrow$) Suponha que existe um termo R. Então pela simulação de $\lambda\sigma$ temos que $R \underset{\lambda\sigma}{\twoheadrightarrow} P$ e $R \underset{\lambda\sigma}{\twoheadrightarrow} Q$, logo

$$\begin{array}{c}
(\lambda(\underline{\mathbf{1}}[(\underline{\mathbf{1}} \cdot ((P \cdot id) \circ \uparrow))]\ \underline{\mathbf{1}}[(\underline{\mathbf{1}} \cdot ((Q \cdot id) \circ \uparrow))])) \\
\Big\uparrow\!\!\uparrow {\scriptstyle \lambda\sigma} \\
(\lambda(\underline{\mathbf{1}}[(\underline{\mathbf{1}} \cdot ((R \cdot id) \circ \uparrow))]\ \underline{\mathbf{1}}[(\underline{\mathbf{1}} \cdot ((R \cdot id) \circ \uparrow))])) \\
\Big\uparrow {\scriptstyle App} \\
(\lambda((\underline{\mathbf{1}}\ \underline{\mathbf{1}})[(\underline{\mathbf{1}} \cdot ((R \cdot id) \circ \uparrow))]\ )) \\
\Big\uparrow {\scriptstyle Abs} \\
(\lambda(\underline{\mathbf{1}}\ \underline{\mathbf{1}}))[(R \cdot id)] \\
\Big\uparrow {\scriptstyle Beta} \\
((\lambda\lambda(\underline{\mathbf{1}}\ \underline{\mathbf{1}}))\ R)
\end{array}$$

que é um termo puro, $((\lambda\lambda(\underline{\mathbf{1}}\ \underline{\mathbf{1}}))\ R)$.

- ($\Rightarrow$) Teremos que provar, fazendo provavelmente uma análise de contexto, que $\exists R' \in \Lambda^t_\sigma$ tal que $R' \underset{\lambda\sigma}{\twoheadrightarrow} P$ e $R' \underset{\lambda\sigma}{\twoheadrightarrow} Q$. Então, por P e Q serem puros, pelo Lema 13, existirá um termo puro R tal que $R \underset{\beta}{\twoheadrightarrow} P$ e $R \underset{\beta}{\twoheadrightarrow} Q$.

$\square$

Capítulo 5

Conclusão

Estudamos neste trabalho o cálculo λ, assim como os cálculos $\lambda_\emptyset$, $\lambda\sigma$, $\lambda\upsilon$, e λx, olhando suas propriedades, para depois aplicá-las no estudo da expansibilidade.

Apresentamos a seguir uma tabela com as propriedades estudadas com as devidas referências.

	Simulação	Correção	Projeção	Expansível
$\lambda_\emptyset$	sim [2]	sim [2]		
λx	sim	sim		sim [3]
$\lambda\upsilon$	sim [9]	sim [9]	sim [9]	não [3]
$\lambda\sigma$	sim [1]		sim [1]	não*

Como pode-se verificar na tabela, a propriedade de simulação é válida para todos os cálculos estudados. A propriedade de correção é válida para os cálculos $\lambda_\emptyset$, λx e $\lambda\upsilon$. A propriedade de projeção é válida para os cálculos $\lambda\upsilon$ e $\lambda\sigma$. E a propriedade de ser expansível a termos puros vale para o cálculo λx, não vale para o $\lambda\upsilon$ e conjecturamos que não vale para o $\lambda\sigma$.

Para o cálculo λx provamos a simulação e a correção. A expansão foi feita por Ariel Arbiser, no qual a prova é desenvolvida por meio da indução na estrutura de termos.

Desenvolvemos para o $\lambda\upsilon$ lemas auxiliares, os Lemas 1 e 2, para provar

$$((\lambda M)\ N) \underset{\beta}{\rightarrow} \sigma_0(M, N)$$

e utilizar na prova da simulação e da correção. Com respeito a expansão, foram feitas muitas provas por Ariel Arbiser até chegar ao resultado apresentado na tabela. E aqui no trabalho detalhamos estas provas. A simulação e a correção foram utilizadas nas provas de indecidibilidade das relações $\underset{\lambda\upsilon}{\twoheadrightarrow}$ e $=_{\lambda\upsilon}$, que foram utilizadas por Ariel Arbiser no resultado de indecibilidade com relação a expansão a termos puro.

No $\lambda\sigma$-cálculo foi provada a simulação pela referência [1]. Não sabemos se a propriedade de correção vale para o este cálculo, tentamos provar mas sem sucesso.

Desenvolvemos todos os resultados de indecibilidade em $\lambda\sigma$ que se encontram na Seção 4.3.3, dentre eles, a indecibilidade das relações $\underset{\lambda\sigma}{\twoheadrightarrow}$ e $=_{\lambda\sigma}$. Sendo que utilizamos, para isso, lemas e proposições que foram usados no cálculo $\lambda\upsilon$, que modificamos para aplicá-los. Para provar estes resultados, fizemos uso, dentre outras relações, da simulação, da projeção.

No Lema 12, no Corolário 11 e na Proposição 12 fizemos uso do sistema definido na redução β do cálculo λ, usado na Proposição 7. E deixamos uma possível aplicação da Proposição 10 do $\lambda\upsilon$, no $\lambda\sigma$ que é a Proposição 13, que ainda falta provar.

A validade da Proposição 13 implicaria que o conjunto

$$\Lambda_\sigma^P = \{M \in \Lambda_\sigma^t | M \text{ tem uma expansão pura}\}$$

é não-recursivo. Para provar esta proposição estavamos analisando contextos da forma:

$$\begin{array}{lcl} C^t & ::= & \Box \mid C^t\ \Lambda_\sigma^t \mid \Lambda_\sigma^t\ C^t \mid \lambda\ C^t \mid C^t[\Lambda_\sigma^s] \mid \Lambda_\sigma^t[C^s] \\ C^s & ::= & C^t \cdot \Lambda_\sigma^s \ \mid \ (C^t \cdot \Lambda_\sigma^s) \circ \Lambda_\sigma^s \ \mid \ \Lambda_\sigma^s \circ (C^t \cdot \Lambda_\sigma^s) \end{array}$$

Conjecturamos que o $\lambda\sigma$-cálculo seja indecidível a respeito da expansão a termos puros. Apesar de não conseguimos provar isso, foi possível desenvolver uma base sólida a respeito da expansibilidade.

E não podemos esquecer de como é foi importante o Teorema de Scott, que aplicado ao cálculo λ, obteve como resultados de indecibilidadde $\underset{\beta}{\twoheadrightarrow}$, $=_\beta$, o Corolário 4 e a Proposição 8; que são a base para os resultados de indecibilidade dos cálculos $\lambda\upsilon$ e $\lambda\sigma$. Os resultados de indecibilidade do cálculo λ podem ser encontrados na referência [3].

Referências

[1] M. Abadi, L. Cardelli, P.-L. Curien, and J.-J. Lévy. Explicit substitutions. *Journal of Functional Programming*, 1:375–416, 1991. 24, 25, 28, 43, 46

[2] A. Arbiser. *Explicit Substituion Systems and Subsystems.* PhD thesis, Universidad de Buenos Aires, 2005. 9, 10, 12, 13, 16, 17, 46

[3] A. Arbiser. The expansion problem in lambda calculi with explicit substitution. *Journal of Logic and Computation*, 18(6):849–883, 2008. 1, 15, 30, 32, 33, 34, 46, 47

[4] M. Ayala-Rincón and F. L. C. Moura. Fundamentos da Programação Lógica e Funcional | O princípio de resolução e a teoria de reescrita | . Notas de aula, Departamentos de Matemática e de Ciência da Computação, Universidade de Brasília, 2012. Disponível na página `http://www.mat.unb.br/~ayala`. Em Português. 3, 4, 5

[5] M. Ayala-Rincón, F. L. C. Moura, and F. Kamareddine. Comparing and implementing calculi of explicit substitutions with eta reduction. *Annals of Pure and Applied Logic*, 134(1):5 – 41, 2005. 24

[6] M. Ayala-Rincón and C. Muñoz. Explicit Substitutions and All That. Technical Report 2000-45, NASA ICASE, 2000. 5, 24

[7] F. Baader and T. Nipkow. *Term rewriting and all that.* Cambridge University Press, New York, NY, USA, 1998. 3

[8] H. P. Barendregt. *Lambda Calculi with Types In: Handbook of Logic in Computer Scienc*, volume II. Oxford University Press, 1991. 1, 5, 9, 11, 33, 43

[9] Z. Benaissa, D. Briaud, P. Lescanne, and J. Rouyer-Degli. $\lambda\upsilon$, a calculus of explicit substitutions which preserves strong normalisation. *Journal of Functional Programming*, 6(05):699–722, 1996. 15, 16, 17, 19, 33, 34, 46

[10] M. Bezem, J. W. Klop, and R. Vrijer, editors. *Term Rewriting Systems by "Terese"*, volume 55 of *Cambridge Tracts in Theoretical Computer Science.* Cambridge University Press, 2003. 3

[11] C. Hankin. *An Introduction to Lambda Calculi for Computer Scientists*, volume 2 of *Text in Computing.* King's College Publications, 2nd edition, 2004. 5, 6, 29, 30

[12] F. L. C. Moura, , M. Ayala-Rincón, and F. Kamareddine. SUBSEXPL: a Tool for Simulating and Comparing Explicit Substitutions Calculi, 2006. 12