UNIVERSIDADE DE BRASÍLIA
INSTITUTO DE CIÊNCIAS EXATAS
DEPARTAMENTO DE MATEMÁTICA

Cálculos de Substituições Explícitas à la de Bruijn com Sistemas de Tipos com Interseção

Por

Daniel Lima Ventura

Brasília
2010

UNIVERSIDADE DE BRASÍLIA
INSTITUTO DE CIÊNCIAS EXATAS
DEPARTAMENTO DE MATEMÁTICA

Cálculos de Substituições Explícitas à la de Bruijn com Sistemas de Tipos com Interseção

Por

Daniel Lima Ventura[1]

Orientador: Prof. Mauricio Ayala-Rincón

[1]O autor contou com uma bolsa de Doutorado do CNPq, incluíndo uma bolsa para o Doutorado Sanduíche realizado na Heriot-Watt University em Edimburgo, Escócia.

à Aline.

‘When *I* use a word,’ Humpty Dumpty said in rather a scornful tone,‘it means just what I choose it to mean - neither more nor less.’
‘The question is,’ said Alice, ‘whether you *can* make words mean so many different things.’
‘The question is,’ said Humpty Dumpty,‘which is to be master - that’s all.’

Through the Looking Glass and what Alice found there.

Agradecimentos

A Deus, por ter me trazido são e salvo até este ponto.

A minha família, em especial a minha irmã Luciana.

Ao Professor Mauricio Ayala, pela orientação durante todo o doutorado, e à Professora Fairouz Kamareddine, por ter me recebido no seu grupo de pesquisa ULTRA e pelo total apoio durante e após o estágio de doutorado sanduíche.

Aos integrantes da banca examinadora pelo tempo dispensado na avaliação do presente trabalho.

Aos amigos Vincent Rahli e Laetitia de Freslon, pela agradável companhia durante minha estadia em Edimburgo.

Aos demais membros do grupo ULTRA, Joe Wells, Manuel Maarek, Robert Lamar, Jan Jakubův, Sébastien Carlier e Krzysztof Retel.

Aos amigos do Departamento de Matemática, Evander, Walter, André, Jhone, Nilton e João Paulo. Em especial, às integrantes do Grupo de Teoria da Computação, Daniele, Andréia e Thaynara.

Aos integrantes prestativos do Departamento de Matemática da UNB, professores e funcionários, em especial à Eveline, Manoel e Pereira.

Às amigas Natália, Paloma, Mariana, Pilar, Paula, Julia, Silvie e Tais e aos amigos Lucas, Rogério, Ruiz, Pierre e Gonzalo, pelas atividades extraclasse.

Ao CNPq, pelo suporte financeiro integral, incluíndo o doutorado sanduíche.

Resumo

O λ-calculus é um modelo teórico de computação tão antigo quanto a própria noção de *função computável*. Devido a definição da substituição como uma metaoperação, existem várias formas de tornar esta substituição explícita no sistema, dando surgimento a uma grande variedade de sistemas baseados no λ-calculus. Estudamos dois cálculos de substituições explícitas, o $\lambda\sigma$ e o λs_e, com sistemas de tipos com interseção. Estes cálculos utilizam uma notação à la de Bruijn, onde variáveis são representadas por índices ao invés de nomes. Sistemas de atribuição de tipos permitem uma análise sintática (estática) de propriedades semânticas (dinâmicas) de programas, dispensando qualquer declaração de tipos dentro destes. Os tipos com interseção apresentam uma maneira de integrar polimorfismo ao sistema, que tem se mostrado conveniente computacionalmente com propriedades como a tipagem principal que permite, *e.g.* a compilação separada e a recompilação inteligente para o sistema de tipos computacionais. Para a adição de tipos com interseção aos cálculos estudados, fazemos um estudo do λ-calculus à la de Bruijn com dois sistemas de tipos diferentes. Uma caracterização sintática de tipagens principais, para termos irredutíveis, em um dos sistemas é apresentada. Baseado neste sistema, introduzimos sistemas de tipos com interseção para o $\lambda\sigma$ e o λs_e. A propriedade básica de redução de sujeito, que garante a preservação dos tipos em qualquer computação possível para termos tipáveis, é analisada nas variações dos sistemas propostos. Outra propriedade analisada é a relevância do sistema, garantido que apenas a informação de tipos necessária para inferência é utilizada, impossibilitando a admissibilidade de uma lei de redundância para o sistema de tipos.

Abstract

The λ-calculus is a well known theoretical computation model as old as the concept of *computable functions*. Due to the substitution definition as a meta-operator there exists a great quantity of variations of this computational system in which the operation of substitution is treated explicitly. In this work we investigate intersection type systems for two explicit substitution calculi, the $\lambda\sigma$ and the λs_e, both with de Bruijn indices. Type assignment systems allow one to have a static code analysis through implicit typing inference, where no type declaration is required. Intersection types present a machine-friendly way to add polymorphism to type systems with features such as the principal typing property, allowing *e.g.* a *separate compilation* and the *smartest recompilation*. We study the λ-calculus with de Bruijn indices with two different type systems, in a preliminary step for adding intersection types for both explicit substitution calculi. A characterisation for principal typings of irreducible terms is given in one of the systems, in which the intersection type systems for each $\lambda\sigma$ and λs_e are based on. We analyse the subject reduction property, which guarantees that all terms of the system preserve their types during any possible computation, in some variations for the proposed type systems. Another analysed property is the relevance, in which only necessary suppositions are allowed in a typing inference, turning a weakening rule inadmissible in the type system.

Lista de símbolos

$\mathfrak{R}$	sistema de reescrita de termos
$\rightarrow_{\mathfrak{R}}$	relação de redução de termos induzida por $\mathfrak{R}$
$\rightarrow_{\mathfrak{R}}^{+}$	fecho transitivo de $\rightarrow_{\mathfrak{R}}$
$\rightarrow_{\mathfrak{R}}^{*}$	fecho reflexivo transitivo de $\rightarrow_{\mathfrak{R}}$
$=_{\mathfrak{R}}$	fecho reflexivo transitivo e simétrico de $\rightarrow_{\mathfrak{R}}$
$\mathfrak{R}(t)$	forma $\mathfrak{R}$-normal de t
$\mathfrak{S}$	sistema de atribuição de tipos
Θ	tipagem
$\langle \Gamma \vdash \tau \rangle$	tipagem com contexto Γ e tipo τ
$M : \langle \Gamma \vdash \tau \rangle$	tipagem $\langle \Gamma \vdash \tau \rangle$ do termo M
$\mathfrak{S} \Vdash M : \Theta$	Θ é uma tipagem de M em $\mathfrak{S}$
$M : \langle \Gamma \vdash_{\mathfrak{S}} \tau \rangle$	$\langle \Gamma \vdash \tau \rangle$ é uma tipagem de M em $\mathfrak{S}$
$\cap$	interseção de conjuntos
$\cup$	união de conjuntos
$\in$	pertence
$\subseteq, \subset$	está contido
$\supseteq, \supset$	contém

$\uplus$	união aditiva para multiconjuntos
$A \setminus B$	complementar do conjunto B em relação ao conjunto A
$a \in^n A$	a tem exatamente n cópias em um multiconjunto A
$\mathbb{N}^*$	conjunto dos números naturais positivos
$C+k$, $C\backslash k$, $C_{>k}$, $C_{<k}$, $C_{\leq k}$	subconjuntos gerados a partir de $C \subset \mathbb{N}^*$ e $k \geq 0$
$Termos_{\mathfrak{S}}(\Theta)$	conjunto de termo que tem a tipagem Θ em $\mathfrak{S}$
Λ	conjunto de termos em λ
Λx	conjunto de termos em λx
Λ_{dB}	conjunto de termos em λ_{dB}
Λs	conjunto de termos em λs e em λs_e
$\Lambda\sigma$	conjunto de expressões em $\lambda\sigma$
$\Lambda\sigma^t$	classe dos termos em $\lambda\sigma$
$\Lambda\sigma^s$	classe das substituições em $\lambda\sigma$
$\mathcal{V}$	conjunto de variáveis de termo
$AV(M)$	conjunto das variáveis disponíveis do termo M
$FV(M)$	conjunto de variáveis livres do termo M
$Var(t)$	conjunto de variáveis do termo t
$AI(M)$	conjunto de índices disponíveis do termo M
$FI(M)$	conjunto de índices livres do termo M
$\overline{FI}(M)$	multiconjunto de índices livres do termo M
$\mathcal{X}$	conjunto de metavariáveis
$\mathcal{S}$	conjunto de tipo simples

$\mathcal{T}$	conjunto de tipos com interseção restrita
$\mathcal{U}'$	conjunto de tipos com interseção restrita e idempotente
$FO(\alpha,\Gamma)$	conjunto de tipos em Γ com α como ocorrência final
$L(T)$	conjunto de subtermos à esquerda do $\mathcal{C}$-tipo T
$\mathcal{A}$	conjunto de variáveis de tipo
$TV(\tau)$	conjunto de ocorrências de variáveis de tipo
$\mathcal{P}$	conjunto de pares principais em $\mathcal{C} \times \mathcal{T}_{NF}$
$sup(M)$	o maior índice livre de M
$sav(M)$	o maior índice disponível de M
$\leq_{\mathfrak{S}}$	relação de ordem para tipagens em $\mathfrak{S}$
$\leqslant$	relação binária para tipos com interseção restrita idempotente
$\sqsubseteq$, $\sqsubset$	relação de inclusão para tipos com interseção restrita
$\Gamma' \sqsubseteq \Gamma$, $\Gamma' \sqsubset \Gamma$	extensões de $\sqsubseteq$, $\sqsubset$ para contextos
$T \sqsubseteq T$, $T \sqsubset T$	extensões de $\sqsubseteq$, $\sqsubset$ para $\mathcal{C}$-tipos
Γ_n	n-ésimo elemento do contexto sequencial Γ
$\Gamma_{<n}$, $\Gamma_{\leq n}$, $\Gamma_{>n}$, $\Gamma_{\geq n}$	subsequências do contexto sequencial Γ
$\omega^{\underline{n}}$	contexto formado por ω's de comprimento n
ω^M	contexto $\omega^{\underline{m}}$ tal que $m = sup(M)$
Γ/Γ'	contexto $\Delta \neq \omega^{\underline{n}}$, tal que $\Gamma = \Gamma' \wedge \Delta$
$\Gamma\downharpoonright_M$	restrição para contextos em λ_{dB}^{SM} e em λs^{SM}
$\Gamma\upharpoonright_M$	restrição para contextos em $\lambda_{dB}^{\sqcap}$

s	substituição de tipos
$[x/N]M$	metasubstituição em λ
$\{\underline{n}/N\}M$	metasubstituição β em λ_{dB}
M^{+}	elevação de $M \in \Lambda_{dB}$
M^{+i}	i-elevação de $M \in \Lambda_{dB}$
$\lambda^{\rightarrow}$	sistema de tipos simples para λ
$\lambda_{dB}^{\rightarrow}$	sistema de tipos simples para λ_{dB}
$\lambda_{dB}^{SM}/\lambda_{dB}^{SM_r}$	sistemas de tipos com interseção restrita para λ_{dB}
$\lambda_{dB}^{\sqcap}$	sistema de tipos com interseção restrita e idempotente para λ_{dB}
$\lambda {s_e}^{\rightarrow}$	sistema de tipos simples para λs_e
λs^{SM}	sistema de tipos com interseção para λs
$\lambda {s_e}^{\wedge}$	sistema de tipos com interseção para λs_e
$\lambda\sigma^{\rightarrow}$	sistema de tipos simples para $\lambda\sigma$
$\lambda\sigma_r^{\wedge}$	sistema restrito de tipos com interseção para $\lambda\sigma$
$\lambda\sigma^{\wedge}$	sistema de tipos com interseção para $\lambda\sigma$

Sumário

Capítulo 1

Introdução

O λ-calculus é um modelo para computação equivalente às máquinas de Turing [93]. Por sua sintaxe simples e sua expressividade, problemas como a chamada a procedimentos em linguagens de programação podem ser investigados em sua essência nesse modelo teórico de computação [6]. Além disso, linguagens de programação funcional, como o LISP [73] e ML [45], têm o λ-calculus como fundamento teórico em suas respectivas especificações. Assim, estudos relacionados à implementação de linguagens baseadas nesse sistema formal apresentam novos paradigmas a sua teoria clássica.

A formalização da operação de substituição em Lógica Matemática apresenta desafios, como garantir que o resultado de uma fórmula com uma variável substituída por outra fórmula expresse o mesmo que a fórmula original. Como o exemplo apresentado em [36], o x na fórmula $\exists z.z + z \equiv x$, que na aritmética é satisfeita pelos números pares, ao ser trocado por z obtemos a fórmula $\exists z.z + z \equiv z$, que na mesma interpretação é satisfeita apenas por 0. Assim, para que a substituição não altere as relações descritas em uma fórmula, esta deve garantir que variáveis que não são ligadas a um quantificador permaneçam "livres" na fórmula resultante.

A substituição desempenha um papel crucial na definição de computação no λ-calculus e está definida como uma metaoperação, chamada de metasubstituição. Nesse sentido, as substituições explícitas apresentam uma tentativa em diminuir a diferença entre o modelo teórico representado pelo λ-calculus e suas possíveis implementações. Os cálculos de substituições explícitas estendem o cálculo original incluindo a substituição em sua definição de computação. Assim, a substituição realizada simultaneamente no modelo clássico é dividida em passos menores, permitindo que apenas parte seja executada enquanto o restante fica suspenso. Essa característica evita, por exemplo, que ocorra um saturação da memória (*overflow*) durante a execução de um programa. O trabalho considerado seminal dessa área introduz o $\lambda\sigma$, onde o λ-calculus é estendido com o cálculo σ, que computa as substituições [1]. Apesar desse cálculo simular as computações possíveis na teoria original, termos que são fortemente terminantes no λ-calculus podem ser computados indefinidamente no $\lambda\sigma$. Um processo fortemente terminante é um processo que sempre termina, não importando em que ordem suas instruções são executadas. Portanto, em cálculos de substituições explícitas surge o paradigma de preservação de terminação forte (PSN), onde o tipo de composição para substituições permitido no cálculo tem um papel central [84].

O problema em saber se um programa é terminante ou não tem grande destaque em ciência da computação. Os sistemas de tipos se apresentam com uma ferramenta onde propriedades semânticas, como terminação, podem ser verificadas de maneira estática. Os tipos aparecem pela primeira vez em linguagens de programação na década de 50, como uma maneira de otimizar os recursos disponíveis, *e.g.* distinção na representação de números como inteiro e ponto flutuante. Atualmente, os tipos em computação desempenham papéis como a detecção estática e dinâmica de erros e a possibilidade de abstração. Ainda na década de 70, os cientistas da computação identificaram as similaridades dos sistemas de tipos estudados em computação e os estudados em lógica matemática. Houve um grande avanço nesse campo de pesquisa, através do estudo da relação dos sistemas de tipos computacionais e as propriedades das lógicas associadas, conhecida como isomorfismo de Curry-Howard [91]. Um sistema de tipos atribui tipos aos programas considerados seguros, por este sistema, para a execução. A redução de sujeito (SR) é uma propriedade básica para sistema de tipos, que garante a invariância da tipagem atribuída ao objeto pelo sistema, em relação as suas possíveis computações. Em outras palavras, um programa considerado seguro para execução permanece seguro durante todo o seu processamento.

Um sistema de atribuição de tipos para o λ-calculus, chamado *tipos simples*, atribui tipos apenas para processos fortemente terminantes [49]. O problema de inferência de tipos nesse sistema é decidível, ou seja, existe um algoritmo tal que dado um termo de λ, que corresponde a um programa, decide se este tem algum tipo no sistema. O inconveniente desse sistema é a falta de suporte ao polimorfismo, uma característica presente nas linguagens de programação modernas. Como consequência, existem termos fortemente terminantes que não têm tipo nesse sistema com tipos simples.

Existem algumas alternativas para o tratamento do polimorfismo. O sistema de tipos polimórfico mais conhecido, e estudado, é o sistema HM [28, 75] utilizado pela linguagem ML. O sistema HM tem tipos implícitos, onde nenhuma declaração de tipos é feita dentro do programa, cabendo ao compilador a checagem da correção do programa através da inferência de tipo em HM. O polimorfismo suportado neste sistema é uma restrição de um sistema mais expressivo, o λ-calculus tipado de segunda ordem [82], ou simplesmente $\lambda 2$. Enquanto o sistema com tipos simples para o λ-calculus está relacionado através do isomorfismo de Curry-Howard ao fragmento proposicional da lógica intuicionista, o $\lambda 2$ está relacionado ao sistema F de Girard [43]. Os problemas de verificação e inferência de tipos são indecidíveis neste sistema [102]. O problema de verificação de tipos consiste em, dados um programa e um tipo, verificar se este programa tem este tipo no sistema. Apesar do sistema HM ter ambos os problemas decidíveis, este não permite um algoritmo de inferência modular, outra característica presente em liguagens de programação modernas.

Uma alternativa ao $\lambda 2$ para o suporte ao polimorfismo são os tipos com interseção (IT). O polimorfismo é tratado de modo finitário em tipos com interseção, listando os múltiplos tipos assumidos por um mesmo programa. Os sistemas de IT foram introduzidos em 1978 por M. Coppo e M. Dezani [21] com o intuito de caracterizar os termos fortemente terminantes do λ-calculus e G. Pottinger apresenta em 1980 o primeiro sistema com tal caracterização [81]. Observa-se que decidir a caracterização de tais termos é equivalente a decisão do problema da parada [92] logo o problema de inferência de tipos para tal sistema é indecidível. Em 1983 H. Barendregt, Coppo e Dezani apresentaram o sistema BCD de tipos com interseção, completo em relação à semântica [9]. Por sua importância em relação à apresentação do sistema de IT como uma maneira de investigação para a semântica do λ-calculus, ele se tornou a principal referência no estudo de tais sistemas. Por ser muito flexível, a inferência de tipos é extremamente complexa (sendo realizada por procedimentos de semidecisão, já que a inferência é indecidível). Assim, uma restrição de BCD com as mesmas propriedades é introduzido em [94, 95], onde algumas variantes de sistemas de IT são estudadas. Uma propriedade satisfeita em algumas dessas variantes, que não é satisfeita nos sistemas HM e $\lambda 2$, é a tipagem principal.

A propriedade de tipagem principal (PT) vem sendo valorizada recentemente, pois possibilita a modularização de um sistema de tipos computacional [52]. Em sistemas modulares, cada parte de um programa chamada de módulo precisa apenas de sua assinatura para a compilação deste programa, sem que os detalhes sobre o funcionamento deste módulo sejam conhecidos. Ou seja, o verificador de tipos de uma linguagem modular precisa saber apenas quais são os tipos dos parâmetros passados ao módulo e qual o tipo do objeto que este retorna, para inferir se a aplicação do mesmo está correta no sistema. Essa assinatura é chamada de tipagem, que é composta pelo par formado de contexto e tipo, onde o contexto é um conjunto de declarações de tipos para variáveis. Através do isomorfismo de Curry-Howard, o contexto corresponde ao conjunto de suposições em uma dedução formal. Dado um programa, qualquer tipagem deste em um sistema com PT é obtida a partir de sua tipagem principal, através de operações sintáticas. Assim, para verificar se o programa pode ser integrado a um sistema computacional maior, basta verificar se o tipo exigido pelo sistema pode ser obtido de sua tipagem principal, dispensando a recompilação de todo o sistema com o novo programa incluído. Essa característica permite a compilação separada e a recompilação inteligente (*smartest recompilation*) [52], onde um programa é tratado como um módulo, tendo sua PT como sua assinatura, e que só é necessária a sua recompilação caso sua especificação seja alterada.

Uma outra característica possível em sistemas de IT é a relevância. Em sistemas baseados em relevância [29], o contexto de uma tipagem para o termo designa tipos para todas e apenas as variáveis livres do termo. Assim, sistemas de IT para o λ-calculus que são relevantes não são extensões conservativas do sistemas de *tipos simples*.

Os tipos com interseção têm se mostrado convenientes computacionalmente em seu tratamento para o polimorfismo, com propriedades como PT [24,86,87,94,95]. Além disso, é uma ferramenta poderosa na investigação da semântica do λ-calculus [2, 9, 23, 24, 50]. Para cálculos de substituições explícitas (ES), sistemas de IT têm sido estudados para o λx [69] e para o $\bar{\lambda}x$ [35, 65], a versão com substituições explícitas do $\bar{\lambda}$-calculus, o λ-calculus isomorfo a sistemas de Gentzen [48]. Porém, nenhum trabalho para cálculos de ES onde a composição de substituições é permitida foi realizado até o momento. Como foi provado por E. Ritter em [84] de uma maneira geral, tal iteração entre substituições faz com que o cálculo não preserve a normalização forte de termos do λ-calculus (propriedade de PSN). Assim, os sistemas de IT podem contribuir na investigação de propriedades de redução e da semântica destes cálculos de ES.

Assim, este trabalho tem por finalidade a adição de um sistema de tipos com interseção como suporte ao polimorfismo para cálculos de substituições explícitas que não são PSN. Introduzimos tipos com interseção para o $\lambda\sigma$ e λs_e, ambos com índices de de Bruijn. Investigamos o λ_{dB}-calculus, o λ-calculus à la de Bruijn, com dois sistemas de IT como uma etapa preliminar à adição de IT aos cálculos de ES. Em todos os sistemas apresentados, a propriedade de SR tem um papel central nas considerações e, no caso dos cálculos de ES, nas modificações propostas para a obtenção de um sistema de tipos adequado. Um outra característica levada em consideração para o desenvolvimento de tais sistemas é a propriedade de relevância. Apresentamos portanto, o primeiro trabalho com sistemas de IT, com o seu polimorfismo finitário, para cálculos que buscam a proximidade entre o modelo de computação teórico e possíveis implementações.

No Capítulo 2 apresentamos o fundamento teórico onde introduzimos algumas definições, e notações, para conjuntos, multiconjuntos e sistemas de reescrita. Além disso, os sistemas de atribuição de tipos com a propriedade de tipagem principal, o λ-calculus com tipos simples, as substituições explícitas e os tipos com interseção são explicados com um grau maior de detalhamento neste capítulo. No Capítulo 3 apresentamos o λ-calculus à la de Bruijn e os cálculos de substituições explícitas λs, λs_e e $\lambda\sigma$, com os respectivos sistemas de tipos simples, e suas propriedades. Nos Capítulos de 4 a 8 são apresentadas as contribuições deste trabalho:

- O Capítulo 4 apresenta o trabalho publicado em [98], onde introduzimos a noção de tipagem principal para os cálculos com os sistemas de tipos simples apresentados no capítulo anterior, provando que as respectivas noções de PT para cada sistema são corretas e completas.

- No Capítulo 5 apresentamos dois sistemas de IT para o λ_{dB} com suas respectivas propriedades. Na Seção 5.1 introduzimos um sistema baseado no sistema apresentado em [37] por E. Sayag e M. Mauny enquanto na Seção 5.2 introduzimos um sistema baseado no sistema apresentado em [56] por F. Kamareddine e K. Nour. O último é um trabalho publicado em [100].

- No Capítulo 6 apresentamos o trabalho publicado em [101], que apresenta uma caracterização de PT para formas β-normais em λ_{dB} no sistema da Seção 5.1.

- No Capítulo 7 introduzimos o sistema de tipos com interseção para o λs_e-calculus, provando que este possui a propriedade de SR. Nesse capítulo apresentamos um sistema de IT para λs como um estudo preliminar.

- No Capítulo 8 apresentamos o sistema de IT para o $\lambda\sigma$-calculus, provando a propriedade de SR para o mesmo.

Por fim, no Capítulo 9 concluímos o trabalho e apresentamos as possibilidades de trabalhos futuros.

Capítulo 2

Fundamentos

2.1 Elementos básicos e notação

Nessa seção, apresentamos a notação utilizada no restante do documento para conjuntos, multiconjunto e sistemas de reescrita.

2.1.1 Conjuntos e multiconjuntos

A noção de **conjunto**, com a relação de **pertença** ($\in$) e **inclusão**, composta por **estar contido** ($\subseteq, \subset$) e **conter** ($\supseteq, \supset$), e as operações de **interseção** ($\cap$) e **união** ($\cup$) são as usuais de teoria dos conjuntos. Sejam A e B dois conjuntos, $A \setminus B$ denota o conjunto formado pelos elementos de A que não estão em B. A seguir, apresentamos a noção de multiconjunto.

Definição 2.1.1 (Multiconjuntos):

1. Um **multiconjunto** é um conjunto onde elementos podem ter um número finito de repetições.

2. Para um elemento a e um multiconjunto A, $a \in^n A$ denota que a tem exatamente n repetições em A, onde $a \in^0 A$ denota $a \notin A$.

3. Seja $\uplus$, a **união aditiva** ou **aglutinação**, definida por: Se $a \in^n A$ e $a \in^m B$ então $a \in^{n+m} A \uplus B$.

4. A n-aglutinação de um multiconjunto A é denotada por A^n.

Observe que, para todo multiconjunto, existe um conjunto associado. Seja $\mathbb{N}^* = \mathbb{N} \setminus \{0\}$, ou seja, o conjunto dos números inteiros sem o 0. A seguir, apresentamos algumas definições para conjuntos com elementos em $\mathbb{N}^*$, similar aos apresentados em [61].

Definição 2.1.2 (Subconjuntos em $\mathbb{N}^*$): Sejam $C \subset \mathbb{N}^*$ e $k \geq 0$. Define-se os seguintes conjuntos:

1. $C \backslash k = \{n - k \,|\, n \in C, n > k\}$;
2. $C + k = \{n + k \,|\, n \in C\}$;
3. $C_{>k} = \{n \in C \,|\, n > k\}$;
4. $C_{<k} = \{n \in C \,|\, n < k\}$ e $C_{\leq k} = \{n \in C \,|\, n \leq k\}$.

Note que as definições acima são estendidas aos multiconjuntos com elementos em $\mathbb{N}^*$ de maneira direta. A seguir, apresentamos algumas propriedades destes subconjuntos.

Proposição 2.1.3: Sejam $C, D \subset \mathbb{N}^*$ e $k, k' \geq 0$. A seguintes igualdades são satisfeitas:

1. $(C \cup D) \backslash k = (C \backslash k) \cup (D \backslash k)$;
2. $(C \cup D) + k = (C + k) \cup (D + k)$;
3. $(C \cup D)_{>k} = C_{>k} \cup D_{>k}$;
4. $(C \cup D)_{<k} = C_{<k} \cup D_{<k}$ e $(C \cup D)_{\leq k} = C_{\leq k} \cup D_{\leq k}$;
5. $(C \backslash k) \backslash k' = C \backslash (k + k')$;
6. $(C + k) \backslash 1 = C + (k - 1)$ se $k \geq 1$;
7. $(C \backslash k) + k = C_{>k}$;
8. $(C \backslash 1)_{<k} = (C_{<k+1}) \backslash 1$ e $(C \backslash 1)_{\leq k} = (C_{\leq k+1}) \backslash 1$;
9. $(C \backslash 1)_{>k} = (C_{>k+1}) \backslash 1$.

As igualdades apresentadas na Proprosição 2.1.3 acima são verdadeiras para multiconjuntos, com $\uplus$ no lugar da união para conjuntos.

2.1.2 Sistemas de reescrita

Assumimos que os conceitos da Teoria de Reescrita são de conhecimentos do leitor [5]. Nesta seção apresentamos a notação utilizada para sistemas de reescrita e a definição para algumas das propriedades, mencionadas nos capítulos posteriores. A seguir, uma breve menção de alguns conceitos presentes nesses sistemas.

Um **sistema abstrato de redução** é composto pelo par $(A, \rightarrow)$, formado de um conjunto e uma relação binária sobre este conjunto, onde $(a, a') \in \rightarrow$ é denotado por $a \rightarrow a'$. Em **sistemas de reescrita de termos** (TRS), os objetos do sistema são termos. Um conjunto de **termos** é definido indutivamente, a partir de um **conjunto de variáveis de termo** $\mathcal{V}$ e um **conjunto de símbolos de função** $\mathcal{F}$. Uma **regra de reescrita** é um par ordenado de termos, denotado por $l \rightarrow r$, onde l e r são tais que:(1) $l \notin \mathcal{V}$. (2) $Var(r) \subseteq Var(l)$, onde Var denota o conjunto de variáveis em um termo.

Um TRS é composto de um conjunto de termos e um conjunto de regras de reescrita. Assim, um TRS $\mathfrak{R}$ induz uma relação de redução de termos $\rightarrow_{\mathfrak{R}}$, onde uma instância da regra pode ser aplicada a um termo de forma compatível a sua estrutura de formação. Ou seja, dada a regra $l \rightarrow r$ e uma uma substituição de termos s, mapeando variáveis de termos a termos, se $s(l)$ é subtermo de t, então $t \rightarrow t'$, onde t' é o termo t com a ocorrência de $s(l)$ trocada por $s(r)$. Um subtermo tal que uma regra de $\mathfrak{R}$ seja aplicável é chamado de **$\mathfrak{R}$-redex**. Uma **forma $\mathfrak{R}$-normal**, ou $\mathfrak{R}$-nf, é um termo que não tenha ocorrências de $\mathfrak{R}$-redices.

O fecho transitivo de $\rightarrow_{\mathfrak{R}}$ é denotado por $\rightarrow^{+}_{\mathfrak{R}}$, o fecho reflexivo transitivo denotado por $\rightarrow^{*}_{\mathfrak{R}}$ e o fecho reflexivo transitivo e simétrico denotado por $=_{\mathfrak{R}}$. A seguir, definimos algumas propriedades relativas a TRS.

Definição 2.1.4 (Propriedades de TRS): Sejam $\mathfrak{R}$ e $\mathfrak{R}'$ TRS quaisquer:

1. **Teminação fraca (WN)**: Um termo t é **fracamente terminante** em $\mathfrak{R}$ se a partir de t existe uma sequência de reduções em $\mathfrak{R}$ tal que uma forma $\mathfrak{R}$-normal é obtida.

2. **Teminação forte (SN)**: Um termo t é **fortemente terminante** em $\mathfrak{R}$ se não existem derivações infinitas em $\mathfrak{R}$ a partir de t. Ou seja, qualquer sequência de reduções a partir de t atinge uma forma $\mathfrak{R}$-normal.

3. **Confluência (CR)**: $\mathfrak{R}$ é **confluente** se para todo termo t tal que $t \to^*_{\mathfrak{R}} t_1$ e $t \to^*_{\mathfrak{R}} t_2$, existe uma termo r tal que $t_1 \to^*_{\mathfrak{R}} r$ e $t_2 \to^*_{\mathfrak{R}} r$.

4. **Simulação**: $\mathfrak{R}'$ **simula** $\mathfrak{R}$ se $t \to_{\mathfrak{R}} t'$ implica $t \to^+_{\mathfrak{R}'} t'$.

5. **Preservação de teminação forte (PSN)**: Um $\mathfrak{R}'$ que simula $\mathfrak{R}$ tem a propriedade de **preservação de terminação forte** se todo termo SN em $\mathfrak{R}$ é SN em $\mathfrak{R}'$.

Observe que se um termo é SN então este é WN. Dizemos que um termo é **terminante** se é WN. Ao pensarmos em sistemas de reescrita de termos como modelos computacionais, a propriedade de confluência significa consistência das possíveis computações no sistema. A proriedade de WN significa que existe uma estratégia de redução para a obtenção de uma resposta enquanto a propriedade de SN significa que uma resposta é obtida não importando a ordem de execução das respectivas computações.

Se um TRS $\mathfrak{R}$ é WN e CR então para qualquer termo t, existe uma única forma $\mathfrak{R}$-normal obtida a partir de t, chamada de a forma $\mathfrak{R}$-normal de t, denotada por $\mathfrak{R}(t)$.

2.2 Sistemas de tipos e tipagem principal

Os tipos em matemática e lógica vêm sendo utilizados de maneira implícita desde o surgimento dessas ciências. A idéia fundamental é a de classificar objetos em sistemas formais. De acordo a F. Kamareddine em [55], Euclides em seu tratado sobre geometria, *Os Elementos*, usa implicitamente essa noção quando, a partir da definição de ponto e reta como classes de objetos diferentes, define objetos mais complexos, cada um pertencente a uma classe. Nesse sistema, e em outros introduzimos posteriormente, a intuição sobre o funcionamento das operações permitia uma definição destas usando a noção de tipos informalmente, através da metalinguagem. A partir do século 19, quando os sistemas se tornam mais complexos e abstratos, a formalização para essa classificação de objetos torna-se necessária. G. Frege apresentou em [41] a primeira formalização da lógica, onde o conceito de função presente em Análise Real foi estendido, com algumas restrições sobre quais poderiam ser os argumentos de uma função dada. Porém, a falta de formalização da definição dos tipos dos objetos estudados em [42], onde ele estende o sistema apresentado em [41] para descrever partes elementares da aritmética, permite que o equivalente ao paradoxo de Russell para o sistema seja codificado. A Teoria de Tipos aparece explicitamente pela primeira vez em [89], onde B. Russel identifica a característica em comum dos paradoxos conhecidos até então, sendo esta a possibilidade de autorreferência que um objeto possui dentro do sistema. Os tipos surgem então como uma maneira de evitar autorreferência nos sistemas formais, também conhecida como reflexividade. Nessa abordagem, os paradoxos lógicos são tratados com uma alteração na linguagem em contraposição à abordagem adotada em Teoria dos Conjuntos, onde os axiomas são alterados para tratar o problema [104].

Em Ciência da Computação, os tipos apareceram como uma forma de otimizar os recursos disponíveis. De acordo com B. Pierce em [80], o primeiro sistema de tipos, no início dos anos 50, era utilizado para distinguir entre as representações dos números como inteiros e pontos flutuantes. Como exemplo tem-se o sistema de tipos primitivos na linguagem Fortran. Durante a década de 50 e 60, essa classificação dos objetos foi sendo ampliada para dados mais complexos. Nos anos 70, conceitos mais elaborados estavam presentes nos sistemas de tipos, como sistemas modulares e subtipos. Na mesma época, os cientistas da computação identificaram as similaridades dos sistemas de tipos estudados em computação e os estudados em lógica matemática. O **isomorfismo de Curry-Howard** apresenta a correspondência entre sistemas de lógica formal e sistemas de tipos computacionais [91]. O primeiro sistema de tipos em que essa semelhança foi percebida foi o λ-calculus simples tipado, introduzido na seção a seguir, que tem uma correspondência com a lógica intuicionista proposicional [49]. Os aspectos sintáticos desta correspondência podem ser investigados em vários níveis, tais como fórmulas que correspondem aos tipos, provas no sistema lógico que correspondem a programas, normalização de provas que correspondem a terminação de programas. Assim, o isomorfismo de Curry-Howard ressalta a importância no estudo de sistemas de atribuição de tipos como uma ferramenta no estudo da semântica de programas.

Um **sistema de atribuição de tipos** $\mathfrak{S}$, ou simplesmente sistema de tipos, é um conjunto de regras de dedução, que permitem a designação de tipos para alguns objetos aos quais o sistema se refere. Os contextos fornecem a informação necessária, usada pelas regras de $\mathfrak{S}$ para esta designação de tipos aos objetos. Um objeto ao qual uma atribuição de tipo é possível em $\mathfrak{S}$ é chamado de **tipável** (em $\mathfrak{S}$). No λ-calculus, os objetos de um sistema de tipos são os termos em λ.

O par ordenado $\langle\Gamma \vdash \tau\rangle$ de um contexto e um tipo em um sistema $\mathfrak{S}$ é chamado de uma **tipagem em $\mathfrak{S}$**. Para um termo M, $\Gamma \vdash M : \tau$ é a notação usual para: "M tem o tipo τ no contexto Γ". Usaremos a notação $M : \langle\Gamma \vdash \tau\rangle$, de forma que fique destacado que o par seja entendido como um predicado do termo M. Nesse caso, $\langle\Gamma \vdash \tau\rangle$ é chamado de uma **tipagem de** M. Se $\Theta = \langle\Gamma \vdash \tau\rangle$ é uma tipagem em $\mathfrak{S}$ então $\mathfrak{S} \Vdash M : \Theta$ denota que Θ é uma tipagem de M em $\mathfrak{S}$, também denotado por $M : \langle\Gamma \vdash_{\mathfrak{S}} \tau\rangle$. Sempre que estiver claro a qual sistema a tipagem pertence, usaremos a notação simplificada $\langle\Gamma \vdash \tau\rangle$.

Dado um termo M, uma questão interessante é se esse é ou não tipável em $\mathfrak{S}$ e se esse **problema de tipabilidade** é decidível. Uma propriedade importante para sistemas de tipos é a **redução de sujeito**, ou simplesmente SR. Essa propriedade garante que, ao ser atribuído um tipo para um objeto de um sistema, qualquer transformação (computação) permitida no sistema não altera sua tipabilidade em relação ao tipo atribuído. Como exemplo, se a atribuição de tipos para um programa em um sistema significa que o programa é correto e seguro, essa propriedade garante que o processo de execução desse programa é correto e seguro em todos os seus passos. Outra questão importante é se existe uma **tipagem principal**. A tipagem principal, que é independente do contexto, não deve ser confundida com tipo principal, dependente de um determinado contexto de tipo. Em [52], T. Jim salienta a diferença entre as duas propriedades e a importância da primeira. A tipagem principal, ou simplesmente PT, de um termo em um sistema de tipos é uma tipagem que representa qualquer outra tipagem do termo neste sistema. Em [103], J. Wells apresenta uma definição destas tipagens mais gerais, que representam outras tipagens de um termo no sistema, independente do sistema de tipos. Para apresentar essa definição precisamos introduzir algumas definições preliminares.

Seja Θ uma tipagem em $\mathfrak{S}$ e $\mathbf{Termos}_{\mathfrak{S}}(\Theta) = \{M \mid \mathfrak{S} \Vdash M{:}\Theta\}$. Uma tipagem Θ_1 é **mais forte** do que uma tipagem Θ_2 se, e somente se, $Termos_{\mathfrak{S}}(\Theta_1) \subset Termos_{\mathfrak{S}}(\Theta_2)$. A partir do conceito de mais forte, uma ordem para tipagens é apresentada.

Definição 2.2.1 ([103]): Seja $\Theta_1 \leq_{\mathfrak{S}} \Theta_2 \iff Termos_{\mathfrak{S}}(\Theta_1) \subseteq Termos_{\mathfrak{S}}(\Theta_2)$.

Assim, podemos apresentar a definição de PT independente de sistemas.

Definição 2.2.2 (Tipagem principal [103]): Uma tipagem Θ em um sistema $\mathfrak{S}$ é principal para algum termo M se $\mathfrak{S} \Vdash M : \Theta$ e para qualquer Θ' tal que $\mathfrak{S} \Vdash M : \Theta'$ tem-se que $\Theta \leq_{\mathfrak{S}} \Theta'$.

A Definição 2.2.2 introduzida por Wells representa uma generalização para a noção de PT, que usualmente depende de definições específicas para cada sistema de tipos. Essas definições específicas se baseiam em operações sintáticas características de cada sistema. Portanto, para cada definição específica de PT, existe uma **operação sintática associada**. Dado um termo, uma tipagem é dita principal em tal definição se qualquer outra tipagem do termo no sistema pode ser obtida a partir da primeira através dessas operações sintáticas associadas.

Uma operação sintática comum para essas definições específicas é a substituição de tipos. Quando os tipos são definidos tendo variáveis como tipos atômicos ao invés de constantes, podemos definir uma subsituição agindo sobre essas variáveis. Em geral, as substituição são mapeamentos do conjunto de variáveis para o conjunto de tipos, onde a extensão para o conjunto de tipos é direta. A seguir introduzimos os tipos simples para exemplificar os conceitos discutidos e apresentar mais alguns, utilizados nos capítulos posteriores.

Definição 2.2.3 (Tipos simples):

1. Seja $\mathcal{A}$ um **conjunto de variáveis de tipo**, infinito e enumerável, com seus elementos representados por α, β.
2. O conjunto $\mathcal{S}$ de **tipos simples** é definido por:

$$\tau, \sigma, \rho \in \mathcal{S} ::= \mathcal{A} \,|\, \mathcal{S} \rightarrow \mathcal{S}$$

Tem-se então que os elemento de $\mathcal{S}$ são os tipos atômicos, representados pelas variáveis de tipo, e os tipos funcionais. A associação para a simplificação de parêntese é da direita para a esquerda. Assim, $\alpha_1 \rightarrow \alpha_2 \rightarrow \alpha_3$ denota $\alpha_1 \rightarrow (\alpha_2 \rightarrow \alpha_3)$. Com os tipos simples definidos desta maneira, podemos definir as substituições de tipos.

Definição 2.2.4 (Substituição para tipos simples):

1. A **substituição de tipos** mapeia variáveis de tipo em $\mathcal{A}$ para tipos em $\mathcal{S}$.

2. Dada uma substituição de tipos $s : \mathcal{A} \rightarrow \mathcal{S}$, a extensão de s para o domínio dos tipos funcionais é direta, onde $s(\sigma \rightarrow \tau) = s(\sigma) \rightarrow s(\tau)$.

O **conjunto de ocorrências de variáveis de tipo** em $\tau \in \mathcal{S}$ é denotado por $TV(\tau)$. Uma substituição de tipos bijetiva de variáveis em variáveis é chamada de **renomeamento**. Observe que, em um renomeamento s para um tipo τ, se $\alpha, \beta \in TV(\tau)$ tal que $\alpha \not\equiv \beta$, então $s(\alpha) \not\equiv s(\beta)$, onde $s(\alpha), s(\beta) \in \mathcal{A}$. Uma característica importante é que os habitantes de uma tipagem são invariantes em relação a renomeamentos. Ou seja, $Termos_{\mathfrak{S}}(\Theta) = Termos_{\mathfrak{S}}(\Theta')$ para $\Theta' = s(\Theta)$ onde s é um renomeamento. Logo, qualquer renomeamento de uma tipagem principal é principal.

Um outro problema interessante em sistemas de tipos é o **problema de habitação**. Neste problema, dado um tipo τ em um sistema $\mathfrak{S}$, a questão é se existe algum objeto M no sistema tal que $M : \langle \emptyset \vdash_{\mathfrak{S}} \tau \rangle$. Ou seja, $Termos_{\mathfrak{S}}(\Theta) \neq \emptyset$, para $\Theta = \langle \emptyset \vdash \mathfrak{S} \rangle$. Através do isomorfismo de Curry-Howard podemos, a partir de uma prova no sistema dedutivo associado, extrair um programa correto. Assistentes de prova modernos como o Coq [10] e o PVS [79] são basedos nesta correspondência.

2.3 O λ-calculus com tipos simples

O λ-calculus foi introduzido em 1932 por A. Church [16] como parte de uma teoria baseada na noção de função. O sistema tinha por finalidade propor uma teoria geral para funções e estendê-la com noções lógicas de forma a prover a fundamentação para a lógica e partes da matemática [8]. Com uma sintaxe simples, a computação no λ-calculus é feita a partir de uma única regra, a contração β. Porém, S. Kleene e B. Rosser provaram em 1935 a inconsistência do sistema proposto por Church [67]. Contudo, em 1936 Church apresenta uma prova da indecidibilidade das sentenças de primeira ordem, utilizando o λ-calculus, e define a noção de função computável [17].

Ainda no ano de 1936, Church e Rosser apresentam em [19] a consistência para uma subteoria do λ-calculus, o chamado λI-calculus, mostrando a confluência da redução β, composta por uma sequência de contrações β. Ao propor a noção de funções **definíveis em $\boldsymbol{\lambda}$** e a partir de alguns resultados obtidos por Kleene, Church conjectura a sua Tese de Church, onde afirma que todas as funções intuitivamente computáveis são definíveis em λ. Uma importante evidência para a sua conjectura é apresentada com a prova de Kleene em [66] sobre a coindidência entre as funções definíveis em λ e as funções ***μ*-recursivas**, definidas em 1934 por K. Gödel [44]. Paralelamente, A. Turing define a noção de **computável** em [92], utilizando as suas máquinas de Turing. Turing prova em 1937 que a noção de definível em λ e a sua noção de computável são equivalentes [93]. Portanto, com essa demostração temos a adequabilidade computacional do λ-calculus.

A noção de função presente no λ-calculus é a de um processo definido que inicia com argumentos e calcula os valores correspondentes, em contraposição à noção de função como uma relação, representada por pares ordenados ou pelo seu gráfico. Para denotar uma função $f(x)$ no λ-calculus, usamos a notação $\lambda x.F$, onde F é a representação da função como um termo do λ-calculus. A operação de aplicação no λ-calculus permite que a função $\lambda x.F$ seja aplicada a um termo A, denotada por $(\lambda x.F\ A)$. No cálculo sem tipos, um termo pode ser aplicado a qualquer outro termo. Assim, podemos representar no λ-calculus características como "procedimentos como argumento de procedimentos" e "procedimentos como valores retornados por procedimentos". Portanto, a representação de funções computáveis através de termos em λ fez surgir a chamada **programação funcional**. Além disso, problemas em Computação como a chamada a procedimentos aparecem em sua forma pura e o estudo desses problemas ajuda no desenvolvimento, e análise, de linguagens de programação em geral.

2.3.1 O λ-calculus e propriedades

Nessa seção apresentamos o λ-calculus sem tipos e suas propriedades. A referência para essa seção é o livro de H. Barendregt sobre o λ-calculus [6], onde estão compilados os principais resultados sobre esta teoria.

Definição 2.3.1 (O conjunto Λ): Seja $\mathcal{V}$ o **conjunto de variáveis de termos**, infinito enumerável. O **conjunto de termos em $\boldsymbol{\lambda}$**, denotado por Λ, é definido indutivamente para $x \in \mathcal{V}$ por:

$$M, N \in \Lambda \ \ ::= \ \ x \,|\, (M\ N) \,|\, \lambda x.M$$

Os elementos de $\mathcal{V}$ são representados por x, y e z. Os termos são construídos a partir do conjunto de variáveis e duas operações básicas que são a **aplicação** e a **abstração**. A aplicação de M em N é denotada pelo termo $(M\ N)$ enquanto que a abstração, ou ligação, da variável x em M é denotada pelo termo $\lambda x.M$ onde λx é o abstrator. Os termos da forma $((\ldots((M_1\ M_2)\ M_3)\ldots)\ M_n)$ são denotados por $(M_1\, M_2\, \cdots\, M_n)$.

A seguir, definimos os subtermos de um termo em λ.

Definição 2.3.2: Seja $M \in \Lambda$. Os **subtermos** de M são definidos indutivamente por:

(i) Se $M \in \mathcal{V}$ então M é subtermo de M.

(ii) Se $M \equiv \lambda x.N$, então os subtermos são o próprio M e todos os subtermos de N.

(iii) Se $M \equiv (M_1\ M_2)$, então os subtermos são todos os subtermos de M_1 e M_2 e o próprio termo M.

A **variável** x **ocorre em** M se x é um subtermo de M. Se x ocorre em N tal que $\lambda x.N$ é um subtermo de M, então a ocorrência de x é chamada **ligada**. As ocorrência de M que não são ligadas são chamadas **livres**. Observe que x pode ter ocorrências livres e ligadas em um mesmo termo. O **conjunto de variáveis livres** de um termo M é denotado por $FV(M)$ e um termo que não tem variáveis livres é chamado **fechado**.

A contração β é a regra usada para a computação no λ-calculus e a sua definição precisa do conceito de substituição. No λ-calculus clássico, a substituição está definida como uma metaoperação, chamada de **metasubstituição**. Ou seja, a definição de como esta substituição é efetivamente realizada não faz parte do sistema formal, sendo esta realizada em um metanível. Assim, para um termo M, define-se $[x/N]M$ como sendo o termo resultante da substituição de todas as ocorrências livres de x em M pelo termo N [49]. Assim temos uma definição implícita de substituição para termos em λ. A regra a seguir representa um papel importante em definições formais desta substituição implícita.

Definição 2.3.3 (Conversão $\boldsymbol{\alpha}$): Seja $\lambda x.M$ um termo em λ e $y \notin FV(M)$. A **conversão $\boldsymbol{\alpha}$** de $\lambda x.M$ é definida por:

$$\lambda x.M \to_\alpha \lambda y.[x/y]M$$

A condição $y \notin FV(M)$ é necessária para garantir que não exista captura de variáveis livres de M. Senão teríamos por exemplo que

$$\lambda x.(\lambda y.(xzy)yx) \rightarrow_\alpha \lambda z.(\lambda y.(zzy)yz)$$

mudando o termo λ para outro que representa uma função diferente da função original. A conversão α representa um renomeamento de variáveis ligadas. Seja $=_\alpha$ o fechamento simétrico, reflexivo e transitivo de α. Assim, as α-classes de termos são formadas por termos que representam a mesma função.

Definição 2.3.4 (Contração $\boldsymbol{\beta}$): Seja $(\lambda x.M\ N)$ um termo em λ. A **contração $\boldsymbol{\beta}$** é definida por:

$$(\lambda x.M\ N) \rightarrow_\beta [x/N]M$$

Um termo da forma $(\lambda x.M\ N)$ é chamado **$\boldsymbol{\beta}$-redex**. Um termo sem β-redices como subtermo é chamado de **forma $\boldsymbol{\beta}$-normal**. A **redução $\boldsymbol{\beta}$** é definida como o fechamento transitivo da contração β, compatível com a estrutura dos termos em λ. O teorema a seguir estabelece a confluência desta regra de redução.

Teorema 2.3.5 (CR para redução $\boldsymbol{\beta}$ [6]): O conjunto Λ com a redução β é confluente.

Portanto, se um termo em λ tem uma forma β-normal, ela é única. O lema a seguir apresenta uma decrição geral de uma termo em β-nf.

Lema 2.3.6 (Descrição para formas β-normais [6]): O conjunto das formas β-normais em λ pode ser descrito indutivamente como:

(i) $x \in \mathcal{V}$

(ii) $\lambda x.N$, onde N é uma forma β-normal.

(iii) $(x\ N_1 \cdots N_m)$, onde $\forall 1 \leq j \leq m$, N_j é uma forma β-normal.

No restante dessa seção apresentamos alguns resultados e conceitos utilizados nos capítulos posteriores. Uma subteoria do λ-calculus tem destaque desde o seu surgimento, sendo a teoria originalmente investigada por Church, o **λI-calculus** (*cf.* Cap. 9 de [6]). Nessa teoria a formação de termos é restrita tal que apenas a abstração de variáveis que ocorrem livremente no termo são permitidas. Ou seja, o termo $\lambda x.M$ é permitido em λI

apenas para termos M tais que $x \in FV(M)$. Seja Λ_I o conjunto dos termos formados com essa restrição. No trabalho desenvolvido, os termos do λ-calculus, também chamado de λK-calculus, que não estão em Λ_I desempenham um papel importante. Observe que em um termo desses, existe um β-redex que cancela o argumento da aplicação. Em outras palavras, para $\lambda x.M$ tal que $x \notin FV(M)$ temos que $(\lambda x.M\ N) \rightarrow_\beta M$. Essa redução é chamada de **redução $\boldsymbol{\beta}$ nula** e a substituição associada $[x/N]M = M$ de **substituição nula**. O teorema a seguir estabelece que a redução β não cria variáveis livres novas.

Teorema 2.3.7 ([6]): Se $M \rightarrow_\beta N$, então $FV(N) \subseteq FV(M)$.

A seguir, discriminamos os casos para os termos de Λ_I e de $\Lambda \setminus \Lambda_I$ em relação ao Teorema 2.3.7 acima.

Lema 2.3.8 ([6]):

1. Se $M \rightarrow_\beta N$ e $M \in \Lambda_I$, então $FV(M) = FV(N)$.
2. Se $M \rightarrow_\beta N$ e $M \in \Lambda \setminus \Lambda_I$, então $FV(N) \subset FV(M)$.

Uma regra também estudada para o λ-calculus é a contração η, definida a seguir.

Definição 2.3.9 (Contração $\boldsymbol{\eta}$): Seja $\lambda x.(M\ x)$ um termo em λ tal que $x \notin FV(M)$. A **contração $\boldsymbol{\eta}$** é definida por:

$$\lambda x.(M\ x) \rightarrow_\eta M \qquad \text{se } x \notin FV(M)$$

Portanto, a contração η é uma regra condicional. Observe que a equivalência $=_\eta$ induzida pela regra acima induz o axioma $\lambda x.(M\ x) =_\eta M$, de **igualdade extensional**, para o λ-calculus.

Um conceito usado como referência para estabelecer os termos "sem sentido" do λ-calculus é a solubilidade de um termo. Um termo fechado M é chamado de **solúvel** se este reduz ao termo $I \equiv \lambda x.x$ quando aplicado a um número finito de argumentos. Um termo M qualquer de Λ é chamado de solúvel quando o termo fechado correspondente, *i.e.* o termo $\lambda\vec{x}.M$ onde $\{\vec{x}\} = FV(M)$, é solúvel. Assim, os termo que não são solúveis são chamados de **insolúveis** e são considerados os termos sem sentido algum do λ-calculus [6]. Um exemplo de termo insolúvel é o autorreprodutor $(\lambda x.(x\ x)\ \lambda x.(x\ x))$.

2.3.2 O sistema $\lambda^{\rightarrow}$, o λ-calculus com tipos simples

O primeiro sistema de tipos para o λ-calculus foi proposto em 1940 por Church [18], com uma teoria de tipos simples. Nesse sistema, a sintaxe dos termos em λ é alterada, incluindo uma anotação do tipo atribuído para a variável no abstrator. Como exemplo, para cada tipo τ temos a função identidade $\lambda x{:}\tau.x$ com tipo $\tau \to \tau$. Essa versão à la Church é conhecida como o **λ-calculus tipado**, que considera apenas os termos que possuem um tipo no sistema. Assim, termos como $(x\ x)$, chamado de **autoaplicação**, e $(\lambda x.(x\ x)\ \lambda x.(x\ x))$, chamado de **autorreprodutor**, não fazem parte da versão tipada.

Em 1958 H. Curry apresentou um sistema de atribuição de tipos simples para o λ-calculus [27]. Nessa versão à la Curry, todos os elementos de Λ fazem parte do sistema de tipos mas apenas alguns são tipáveis neste sistema. A sintaxe do λ-calculus é inalterada em sistemas de atribuição de tipos. Assim, ao invés de uma função identidade para cada tipo, o termo $\lambda x.x$ representa esta função, que tem o tipo atribuído dependendo do argumento ao qual ela é aplicada. Portanto, temos um forma de polimorfismo em sistemas à la Curry, também chamados de sistemas com tipos implícitos.

Para um sistema de tipos para o λ-calculus precisamos dos contextos de tipo, que provêm informação de tipo para as variáveis livres de um termo.

Definição 2.3.10 (Contextos):

1. Os **contextos de tipo**, ou simplesmente contexto, são conjuntos finitos de designações de tipo para variáveis de termo.

2. Dado um contexto Γ, cada variável de termo tem no máximo uma designação de tipo.

3. $x \in \Gamma$ denota que $x : \tau \in \Gamma$ para algum tipo τ. Caso contrário, $x \notin \Gamma$.

O conjunto de tipos $\mathcal{S}$ do sistema é o apresentado na Definição 2.2.3. Assim, um contexto Γ tem a forma $\{x_1{:}\tau_1,\ \ldots,\ x_m{:}\tau_m\}$ onde $\tau_1, \ldots, \tau_m \in \mathcal{S}$. A seguir, apresentamos um sistema de tipos simples para o λ-calculus, similar ao sistema TA_λ de J. Hindley [49].

Definição 2.3.11 (O sistema $\lambda^{\rightarrow}$): O sistema $\lambda^{\rightarrow}$ é composto pelas seguintes regras:

$$\text{(Start)} \qquad \frac{x \notin \Gamma}{x:\langle\{x{:}\tau\} \cup \Gamma \vdash \tau\rangle}$$

$$\text{(Abs)} \qquad \frac{x \notin \Gamma \qquad M:\langle\{x{:}\sigma\} \cup \Gamma \vdash \tau\rangle}{\lambda x.M:\langle\Gamma \vdash \sigma \rightarrow \tau\rangle}$$

$$\text{(App)} \qquad \frac{M:\langle\Gamma \vdash \sigma \rightarrow \tau\rangle \qquad N:\langle\Gamma \vdash \sigma\rangle}{(M\ N):\langle\Gamma \vdash \tau\rangle}$$

Observe que usamos a notação $M : \langle\Gamma \vdash \tau\rangle$ para dizer que M tem tipo τ em um contexto Γ, ao invés da notação $\Gamma \vdash M : \tau$, usual na literatura. A seguir, apresentamos algumas propriedades de $\lambda^{\rightarrow}$.

Teorema 2.3.12 (Propriedades para $\lambda^{\rightarrow}$ [49]):

1. **(SR para redução $\boldsymbol{\beta}$)** Se $M:\langle\Gamma \vdash_{\lambda^{\rightarrow}} \tau\rangle$ e $M \rightarrow_{\beta} N$ então $N:\langle\Gamma \vdash_{\lambda^{\rightarrow}} \tau\rangle$.

2. **(SN para termos tipáveis)** Se $M:\langle\Gamma \vdash_{\lambda^{\rightarrow}} \tau\rangle$ então M é SN.

3. **(Decidibilidade da tipabilidade)** O problema de tipabilidade para $\lambda^{\rightarrow}$ é decidível.

4. **(Decidibilidade da habitação)** O problema de habitação para $\lambda^{\rightarrow}$ é decidível.

Seja M um termo em λ qualquer. Pelo item 3 do teorema acima podemos escrever um algoritmo que decide se M tem um tipo em $\lambda^{\rightarrow}$. Caso tenhamos uma reposta afirmativa, pelo item 2 sabemos que este termo é terminante, não importanto a ordem em que as reduções β sejam realizadas. Ou seja, nesse sistema de tipos obtemos propriedades semânticas para alguns termos do λ-calculus de maneira estática.

A decidibilidade da tipabilidade, e da verificação de tipos, em $\lambda^{\rightarrow}$ é uma consequência da decidibilidade do problema de PT neste sistema. Em [49], Hindley mostra que todo termo tipável em $\lambda^{\rightarrow}$ tem PT apresentando um algoritmo que decide a tipabilidade de um termo e retorna a tipagem principal, chamada de par principal (*principal pair*), caso a resposta seja afirmativa. Por conter idéias centrais do trabalho desenvolvido, apresentamos as noções utilizadas na demonstração da **propriedade de PT**.

Primeiramente, a substituição de tipos apresentada na Definição 2.2.4 é estendida de forma direta para contextos e tipagens em $\lambda^{\rightarrow}$. Ou seja, $s(\langle\Gamma \vdash \tau\rangle) = \langle s(\Gamma) \vdash s(\tau)\rangle$, onde

$s(\Gamma)$ representa a aplicação de s em cada elemento de Γ, tal que $s(x : \sigma) = x : s(\sigma)$ e $s(\emptyset) = \emptyset$. Com essa definição de substituição de tipos, apresentamos uma proposição que aparece como uma observação em [49].

Proposição 2.3.13: Se $M : \langle \Gamma \vdash_{\lambda^{\rightarrow}} \tau \rangle$, então $M : \langle s(\Gamma) \vdash_{\lambda^{\rightarrow}} s(\tau) \rangle$ para qualquer substituição de tipos s.

Demonstração. Indução na derivação de $M : \langle \Gamma \vdash_{\lambda^{\rightarrow}} \tau \rangle$. Basta observar que, no passo indutivo, as condições laterais das regras de inferência são satisfeitas após a aplicação da substituição nas respectivas premissas. $\square$

Uma noção importante utilizada no algoritmo de inferência de PT para $\lambda^{\rightarrow}$ apresentado em [49] são as **variáveis de tipo novas**. O algoritmo é aplicado recursivamente na estrutura do termo em λ. A cada passo em que o algoritmo precise escolher um elemento de $\mathcal{A}$, uma variável que ainda não foi utilizada em nenhum dos passos anteriores é escolhida. Um outro fator importante é o de que a unificação necessária em um passo recursivo é de primeira ordem. Logo, o algoritmo de unificação de primeira ordem de Robinson [85] garante **unificadores mais gerais**, ou simplesmente mgu, em cada passo do algoritmo. Apresentamos a seguir a definição de PT para $\lambda^{\rightarrow}$, como introduzida em [103].

Definição 2.3.14 (Tipagem principal de Hindley [103]): Uma **tipagem principal em** $\lambda^{\rightarrow}$ de um termo M é uma tipagem $\Theta = \langle \Gamma \vdash \tau \rangle$ tal que:

1. $\lambda^{\rightarrow} \Vdash M : \Theta$.

2. Se $\lambda^{\rightarrow} \Vdash M : \Theta'$ para alguma tipagem $\Theta' = \langle \Gamma' \vdash \tau' \rangle$, então existe alguma substituição s tal que $s(\Gamma) \subseteq \Gamma'$ e $s(\tau) = \tau'$.

A definição de *principal pair* em [49] tem uma pequena diferença em relação a definição acima. As tipagens obtidas a partir de um par principal de um termo M não incluem a possibilidade de uma lei de redundância. Assim, apenas as tipagens que têm contextos com designações para todas e apenas as variáveis em $FV(M)$ são obtidas a partir de um par principal. Ou seja, $s(\Gamma) = \Gamma'$, onde $\Theta' = \langle \Gamma \vdash \tau' \rangle$ é uma tipagem de M em $\lambda^{\rightarrow}$. Para obtermos a tipagem principal como na Definição 2.3.14, apresentamos o lema a seguir.

Lema 2.3.15 (Lei de redundância em $\lambda^{\rightarrow}$): Seja M um termo em λ. Se $M : \langle \Gamma \vdash_{\lambda^{\rightarrow}} \tau \rangle$, então $M : \langle \Gamma' \vdash_{\lambda^{\rightarrow}} \tau \rangle$ para qualquer contexto $\Gamma' \supseteq \Gamma$. Assim, a regra (Weak) abaixo é dedutível em $\lambda^{\rightarrow}$.

$$\text{(Weak)}\ \frac{x \notin \Gamma \qquad M : \langle \Gamma \vdash \tau \rangle}{M : \langle \{x{:}\sigma\} \cup \Gamma \vdash \tau \rangle}$$

Demonstração. Pelo Lema 2A9.1 em [49]. □

A **lei de redundância** (ou *weakening*) permite que sejam inseridas informações desnecessárias aos contextos. Note que uma tipagem em $\lambda^{\rightarrow}$ é principal se, e somente se, é um par principal. Elas diferem apenas na definição de quais operações são permitidas na obtenção das tipagens do termo relacionado. Em [49] PT denota *principal type*, porém como o algoritmo retorna pares principais e estes por sua vez são tipagens principais em $\lambda^{\rightarrow}$, mantemos o nome do algoritmo, mas com PT denotando tipagem principal.

Em [103], Wells mostra que a Definição 2.3.14, específica para o sistema $\lambda^{\rightarrow}$, é equivalente à Definição 2.2.2. Portanto, a tipagem principal de Hindley é uma definição mais geral para o sistema $\lambda^{\rightarrow}$ do que os pares principais e as operações sintáticas associadas a essa definição são a substituição de tipos e a redundância. Assim, apresentamos o teorema a seguir estabelecendo que $\lambda^{\rightarrow}$ satisfaz a propriedade de PT de acordo à Definição 2.3.14.

Teorema 2.3.16 (PT para o $\lambda^{\rightarrow}$): O sistema $\lambda^{\rightarrow}$ satisfaz a propriedade de tipagem principal.

Demonstração. Seja M um termo em λ. Se M é tipável em $\lambda^{\rightarrow}$ então seja $\Theta = \langle \Gamma \vdash \tau \rangle$ o par principal retornado pelo algoritmo de inferência de PT de Hindley [49]. Para uma tipagem $\Theta' = \langle \Gamma' \vdash \tau' \rangle$ qualquer de M tem-se pelo Lema 2A11 em [49] que $M : \langle \Gamma' \restriction M \vdash \tau' \rangle$, onde $\Gamma' \restriction M$ denota a restrição de Γ' ao conjunto $FV(M)$. Assim, pela propriedade da par principal tem-se que existe uma substituição de tipos s tal que $s(\Gamma) = \Gamma' \restriction M$ e $s(\tau) = \tau'$. Observe que $s(\Gamma) \subseteq \Gamma'$. □

Podemos observar que os problemas mencionados na Seção 2.2, e que são interessantes para sistemas de tipos, são todos decidíveis em $\lambda^{\rightarrow}$. Porém, o polimorfismo suportado nesse sistema é limitado, onde termos em λ que correspondem a programas seguros não são tipáveis . Assim, diferentes maneiras de incluir o polimorfismo e ampliar o conjunto de termos tipáveis têm sido estudados. Na Seção 2.5 os tipos com interseção são apresentados como uma alternativa conveniente computacionalmente.

2.4 Substituições explícitas

A definição da contração β no λ-calculus tem a substituição como uma metaoperação. Implementações que simulem o λ-calculus precisam tratar a substituição de maneira explícita, o que torna o sistema implementado diferente do modelo teórico. Assim, **cálculos de substituições explícitas**, ou simplesmente ES, surgiram como uma extensão do λ-calculus, onde a substituição é parte integrante do cálculo. Em 1991, M. Abadi *et al.* introduziram o $\lambda\sigma$-calculus no trabalho considerado seminal em ES [1], não obstante cálculos de ES estão presentes anteriormente em sistemas computacionais. Um exemplo notável é o AUTOMATH [77], sendo [31] de 1978 a primeira apresentação formal de um cálculo de ES. O $\lambda\sigma$ é apresentado na Seção 3.3. Nesta seção utilizamos o λx [11, 71, 88] para introduzir os conceitos de substituições explícitas utilizados nos capítulos posteriores. O conjunto de termos de λx, para o conjunto de variáveis de termos $\mathcal{V}$ como na Definição 2.3.1, é apresentado abaixo.

Definição 2.4.1 (O conjunto Λx): O **conjunto de termos em $\boldsymbol{\lambda}$x**, denotado por Λx, é definido indutivamente para $x \in \mathcal{V}$ por:

$$M, N \in \Lambda\mathrm{x} \quad ::= \quad x \mid (M\ N) \mid \lambda x.M \mid M\langle x := N\rangle$$

O conjunto Λx estende o conjunto Λ com os termos da forma $M\langle x := N\rangle$, que representam o início do processo que corresponde ao termo $[x/N]M$. Portanto, ao invés de uma notação para o resultado de uma metasubstituição, temos um termo no cálculo que representa substituições pendentes. O termo N é chamado de **corpo da substituição** $\langle x := N\rangle$. Os termos em Λ são chamados de **termos puros**, ou seja os termos sem ocorrência de substituição.

O processo de simulação é iniciado com a regra (b) abaixo.

$$(\lambda x.M\ N) \to M\langle x := N\rangle \qquad \text{(b)}$$

A substituição é então computada pelo x-calculus, apresentado na Figura 2.4.1. Portanto, o λx-calculus é composto pela regra (b) e o x-calculus, este sendo o **cálculo de**

$$
\begin{array}{rll}
(\mathrm{xv}) & x\langle x := N\rangle \to N & \\
(\mathrm{xvgc}) & x\langle y := N\rangle \to x & \text{se } x \not\equiv y \\
(\mathrm{xab}) & (\lambda x.M)\langle y := N\rangle \to \lambda x.M\langle y := N\rangle & \text{se } x \not\equiv y \,\mathrm{e}\, x \notin FV(N) \\
(\mathrm{xap}) & (M_1\ M_2)\langle y := N\rangle \to (M_1\langle y := N\rangle\ M_2\langle y := N\rangle) &
\end{array}
$$

Figura 2.4.1: O cálculo de substituição x

substituição associado a λx.

Note que, na Figura 2.4.1, as condições laterais da regra (xab) são garantidas pelo renomeamento de variáveis ligadas. Logo o x-calculus, e consequentemente o λx-calculus, está definido módulo α-equivalência de termos. O fecho simétrico, transitivo e reflexivo de x é denotado por $=_{\mathrm{x}}$. A seguir, apresentamos algumas propriedades de λx e do cálculo de substituição associado.

Teorema 2.4.2 (Propriedades para λx/x-calculi [11]):

1. **(SN e CR para** x**)** O x-calculus é fortemente terminante e confluente.
2. **(Extensão conservativa para redução $\boldsymbol{\beta}$)** Se $M, N \in \Lambda$, então $M \to_\beta N$ sse $M \to^+_{\lambda\mathrm{x}} N$.
3. **(CR para $\boldsymbol{\lambda}$x)** O λx é confluente.
4. **(PSN para $\boldsymbol{\lambda}$x)** Se um termo puro é SN para β então este é SN em λx.

Dado um termo M em λx, seja x(M) a forma normal correspondente. Note que x(M) está bem definido para todo termo, pois o **x**-calculus é fortemente terminate e confluente, e que x(M) é um termo puro. O conjunto de variáveis livres é similar ao definido para termos em λ, estendido por $FV(M\langle x := N\rangle) = (FV(M) \setminus \{x\}) \cup FV(N)$. A seguir apresentamos os lemas que relacionam o conjunto de variáveis livres de um termo e as reduções em λx.

Lema 2.4.3 ([11]): Sejam $M, N \in \Lambda$x:

1. Se $M \to_{\mathrm{b}} N$ então $FV(M) = FV(N)$

2. Se $M \rightarrow_{\mathrm{x}} N$ então $FV(N) \subseteq FV(M)$

Apesar da semelhança com o Teorema 2.3.7 para o λ-calculus, o conjunto FV não representa uma relação apropriada para a identificação de uma substituição nula. O exemplo a seguir ilustra o problema em tentar identificar uma substituição nula usando o conceito de variáveis livres.

Exemplo 2.4.1: Sejam $M \equiv x\langle y := z\rangle$ e $N \equiv y\langle y := z\rangle$ logo $FV(M) = \{x, z\}$ e $FV(N) = \{z\}$. Apesar de $z \in FV(M)$ e $z \in FV(N)$, temos que $M\langle z := w\rangle$ representa uma substituição nula, ao contrário de $N\langle z := w\rangle$.

Em [69], S. Lengrand *et al.* introduziram o conceito de *available variables* para o $\lambda\mathrm{x}$. Como a composição para substituições não é permitida, a idéia para as **variáveis disponíveis** é a de incluir as ocorrências de variáveis dos subtermos que não sejam ligadas e que não integrem o corpo de uma substituição nula. Apesar de parecer redundante a primeira vista, a definição é indutiva na estrutura dos termos. Assim, o conjunto de variáveis disponíveis para $M \in \Lambda$ coincide com $FV(M)$ logo $M\langle x := N\rangle$ representa uma substituição nula se, e somente se, $x \notin FV(M)$. A seguir apresentamos a definição formal do conjunto de variáveis disponíveis de um termo.

Definição 2.4.4: Seja $M \in \Lambda\mathrm{x}$. O **conjunto das variáveis disponíveis** de M, denotado por $AV(M)$, é definido indutivamente por:

(i) $AV(x) = \{x\}$

(ii) $AV(\lambda x.M) = AV(M) \setminus \{x\}$

(iii) $AV(M\ N) = AV(M) \cup AV(N)$

(iv) $AV(M\langle x := N\rangle) = \begin{cases} (AV(M) \setminus \{x\}) \cup AV(N) & \text{se } x \in AV(M) \\ AV(M) & \text{se } x \notin AV(M) \end{cases}$

A partir da definição acima, temos um conjunto de variáveis ocorrendo em um termo M tal que: (1) Se M é um termo puro então $FV(M) = AV(M)$. (2) $M\langle x := N\rangle$ representa uma substituição nula sse $x \notin AV(M)$.

O $\lambda\mathrm{x}$ não é confluente para os chamados **termos abertos** [62]. A sintaxe do λ-calculus, e do $\lambda\mathrm{x}$, pode ser estendida tal que inclua as chamadas **metavariáveis** de termos.

As metavariáveis de termos, que podem ser vistas como buracos no termo, representam as variáveis do problema de unificação de ordem superior (HOU), onde as variáveis de termo são tratadas como constantes. Os termos que têm ocorrências de metavariáveis são chamados de termos abertos e para obtermos a confluência para estes termos, a composição de substituições deve ser permitida em cálculos de SE, que são utilizados em implementações para HOU [3, 33].

O $\lambda\sigma$-calculus, que permite a composição, é confluente na respectiva extensão para a inclusão de metavariáveis para termos [34]. Outro cálculo, que também permite a composição para obter a confluência em termos abertos é o λs_e de F. Kamareddine e A. Ríos [59], apresentado na Seção 3.2. Assim como para o λ-calculus, um aspecto importante do ponto de vista computacional é a adição de um sistema de tipos para estes cálculos de SE. E surpreendentemente, na tentativa de estender os dois cálculos supracitados com um sistema de tipos simples perde-se a propriedade do λ-calculus, de normalização forte para termos tipáveis. Após P.-A. Melliès ter apresentado em 1995 um contraexemplo para o $\lambda\sigma$ [74], onde um termo com tipos simples tem uma estratégia de reduções infinitas, B. Guillaume apresentou em 2000 um contraexemplo análogo para o λs_e [47]. Em [74], Melliès aponta a regra ($MapEnv$), ou regras similares a esta, como a causa da não terminação em uma série de cálculos de substituições explícitas (*cf.* [40]).

Em [84], E. Ritter identificou a ligação entre a composição permitida para estes cálculos, chamada de composição completa (*full composition*), e o problema com a propriedade de PSN . Apenas recentemente D. Kesner apresentou o cálculo de SE λex [63], satisfazendo simultaneamente as propriedades desejáveis para um mecanismo de simulação da contração β. Apesar disso, o λex tem as variáveis de termo representadas por nomes, e portanto sendo um sistema de reescrita módulo α-equivalência de termos.

2.5 Os tipos com interseção

Sistemas de tipos com interseção (IT), introduzido por M. Coppo e M. Dezani-Ciancaglini em [21, 22] como uma extensão para o sistema $\lambda^{\rightarrow}$, têm sido amplamente estudados tanto pelo seu aspecto teórico, como a análise de modelos para o λ-calculus e de propriedades referentes à terminação, quanto pelo seu aspecto prático, com aplicações como a análise de programas baseadas em tipo e a detecção de erros. Em sistemas de IT, $M : \tau \cap \sigma$ denota que o termo M tem os tipos τ e σ. Apesar dos IT serem denotados por listas de tipos em [22], a intenção expressa pelos autores para interpretação de $[\tau_1, \ldots, \tau_n]$ é a interseção dos conjuntos de termos que tenham τ_i como tipo no sistema. Assim, o polimorfismo é integrado ao sistema listando os tipos assumidos pelo mesmo termo. Como exemplo, com o contexto $\{x : (\alpha \to \beta) \cap \alpha\}$, podemos inferir $A : \beta$ para a autoaplicação $A \equiv (x\ x)$.

A motivação inicial para os estudo de IT foi a de caracterização dos termos SN no λ-calculus. Esta caracterização foi provada com um sistema de IT por G. Pottinger em [81]. Pela equivalência do λ-calculus à máquina de Turing, a decisão da propriedade de SN para termos no primeiro é equivalente à decisão do problema da parada para o segundo [92]. Portanto, o problema de inferência de tipos para o sistema em [81] é indecidível. Apesar disso, esses sistemas apresentam uma alternativa conveniente para sistemas de tipos computacionais ao $\lambda 2$ (*cf.* [7] para a versão à la Curry), por tratarem o polimorfismo de maneira finitária. Além disso, sistemas de IT possuem PT, propriedade não satisfeita pelo sistema HM [103]. O polimorfismo suportado no sistema HM [28, 75], utilizado pelo Standard ML, é uma versão mais restrita dos tipos universais de $\lambda 2$.

Nos sistemas de [22, 81] os tipos com interseção têm a restrição de não ocorrerem imediatamente à direita de "$\to$". Em [24], M. Coppo *et al.* apresentaram um sistema de IT sem esta restrição e incluem o tipo ω, introduzido por Sallé em [90], como um tipo universal. Esse sistema é **relevante** [29], ou seja, uma lei de redundância não é admissível no sistema. No estudo de PT para esse sistema, os autores introduzem a **expansão** para os tipos, pois apenas a substituição e a redundância, sendo esta portanto eliminada do sistema, não são suficientes para a obtenção das tipagens possíveis de um termo. A expansão é a operação sintática que introduz uma interseção nos tipos, trocando um subtermo de um tipo por um número finito de cópias. Como exemplo, a expansão de $\alpha \to \alpha'$ para o tipo $(\alpha \to \alpha') \to \beta$ resulta em $(\alpha_1 \to \alpha'_1) \cap (\alpha_2 \to \alpha'_2) \to \beta$.

H. Barendregt *et al.* apresentaram em [9] o sistema BCD de IT, completo em relação à semântica. O sistema BCD é baseado no sistema irrestrito de [25] e com o conjunto de tipos usado em [24]. Além disso, é incluída uma relação de ordem para subtipos usada no resultado de completitude do sistema de tipos. Por sua importância em relação à apresentação do sistema de IT como uma maneira de investigação para a semântica do

λ-calculus, ele se tornou a principal referência no estudo de tais sistemas. Por ser muito flexível, a inferência de tipos é extremamente complexa (sendo realizadas por procedimentos de semidecisão, já que a inferência é indecidível). Assim, uma restrição de BCD com as mesmas propriedades é introduzido em [94, 95], onde algumas variantes de sistemas de IT são estudadas.

A propriedade de PT para sistemas de IT foi investigada em [24, 86, 94, 95], onde a ligação de PT com formas β-normais fica em evidência. A diferença entre os sistemas apresentados nestes trabalhos são as operações sintáticas associadas para a obtenção da tipagem do termo, a partir de sua PT, entre elas a expansão e suspensão (*lifting*). Um problema na formalização da expansão é definir os subtermos de um tipo que devem ser expandidos. Em [24] o conjunto de tipos a serem expandidos é formalizado, chamado de *nucleus*, de forma a manter a consistência com a tipabilidade no sistema. De fato, o nucleus corresponde a uma tipagem onde os subtermos a serem expandidos são sublinhados. A seguir, um exemplo para ilustrar a noção de nucleus. Definimos a notação de IT com o símbolo que representa a interseção com uma associação mais forte do que $\rightarrow$. Assim, denotamos *e.g.* $\alpha_1 \cap \alpha_2 \rightarrow \beta$ para representar o tipo $(\alpha_1 \cap \alpha_2) \rightarrow \beta$.

Exemplo 2.5.1: Seja $(x\ y) : \langle\{x : \alpha \rightarrow \beta,\ y : \alpha\} \vdash \beta\rangle$. Assim, $\langle\{x : \underline{\alpha \rightarrow \beta},\ y : \underline{\alpha}\} \vdash \underline{\beta}\rangle$ e $\langle\{x : \underline{\alpha} \rightarrow \beta,\ y : \underline{\alpha}\} \vdash \beta\rangle$ são dois nuclei possíveis. Ao expandirmos de acordo ao primeiro nucleus selecionado tem-se a tipagem $\langle\{x : (\alpha_1 \rightarrow \beta_1) \cap (\alpha_2 \rightarrow \beta_2),\ y : \alpha_1 \cap \alpha_2\} \vdash \beta_1 \cap \beta_2\rangle$ correpondente a derivação

$$\dfrac{\dfrac{x : \alpha_1 \rightarrow \beta_1 \qquad y : \alpha_1}{(x\ y) : \beta_1} \qquad \dfrac{x : \alpha_2 \rightarrow \beta_2 \qquad y : \alpha_2}{(x\ y) : \beta_2}}{(x\ y) : \beta_1 \cap \beta_2}$$

e o segundo nucleus a tipagem $\langle\{x : \alpha_1 \cap \alpha_2 \rightarrow \beta,\ y : \alpha_1 \cap \alpha_2\} \vdash \beta\rangle$ que corresponde a derivação

$$\dfrac{x : \alpha_1 \cap \alpha_2 \rightarrow \beta \qquad \dfrac{y : \alpha_1 \qquad y : \alpha_2}{y : \alpha_1 \cap \alpha_2}}{(x\ y) : \beta}$$

Supondo que $(x\ y)$ é o argumento de uma aplicação, a definição do conjunto de subtipos a ser expandido deve "propagar" a expansão realizada na tipagem de $(x\ y)$ de forma que o resultado também seja uma tipagem no sistema correspondente.

E. Sayag e M. Mauny apresentaram em [37], um sistema onde as PT para formas β-normais têm apenas a substituição de tipos como operação associadas. Em [38, 39] os autores propõem uma extensão do resultado para termos WN em λ, apresentando uma noção de PT que tem como operação sintática associada a substituição e uma versão de expansão simplificada em relação as definições propostas anteriormente. Apresentamos o sistema de Sayag e Mauny apresentado em [37] com mais detalhes adiante.

Em [64], J. Wells e A. Kfoury apresentaram o Sistema I de IT, onde a operação sintática associada a noção de PT é uma substituição, que também aplica expansões. No Sistema I, os tipos com interseção são acrescidos com as **variáveis de expansão**, que são utilizadas na formalização da operação de expansão como uma espécie de substituição. As PT deste sistema possuem variáveis de expansão, indicando onde uma substituição pode aplicar uma expansão. O procedimento de inferência para o Sistema I é baseado em unificação e um conjunto de passos corresponde a um passo da redução β. S. Carlier e J. Wells apresentaram em [15] o Sistema E tal que, através das variáveis de expansão, possui um procedimento de inferência baseado em unificação com uma correspondência de um para um com a redução β. Enquanto em [64] foi provado que as tipagens retornadas pelo algoritmo é principal, a noção de PT para o Sistema E não está definida.

Em relação aos cálculos de SE baseados no λ-calculus, S. Lengrand *et al.* apresentaram um sistema de IT para o λx-calculus em [69], onde o conceito de variáveis disponíveis foi utilizado para a obtenção de caracterização dos termos SN em λx[1]. Um sistema de IT também foi proposto para o $\lambda\upsilon$-calculus [46]. Em seu curso [46], J. Goubault-Larrecq apresenta um sistema de IT para $\lambda\upsilon$ [70] tal que cada termo tipável é SN mas o outro lado da implicação não é verdaderia. Porém, nenhum trabalho para cálculos de ES onde a composição de substituições é permitida foi realizado até o momento.

Em sistemas onde todo termo é trivialmente tipado com ω, este é chamado de tipo universal. A possibilidade de adicionar termos não terminantes ao conjunto de termos tipáveis possibilita a análise da propriedade de WN. Além disso, em [25] M. Coppo *et al.* apresentam uma estratificação de todo o conjunto Λ, onde todo termo solúvel tem uma tipagem no sistema introduzido. Assim, apenas termos insolúveis não são tipáveis com

[1] E. Bonelli apresentou em [12,13] um sistema de tipos com polimorfismo paramétrico, ou seja baseado no sistema F de Girard, para o λx-calculus.

um tipo considerado significativo, ou seja, que não seja equivalente ao tipo universal ω. A seguir, apresentamos um termo insolúvel e que portanto não é tipável com uma tipagem significativa em nenhum dos sistemas de IT mencionados. A não tipabilidade do termo e o problema relacionado são usados nos capítulos adiante.

Exemplo 2.5.2: Seja $R \equiv (\lambda x.(x\ x)\ \lambda x.(x\ x))$ o autorreprodutor. Este termo não tem tipagens diferente de ω em sistemas mais complexos como o BCD. No sistema de [15], com a correspondência entre o algoritmo de unificação e a redução β, o processo de unificação tem um laço infinito. Explicamos de maneira simples este problema de unificação.

Sejam as tipagens $\langle \emptyset \vdash (\alpha \to \beta) \cap \alpha \to \beta \rangle$ e $\langle \emptyset \vdash (\alpha' \to \beta') \cap \alpha' \to \beta' \rangle$ de $\lambda x.(x\ x)$. Assim, para uma tipagem de R devemos satisfazer a seguinte condição : $(\alpha \to \beta) \cap \alpha = (\alpha' \to \beta') \cap \alpha' \ \to \beta'$. Observe que temos que "igualar" um $\to$ com um $\cap$. Assim, uma expansão é aplicada em $(\alpha' \to \beta') \cap \alpha' \to \beta'$, de onde obtemos:

$$((\alpha'_1 \to \beta'_1) \cap \alpha'_1 \to \beta'_1) \cap ((\alpha'_2 \to \beta'_2) \cap \alpha'_2 \to \beta'_2)$$

O problema é então transformado em:

$$\alpha \to \beta = (\alpha'_1 \to \beta'_1) \cap \alpha'_1 \to \beta'_1, \qquad \alpha = (\alpha'_2 \to \beta'_2) \cap \alpha'_2 \to \beta'_2$$

e então a primeira igualdade se reduz ao problemas $\alpha = (\alpha'_1 \to \beta'_1) \cap \alpha'_1$ e $\beta = \beta'_1$. Portanto, temos a igualdade $(\alpha'_1 \to \beta'_1) \cap \alpha'_1 = (\alpha'_2 \to \beta'_2) \cap \alpha'_2 \to \beta'_2$, que equivale ao problema original.

2.5.1 O sistema λ^{SM_r}

E. Sayag e M. Mauny introduziram em [37] um sistema de IT para o λ-calculus, com a finalidade de investigar a propriedade de PT para formas β-normais. O conjunto de IT usado pelo sistema de Sayag e Mauny é equivalente ao usado por van Bakel em [94], onde tipos com interseção não ocorrem imediatamente a direita de "$\to$". Porém, o sistema de [94] possui uma relação de ordem para subtipos e uma regra de inferência associada. Assim, para o sistema de [94] a noção de PT tem a substituição, expansão e a suspensão como operações sintáticas associadas. Enquanto uma expansão está relacionada a regras de inferência que introduzem uma interseção, a suspensão equivale a aplicação de uma regra de inferência para a ordem de subtipos. Portanto, na tentativa em estender os resultados obtidos em [37] para qualquer termo terminante [38,39], as operação associadas a noção de PT são apenas a substituição e a expansão. Porém, no relatório técnico apresentado em [38], que contém provas omitidas em [39], uma hipótese falsa é utilizada nas provas de redução e expansão de sujeito. Na análise sobre a propriedade de SR para a versão à la de Bruijn dos sistemas de Sayag e Mauny, introduzidos na Seção 5.1, apresentamos mais detalhes sobre este problema. Assim, nos concentramos no sistema de [37], onde a única opereção sintática associada para a noção de PT para formas β-normais é a substituição.

Apresentamos nesta seção o sistema λ^{SM}, onde o axioma de designação de tipos para variáveis é apresentado em uma versão mais geral, como apresentado em [38, 39], cujo axioma do sistema em [37], que denotamos por λ^{SM_r}, é uma restrição. Em [37] os tipos com interseção são denotados por $[\tau_1, \ldots, \tau_m]$, que são tratados como multiconjuntos de tipos. Usamos a notação $\tau_1 \wedge \cdots \wedge \tau_m$ no lugar da notação original e definimos $\mathcal{U}$ como sendo o conjunto de multiconjuntos de tipos em $\mathcal{T}$ mais o elemento ω. Assim, a interseção é comutativa e associativa, onde ω, que representa o multiconjunto vazio $[\,]$, é o elemento neutro. Os elementos de $\mathcal{T}$ são denotados por τ e σ enquanto os elementos de $\mathcal{U}$ são denotados por u. Observe que a interseção $u_1 \wedge u_2$ de dois elementos de $\mathcal{U}$ equivale a união aditiva para multiconjuntos. Assim, o conjunto $\mathcal{T}$ é definido por

$$\tau := \alpha \,|\, (\tau_1 \wedge \cdots \wedge \tau_n) \rightarrow \tau$$

onde $\alpha \in \mathcal{A}$ e $n \geq 0$. O contexto de tipos A é definido com um mapeamento de elementos em $\mathcal{V}$ para os elementos de $\mathcal{U}$. Assim, o domínio de um contexto A é definido por $Dom(A) = \{x \in \mathcal{V} \,|\, A(x) \neq \omega\}$. Para um contexto A tal que $Dom(A) = \{x_1, \ldots, x_m\}$ e $\forall 1 \leq j \leq m, A(x_j) = u_j$ para $u_j \in \mathcal{U}$, temos a notação $A = \{x_1 : u_1, \ldots, x_m : u_m\}$. Além disso temos a definição de $A \setminus \{x\}$, que mantém o mapeamento original para todos os elementos exceto o x que é mapeado para ω, e a definição de $A_1 + A_2$ tal que $A_1 + A_2(x) = A_1(x) \wedge A_2(x)$.

A seguir, as definições dos sistemas λ^{SM} e λ^{SM_r}.

Definição 2.5.1 (Os sistemas $\lambda^{SM}/\lambda^{SM_r}$):

1. As regras de inferência para o sistema λ^{SM} são dadas por:

$$\frac{}{x\!:\!\langle\{x\!:\!\tau\} \vdash \tau\rangle}\text{ var} \qquad \frac{M\!:\!\langle A \vdash \tau\rangle}{\lambda x.M\!:\!\langle A \setminus \{x\} \vdash A(x) \rightarrow \tau\rangle} \rightarrow_i$$

$$\frac{M_1\!:\!\langle A_1 \vdash \omega \rightarrow \tau\rangle \qquad M_2\!:\!\langle A_2 \vdash \sigma\rangle}{(M_1\ M_2)\!:\!\langle A_1 + A_2 \vdash \tau\rangle} \rightarrow'_e$$

$$\frac{M_1\!:\!\langle A_1 \vdash \wedge_{i=1}^{n}\sigma_i \rightarrow \tau\rangle \qquad M_2\!:\!\langle A_2^1 \vdash \sigma_1\rangle \ \ldots\ M_2\!:\!\langle A_2^n \vdash \sigma_n\rangle}{(M_1\ M_2)\!:\!\langle A_1 + A_2^1 + \cdots + A_2^n \vdash \tau\rangle} \rightarrow_e$$

2. O sistema λ^{SM_r} é obtido a partir do sistema λ^{SM}, onde a regra var é substituída por:

$$x\!:\!\langle\{x\!:\!\sigma_1 \rightarrow \cdots \rightarrow \sigma_n \rightarrow \alpha\} \vdash \sigma_1 \rightarrow \cdots \rightarrow \sigma_n \rightarrow \alpha\rangle\ (n \geq 0) \quad \text{var}_r$$

A notação original para tipagens foi adaptada para a utilizada no presente trabalho. Podemos observar nas regras acima que o axioma var está restrito aos elementos de $\mathcal{T}$ e as regras $\rightarrow_e$ e $\rightarrow'_e$ para a aplicação introduzem a interseção nos contextos. Observe que a regra $\rightarrow_e$ dispensa uma regra de introdução para interseção. Assim, os elementos de $\mathcal{U}$ não podem ser atribuídos como tipos de um termo em λ.

Um algoritmo que infere tipagens em λ^{SM_r} para todas formas β-normais $Infer(N)$ é introduzido, similar a algoritmos para pares principais em [24, 86, 87, 94]. O algoritmo de inferência para a versão à la de Bruijn apresentado na Seção 6.1 é análogo ao apresentado em [37] logo o omitimos nesta seção. A seguir apresentamos um teorema estabelecendo as propriedade de $Infer$.

Teorema 2.5.2 (Propriedades de $Infer$ [37]): Seja N uma β-nf:

1. **(Correção)** Se $Infer(N) = (A, \tau)$ então $N\!:\!\langle A \vdash_{\lambda^{SM_r}} \tau\rangle$.
2. **(Completitude)** Se $N : \langle A' \vdash_{\lambda^{SM_r}} \tau'\rangle$ então $Infer(N) = (A, \tau)$ e existe uma substituição s tal que $s(A) = A'$ e $s(\tau) = \tau'$.

Assim, o item 2 acima estabelece que as tipagens retornadas por $Infer$ são principais, onde apenas a substituição é a operação sintática associada. A substituição necessária para esta demonstração é definida no início do Capítulo 6, sendo uma extensão direta dos mapeamentos de $\mathcal{A}$ em $\mathcal{T}$ para o domínio $\mathcal{U}$. Logo, essa substituição não aplica expansões. A restrição do axioma var$_r$ é essencial para que esta noção de PT seja possível em λ^{SM_r}. A seguir um contraexemplo no sistema λ^{SM}.

Exemplo 2.5.3: Temos que $Infer(x\ y) = (\{x : \alpha \to \beta,\ y : \alpha\}, \beta)$ e em λ^{SM} a tipagem $(x\ y) : \langle\{x : \alpha_1 \wedge \alpha_2 \to \beta,\ y : \alpha_1 \wedge \alpha_2\} \vdash \beta\rangle$. Assim, para qualquer substituição de tipos temos que $s(\alpha \to \beta) = s(\alpha) \to s(\beta)$, e por definição $s(\alpha), s(\beta) \in \mathcal{T}$. Portanto, a tipagem em λ^{SM} só pode ser obtida através de expansão.

Os conjuntos $\mathcal{T}_E$ e $\mathcal{T}_{NF}$ são dois subconjuntos de $\mathcal{T}$, onde σ representa os elementos de $\mathcal{T}_E$ e τ os elementos de $\mathcal{T}_{NF}$, definidos por indução mútua:

$$\begin{array}{lcll} \sigma & ::= & \alpha \mid \tau \to \sigma, & \text{onde } TV(\tau) \cap TV(\sigma) = \emptyset \\ \tau & ::= & \alpha \mid \sigma_1 \wedge \cdots \wedge \sigma_n \to \tau, & \text{para } n \geq 0 \end{array}$$

Os elementos de $\mathcal{T}_E$ são chamados de tipos de contexto e os elementos de $\mathcal{T}_{NF}$ são chamados de tipos principais. O conjunto $\mathcal{E}$ é formado pelo contexto vazio, os contextos de tipos com designações $\{x : \sigma_1 \wedge \cdots \wedge \sigma_n\}$ e fechado para a interseção de contextos. Ou seja, se $A_1, A_2 \in \mathcal{E}$ então $A_1 + A_2 \in \mathcal{E}$. Como os tipos a esquerda de $\to$ para os elementos em $\mathcal{T}_{NF}$ são interseções de elementos em $\mathcal{T}_E$, podemos definir o subconjunto $\mathcal{U}_E \subset \mathcal{U}$, tal que $\mathcal{U}_E$ é composto da interseção dos elementos em $\mathcal{T}_E$ mais o ω. Assim, os elementos a esquerda de $\to$ para os tipos em $\mathcal{T}_{NF}$ e as designações em $A \in \mathcal{E}$ são elementos de $\mathcal{U}_E$.

Com as definições acima e o conjunto $Im(Infer)$ composto por todos os pares retornados pela aplicação de $Infer$ na formas β-normais, foi provado em [37] que:

$$Im(Infer) \subseteq \mathcal{E} \times \mathcal{T}_{NF}$$

Assim, temos uma primeira descrição para PT em λ^{SM_r} de formas β-normais. Na Seção 6.2 apresentamos uma descrição análoga para o sistema $\lambda_{dB}^{SM_r}$.

A caracterização de PT depende de um série de definições sintáticas, as quais mencionamos brevemente nesta seção, pois apresentamos definições similares na Seção 6.2. Assim, ressaltamos as pequenas diferenças das definições de [37] e as utilizadas neste trabalho. Em relação às variáveis de tipo, as definições de ocorrências **positiva**, **negativa** e **final** são as usuais da literatura (*cf.* [68]), restritas aos conjuntos $\mathcal{T}_{NF}, \mathcal{T}_E$ e $\mathcal{E}$. A seguir, definimos os A-tipos.

Definição 2.5.3 (A-tipos): Os A-tipos são definidos para por:

$$\begin{array}{lcll} T & ::= & [\sigma_1, \ldots, \sigma_n] \Rightarrow \tau & \text{para } n \geq 0 \\ & \mid & [\sigma_1, \ldots, \sigma_n] \Rightarrow & \text{para } n \geq 1 \end{array}$$

Mantemos a notação de multiconjuntos para A-tipos pois esta é consistente com a definição da operação $\bar{A}$, utilizada na transformação de uma tipagem para um A-tipo. Definimos $\bar{A}$ a seguir.

Definição 2.5.4: Seja A um contexto, $\bar{A}$ é definido indutivamente por:

(i) Se $A = \{\}$, então $\bar{A} = [\,]$

(ii) Se $A = \{x\!:\!\sigma_1 \wedge \cdots \wedge \sigma_n\}$, então $\bar{A} = [\sigma_1, \ldots, \sigma_n]$.

(iii) Se $A = A_1 + A_2$, então $\bar{A} = \bar{A}_1 \uplus \bar{A}_2$.

Assim, para qualquer contexto A e tipo τ, $\bar{A} \Rightarrow \tau$ é um A-tipo. Observe que na formação de um A-tipo, o conjunto de multiconjuntos que um contexto de tipo representa neste sistema é transformado em um único multiconjunto.

Dizemos que um A-tipo T' **está contido** em T se $T = [\sigma_1, \ldots, \sigma_n] \Rightarrow \tau$ e $T' = [\sigma_{i_1}, \ldots, \sigma_{i_p}] \Rightarrow$ ou $T' = [\sigma_{i_1}, \ldots, \sigma_{i_p}] \Rightarrow \tau$, onde $[\sigma_{i_1}, \ldots, \sigma_{i_p}]$ é um submulticonjunto de $[\sigma_1, \ldots, \sigma_n]$, diferente de vazio no primeiro caso. Se $T' \neq T$ dizemos que T' está **estritamente contido** em T.

Algumas definições em [37] são apresentas para tipagens e depois estendidas para A-tipos. Como exemplo, a polaridade para ocorrência de variáveis de tipo, positiva ou negativa, é estendida de maneira direta trocando o sinal de ocorrência a esquerda de $\Rightarrow$ e mantendo o sinal para elementos do multiconjunto. Outras definições nós apresentamos com a respectiva extensão para A-tipos incluída. A seguir, a definição de subtermos a esquerda para um A-tipo.

Definição 2.5.5 (Subtermos a esquerda): Seja $T = [\sigma_1, \ldots, \sigma_n] \Rightarrow \tau$ um A-tipo. O **conjunto de subtermos a esquerda de** T, denotado por $E(T)$, é definido por:

(i) $E([\sigma_1, \ldots, \sigma_n] \Rightarrow) = \{\sigma_1, \ldots, \sigma_n\}$.

(ii) $E([\sigma_1, \ldots, \sigma_n] \Rightarrow \tau) = \{\sigma_1, \ldots, \sigma_n\} \cup E(\tau)$.

(iii) $E(\sigma_1 \wedge \cdots \wedge \sigma_n \rightarrow \tau) = \{\sigma_1, \ldots, \sigma_n\} \cup E(\tau)$ se $n \neq 0$ e $E(\tau)$ caso contrário.

(iv) $E(\alpha) = \emptyset$.

A definição acima quando restrita aos conjunto $\mathcal{T}_E$ e $\mathcal{T}_{NF}$ corresponde a definição de $G(T)$ extendida para A-tipos em [37]. A seguir as definições para o fechamento de A-tipos.

Definição 2.5.6 (Fechamento para A-tipos):

1. Dizemos que um A-tipo T é **fechado** se para toda variável de tipos ocorrendo em $TV(T)$ existe exatamente uma ocorrência positiva e uma negativa em T.
2. Um A-tipo $T = [\sigma_1, \ldots, \sigma_n] \Rightarrow \tau$ é **fechado ao final** se a ocorrência final de τ é também uma ocorrência final em um tipo de $E(T)$.
3. Um A-tipo T é **fechado minimal** se não existe um A-tipo fechado estritamente contido em T.

A partir das definições de fechamento apresentadas acima temos a definição a seguir.

Definição 2.5.7 (A-tipo completo): Um A-tipo é **completo** se este é fechado, fechado ao final e fechado minimal.

Em [37] está demonstrado que para todo par $(A, \tau) \in Im(Infer)$, $\bar{A} \Rightarrow \tau$ é um A-tipo completo. A partir da restrição na formação de A-tipos completo temos os A-tipos principais, definidos como segue.

Definição 2.5.8 (A-tipo principal): Um A-tipo completo T é **principal** se:

(i) $T = [\alpha] \Rightarrow \alpha$

(ii) $T = [\sigma_1, \ldots, \sigma_n] \Rightarrow \alpha$ e existe a partição $\uplus_{j=1}^{p} C_j = [\sigma_1, \ldots, \sigma_{i-1}, \sigma_{i+1}, \ldots, \sigma_n]$ para $\sigma_i = [\tau_1] \to \cdots \to [\tau_p] \to \alpha$ tal que $\forall 1 \leq j \leq p$, $C_j \Rightarrow \tau_j$ é principal.

(iii) $T = [\sigma_1, \ldots, \sigma_n] \Rightarrow [\sigma'_1, \ldots, \sigma'_p] \to \tau'$ e $[\sigma_1, \ldots, \sigma_n, \sigma'_1, \ldots, \sigma'_p] \Rightarrow \tau'$ é principal.

Assim, definindo o conjunto $\mathcal{P} = \{(A, \tau) \in \mathcal{E} \times \mathcal{T}_{NF} \mid \bar{A} \Rightarrow \tau \text{ é principal}\}$ temos em [37] que $Im(Infer) \subseteq \mathcal{P}$. Ou seja, as tipagens principais para β-nf retornadas por $Infer$ estão contidas nesta caracterização sintática para A-tipos principais.

O algoritmo de reconstrução $R(A, \tau)$ é apresentado, e provado que ao ser aplicado em elementos $(A, \tau) \in \mathcal{P}$ sempre retorna uma forma β-normal N tal que $Infer(N) = (A, \tau)$.

Portanto, $\mathcal{P} \subseteq Im(Infer)$. Assim, temo em [37] que:

$$Im(Infer) = \mathcal{P}$$

Portanto, o conjunto $\mathcal{P}$ é uma caracterização sintática de PT para β-nf no sistema λ^{SM_r}.

2.5.2 O sistema $\lambda^{\sqcap}$

F. Kamareddine e K. Nour introduziram em [56] um sistema para o estudo da interpretação de IT, o que é feito através da semântica de realização. Na semântica de realização (*realisability semantics*), os tipos são interpretados por conjuntos de termos. Nesse sistema, o ω é um tipo universal e uma relação binária é introduzida para os tipos em $\mathbb{U}$, de forma a obter completitude para a semântica proposta. A interseção, representada por $\sqcap$ é idempotente, também chamada de não linear. Além disso, com um subconjunto $\mathbb{U}^+$ dos tipos com interseção, os chamados tipos positivos, a tipabilidade e a realização coincidem. Ou seja, um termo M é tipável com $u \in \mathbb{U}^+$ em $\lambda^{\sqcap}$ se, e somente se, M está no conjunto de interpretação de u.

Apresentamos nesta seção algumas definições, lemas e teoremas relacionados a prova de SR para este sistema em [56]. Assim como o feito para o sistema λ^{SM} na seção anterior, o objetivo da apresentação é o de permitir a comparação com o trabalho feito para a versão à la de Bruijn, o sistema $\lambda^{\sqcap}_{dB}$, apresentado na Seção 5.2. Além do Teorema de SR, fazemos um breve comentário sobre os resultados apresentados no artigo de Kamareddine e Nour. A seguir, apresentamos o conjunto de tipos e os contextos utilizados pelo sistema $\lambda^{\sqcap}$.

Definição 2.5.9: 1. Seja $\mathcal{A}$ o conjunto de variáveis de tipos. Os tipos com interseção do sistema $\lambda^{\sqcap}$ são definidos por:

$$\tau, \sigma \in \mathbb{T} ::= \mathcal{A} \,|\, \mathbb{U} \rightarrow \mathbb{T} \qquad\qquad u, v, w \in \mathbb{U} ::= \omega \,|\, \mathbb{U} \sqcap \mathbb{U} \,|\, \mathbb{T}$$

Os tipos são considerados módulo comutatividade ($u_1 \sqcap u_2 = u_2 \sqcap u_1$), associatividade ($(u_1 \sqcap u_2) \sqcap u_3 = u_1 \sqcap (u_2 \sqcap u_3)$), idempotência ($u \sqcap u = u$) e ω o elemento neutro ($\omega \sqcap u = u$) de $\sqcap$.

Os conjunto descritos acima são semelhantes aos conjuntos descritos para o sistema λ^{SM}, com a diferença da idempotência para $\sqcap$.

Definição 2.5.10: 1. Um contexto de tipo $\Gamma = \{x_1 : u_1, \ldots, x_n : u_n\}$ é um conjunto de designações em $\mathbb{U}$, denotado por $(x_i : u_i)_n$. Assim, defini-se $dom(\Gamma) = \{x_1, \ldots, x_n\}$.

2. O contexto vazio é denotado por $()$.

3. Se $FV(M) = \{x_1, \ldots, x_n\}$, então env_ω^M denota o contexto $(x_i : \omega)_n$.

4. Seja $\Gamma_1 = (x_i : u_i)_n, (y_j : v_j)_m$ e $\Gamma_2 = (x_i : u'_i)_n, (z_k : w_k)_l$. Defini-se a interseção para contextos por $\Gamma_1 \sqcap \Gamma_2 = (x_i : u_i \sqcap u'_i)_n, (y_j : v_j)_m, (z_k : w_k)_l$.

A seguir, apresentamos as regras de inferência para o sistema $\lambda^{\sqcap}$.

Definição 2.5.11 (O sistema $\lambda^{\sqcap}$): As regras de tipagem do sistema $\lambda^{\sqcap}$ são dadas por:

$$\frac{}{x : \langle (x:\tau) \vdash \tau \rangle}\ \text{var} \qquad \frac{M : \langle \Gamma \vdash \tau \rangle \quad x \notin dom(\Gamma)}{\lambda x.M : \langle \Gamma \vdash \omega \to \tau \rangle}\ \to'_i$$

$$\frac{}{M : \langle env_\omega^M \vdash \omega \rangle}\ \omega \qquad \frac{M : \langle \Gamma, x : u \vdash \tau \rangle}{\lambda x.M : \langle \Gamma \vdash u \to \tau \rangle}\ \to_i$$

$$\frac{M : \langle \Gamma \vdash u_1 \rangle \quad M : \langle \Gamma \vdash u_2 \rangle}{M : \langle \Gamma \vdash u_1 \sqcap u_2 \rangle}\ \sqcap_i \qquad \frac{M_1 : \langle \Gamma \vdash u \to \tau \rangle \quad M_2 : \langle \Gamma' \vdash u \rangle}{(M_1\ M_2) : \langle \Gamma \sqcap \Gamma' \vdash \tau \rangle}\ \to_e$$

$$\frac{M : \langle \Gamma \vdash u \rangle \quad \langle \Gamma \vdash u \rangle \leqslant \langle \Gamma' \vdash u' \rangle}{M : \langle \Gamma' \vdash u' \rangle}\ \leqslant$$

onde a relação binária $\leqslant$ é definida por:

$$\frac{}{\Phi \leqslant \Phi}\ \text{ref} \qquad \frac{\Phi_1 \leqslant \Phi_2 \quad \Phi_2 \leqslant \Phi_3}{\Phi_1 \leqslant \Phi_3}\ \text{tr}$$

$$\frac{}{u_1 \sqcap u_2 \leqslant u_1}\ \sqcap_e \qquad \frac{u_1 \leqslant v_1 \quad u_2 \leqslant v_2}{u_1 \sqcap u_2 \leqslant v_1 \sqcap v_2}\ \sqcap$$

$$\frac{u_2 \leqslant u_1 \quad \tau_1 \leqslant \tau_2}{u_1 \to \tau_1 \leqslant u_2 \to \tau_2}\ \to \qquad \frac{u_1 \leqslant u_2 \quad x \notin dom(\Gamma)}{\Gamma, x : u_1 \leqslant \Gamma, x : u_2}\ \leqslant_c$$

$$\frac{u_1 \leqslant u_2 \quad \Gamma' \leqslant \Gamma}{\langle \Gamma \vdash u_1 \rangle \leqslant \langle \Gamma' \vdash u_2 \rangle}\ \leqslant_{\langle\rangle}$$

Φ, Φ', $\Phi_1, \ldots$ são usados para denotar $u \in \mathcal{U}'$, contextos Γ ou tipagens $\langle \Gamma \vdash u \rangle$. Observe que em $\Phi \leqslant \Phi'$, Φ e Φ' pertencem a mesma classe de objetos.

Como podemos observar, a interseção é um construtor mais geral de tipos do que $\wedge$ em λ^{SM}, pois elementos em $\mathbb{U}$ podem ser atribuídos como tipos para os termos, enquanto a eliminação de $\sqcap$ é feita através da relação $\leqslant$. Porém, observe que as regras $\to_i$ e $\to_i'$ estão restrita a termos com tipos em $\mathbb{T}$. As definições das regras var, $\to_i'$ e ω garantem a relevância do sistema, como estabelecida no lema a seguir.

Lema 2.5.12 (Relevância para $\lambda^{\sqcap}$ [56]): Sejam M um termo e Γ um contexto em $\lambda^{\sqcap}$:

1. Se $M:\langle\Gamma \vdash_{\lambda^{\sqcap}} u\rangle$ então $dom(\Gamma) = FV(M)$.

2. Se $dom(\Gamma) = FV(M)$ então $M:\langle\Gamma \vdash_{\lambda^{\sqcap}} \omega\rangle$.

Apesar do axioma var estar restrito a tipos em $\mathbb{T}$ e que os contextos na regra $\sqcap_i$ devem coincidir, o lema a seguir mostra que uma outra versão de cada regra é admissível.

Lema 2.5.13 (Regras admissíveis em $\lambda^{\sqcap}$ [56]):

1. A regra $\sqcap_i'$ é admissível em $\lambda^{\sqcap}$.

$$\frac{M:\langle\Gamma_1 \vdash u_1\rangle \qquad M:\langle\Gamma_2 \vdash u_2\rangle}{M:\langle\Gamma_1 \sqcap \Gamma_2 \vdash u_1 \sqcap u_2\rangle}\ \sqcap_i'$$

2. A regra var$'$ é admissível em $\lambda^{\sqcap}$.

$$\frac{}{x:\langle(x:u) \vdash u\rangle}\ \text{var}'$$

A redução de sujeito é demonstrada de maneira padrão, onde um lema de geração e um lema de substituição são apresentados e então a propriedade é provada. A seguir apresentamos os lemas preliminares, de forma que possam ser comparados aos lemas da versão à la de Bruijn apresentada na Seção 5.2.

Lema 2.5.14 (Geração para $\lambda^{\sqcap}$ [56]):

1. Se $x:\langle\Gamma \vdash u\rangle$, então $\Gamma = (x:v)$ onde $v \leqslant u$.

2. Se $(M\ x) : \langle\Gamma, x:u \vdash v\rangle$ e $x \notin FV(M)$, então $v = \omega$ ou $v = \sqcap_{i=1}^{k}\tau_i$ onde $k \geq 1$ e $\forall 1 \leq i \leq k$, $M:\langle\Gamma \vdash u \to \tau_i\rangle$.

3. Se $\lambda x.M : \langle \Gamma \vdash u \rangle$ e $x \in FV(M)$, então $u = \omega$ ou $u = \sqcap_{i=1}^{k}(v_i \to \tau_i)$ onde $k \geq 1$ e $\forall 1 \leq i \leq k,\ M : \langle \Gamma, x : v_i \vdash \tau_i \rangle$.

4. Se $\lambda x.M : \langle \Gamma \vdash u \rangle$ e $x \notin FV(M)$, então $u = \omega$ ou $u = \sqcap_{i=1}^{k}(v_i \to \tau_i)$ onde $k \geq 1$ e $\forall 1 \leq i \leq k,\ M : \langle \Gamma \vdash \tau_i \rangle$.

Lema 2.5.15 (Substituição para $\lambda^{\sqcap}$ [56]): Se $M : \langle \Gamma, x : u \vdash v \rangle$ e $N : \langle \Delta \vdash u \rangle$, então $[x/N]M : \langle \Gamma \sqcap \Delta \vdash v \rangle$.

Observe que pela relevância do sistema no lema acima temos que $x \in FV(M)$. Para o caso $x \notin FV(M)$, o termo N é eliminado no termo $[x/N]M$. Assim, para SR em um sistema relevante precisamos da noção de restrição, introduzida a seguir como em [56].

Definição 2.5.16: Se Γ é um contexto e $C \subseteq dom(\Gamma)$, então $\Gamma\restriction_C$ denota a restrição de Γ a C. Se $C = FV(M)$, então denota-se $\Gamma\restriction_M$ no lugar de $\Gamma\restriction_{FV(M)}$.

Com a noção acima podemos apresentar o teorema de SR para o sistema $\lambda^{\sqcap}$.

Teorema 2.5.17 (SR para redução β [56]): Se $M : \langle \Gamma \vdash_{\lambda^{\sqcap}} \tau \rangle$ e $M \to_{\beta} N$, então $N : \langle \Gamma\restriction_N \vdash_{\lambda^{\sqcap}} \tau \rangle$.

Portanto, obtemos a propriedade SR para $\lambda^{\sqcap}$, considerando que a informação do contexto de tipo do argumento de uma redução β nula poderá ser eliminada. Para a propriedade de expansão de sujeito em [56] apresenta-se um conceito similar, de alargamento de um contexto de tipos.

A semântica de realização é apresentada para os tipos em $\mathbb{U}$, a partir da saturação de duas relações diferentes, a dizer a redução β e a redução fraca (*weak head reduction*)[2]. Na semântica proposta, uma variável de tipo é interpretada por um conjunto de termos em λ saturados na relação de redução correspondente e então os tipos funcionais e com interseção são interpretados de maneira intuitiva. Ou seja, a interpretação da interseção é a interseção dos conjuntos de interpretação e um tipos funcioanal $\sigma \to \tau$ é interpretado por termos que são funções de σ em τ. Contudo, a interpretação para os tipos de $\mathbb{U}$ em ambas coincide. A seguir definimos os tipos positivos e negativos.

Definição 2.5.18 (IT positivos e negativos): O conjuntos $\mathbb{U}^+$ e $\mathbb{U}^-$ são definidos

[2] $M \to N$ se $M \equiv (\lambda x.P\ Q\ Q_1 \cdots Q_n)$ e $N \equiv [x/Q]P\ Q_1 \cdots Q_n$

indutivamente por:

(i) $\forall \alpha \in \mathcal{A}$, $\alpha \in \mathbb{U}^+$ e $\alpha \in \mathbb{U}^-$

(ii) $\omega \in \mathbb{U}^-$.

(iii) Se $u \in \mathbb{U}^+$ então $u \wedge v \in \mathbb{U}^+$.

(iv) Se $u, v \in \mathbb{U}^-$ então $u \wedge v \in \mathbb{U}^-$.

(v) Se $u \in \mathbb{U}^-$ e $\tau \in \mathbb{U}^+$ então $u \rightarrow \tau \in \mathbb{U}^+$.

(vi) Se $u \in \mathbb{U}^+$ e $\tau \in \mathbb{U}^-$ então $u \rightarrow \tau \in \mathbb{U}^-$.

O conjunto descrito acima são subconjuntos de $\mathbb{U}$ que caracterizam tipos com ocorrências de ω em posições específicas. Assim, em [56] foi demonstrado que todo termo tipável com tipo em $\mathbb{U}^+$ é terminante e, além disso, que um termo M é tipável com $u \in \mathbb{U}^+$ se, e somente se, M está no conjunto de termos da interpretação de u. De fato, $\beta(M) \in Termos_{\lambda^{\sqcap}}(\Theta)$, onde $\Theta = \langle () \vdash u \rangle$.

F. Kamareddine *et al.* apresentaram em [57] uma semântica para um sistema que inclui variáveis de expansão. Nesse sistema, o conjunto $\mathcal{E}$ de variáveis de expansão é acrescentado a formação dos tipos, bem como uma regra para a relação binária e uma regra de inferência apropriadas.

Capítulo 3

Cálculos à la de Bruijn e sistemas de tipos simples

Apresentamos neste capítulo o λ-calculus à la de Bruijn e dois cálculos de substituições explícitas baseados nessa notação, o λs_e-calculus e o $\lambda\sigma$-calculus. As seções são compostas pela apresentação dos cálculos e suas propriedades, seguidas da versão com tipos simples onde as respectivas propriedades são também apresentadas.

Os sistemas de tipos apresentados para λs_e e $\lambda\sigma$ são versões à la Curry, ao invés das usuais versões à la Church. Propriedades como confluência, terminação do cálculo de substituição associado e simulação para redução β são propriedades dos cálculos sem tipos logo herdadas de maneira trivial nessa versão com tipos implícitos. Por outro lado, a propriedade de redução de sujeito para os cálculos é provada através de uma adaptação das provas para a versão tipada.

3.1 O λ-calculus à la de Bruijn

O matemático holandês N.G. de Bruijn inicia na década de 60 um projeto para a verificação automática de provas formais em matemática, chamado AUTOMATH [77]. O livro *Grundlagen der Analysis* de Edmund Landau foi inteiramente especificado e checado nessa linguagem [77, 96]. Considerado o precursor de assistentes de prova modernos, e.g. Coq [10], Nuprl [20] e Isabelle/HOL [78], de Bruijn usa em seu sistema conceitos como substituições explícitas, com a formalização do cálculo de ES utilizado pelo sistema apresentado em [31], e o isomorfismo de Curry-Howard, também chamado de isomorfismo de Curry-Howard-de Bruijn [54]. N.G. de Bruijn apresenta em [30] uma notação para termos em λ onde os nomes das variáveis são substituídos por índices. Assim, termos α-equivalentes em Λ têm um único representante no conjunto de termos desta notação, o conjunto Λ_{dB}. Como exemplo, a classe das funções identidade $\lambda x.x$ é representada unicamente pelo termo $\lambda.\underline{1}$. Essa característica é conveniente computacionalmente pois, entre outras vantagens, permite que sistemas baseados no λ-calculus sejam especificados como sistemas de reescrita de primeira ordem. Em [53] são apresentadas as contribuições de N. de Bruijn para a Teoria da Computação, frutos do sistema AUTOMATH, fazendo uma análise de sua influência em conceitos usados atualmente.

Nessa seção apresentamos o λ_{dB}-calculus, o λ-calculus à la de Bruijn, em suas versões sem tipos e com um sistema de tipos simples, seguidos das respectivas propriedades.

3.1.1 O λ_{dB}-calculus

Apresentamos nessa seção o **$\boldsymbol{\lambda_{dB}}$-calculus**, o λ-calculus com índices de de Bruijn, e algumas de suas propriedades.

Definição 3.1.1: O **conjunto de termos em $\boldsymbol{\lambda_{dB}}$**, denotado por Λ_{dB}, é definido indutivamente para $n \in \mathbb{N}^*$ por:

$$M, N \in \Lambda_{dB} \quad ::= \quad \underline{n} \,|\, (M\ N) \,|\, \lambda.M$$

O n-ésimo índice refere-se à variável ligada pelo n-ésimo abstrator. Por exemplo, o termo $\lambda x.\lambda y.(x\ y)$ torna-se $\lambda.\lambda.(\,\underline{2}\ \underline{1}\,)$. Os índices dentro do n-ésimo abstrator com valor absoluto maior do que n são chamados de **índices livres** e correspondem a noção de variáveis livres no λ-calculus. Na transformação de um termo em Λ para essa notação, o valor do índice livre é calculado em relação a uma sequência finita de nomes, chamado de **referencial**. Por exemplo, o termo $(\lambda y.\lambda x.(xzy)\ y)$, calculado em relação a (x, y, z), é $(\lambda.\lambda.(\,\underline{1}\ \underline{5}\ \underline{2}\,)\ \underline{2})$. Note que no termo $\lambda.M$, o abstrator está ligando as ocorrências livres de $\underline{1}$ em M.

A simplicidade da descrição da estrutura de um termo com índices de de Bruijn, apresentada na Definição 3.1.1, facilita o uso da técnica de indução na estrutura dos termos. Os termos da forma $\underline{n}$ servem como base da indução. A hipótese de indução é então aplicada nos subtermos que compõem os termos da forma $(M\ N)$ e $\lambda.M$, onde $M, N \in \Lambda_{dB}$. Um conceito importante referente à estrutura dos termos é a noção de profundidade de um subtermo. Dizemos que um subtermo M_1 de um termo M está na **profundidade** n **de** M se o menor índice livre possível em M_1 é maior do que n. Ou seja, M_1 está dentro de n abstratores. Portanto, para $i \in \mathbb{N}^*$ dizemos que $\underline{i}$ **ocorre livre** em um termo M se há ocorrências de $\underline{i{+}n}$ na profundidade n de M.

A definição da contração β no λ_{dB}-calculus precisa de um mecanismo que detecte e atualize índices livres em termos. A seguir, apresentamos um operador similar ao apresentado em [3].

Definição 3.1.2: Sejam $M \in \Lambda_{dB}$ e $i \in \mathbb{N}$. A ***i*-elevação** de M, denotada por M^{+i}, é definida indutivamente por:

(i) $(M_1\, M_2)^{+i} = (M_1^{+i}\, M_2^{+i})$

(ii) $(\lambda.M_1)^{+i} = \lambda.M_1^{+(i+1)}$

(iii) $\underline{n}^{+i} = \begin{cases} \underline{n+1}\,, & \text{se } n > i \\ \underline{n}\,, & \text{se } n \leq i. \end{cases}$

A **elevação** de um termo M é a sua 0-elevação, denotada por M^{+}. Intuitivamente, a elevação de M corresponde a incrementar em 1 os índices livres de M. Com esse mecanismo de atualização para índices livres, é possível então apresentar uma definição da substituição usada na contração β, análoga à introduzida em [3].

Definição 3.1.3: Sejam $m, n \in \mathbb{N}^*$. A **substituição $\boldsymbol{\beta}$** para as ocorrências livres de $\underline{n}$ em $M \in \Lambda_{dB}$ pelo termo N, denotada por $\{\,\underline{n}\,/N\}M$, é definida indutivamente por:

(i) $\{\,\underline{n}\,/N\}(M_1\ M_2) = (\{\,\underline{n}\,/N\}M_1\ \{\,\underline{n}\,/N\}M_2)$

(ii) $\{\,\underline{n}\,/N\}(\lambda.M_1) = \lambda.\{\,\underline{n{+}1}\,/N^{+}\}M_1$

(iii) $\{\,\underline{n}\,/N\}\,\underline{m} = \begin{cases} \underline{m-1}\,, & \text{se } m > n \\ N, & \text{se } m = n \\ \underline{m}\,, & \text{se } m < n \end{cases}$

Observe que no item (ii) da Definição 3.1.3, o operador de elevação é usado para evitar a captura de índices livres em N. Apesar da definição formal, a substituição β é uma

metaoperação no λ_{dB}. Ou seja, $\{\,\underline{i}\,/N\}M$ denota o resultado da aplicação em M da substituição $\{\,\underline{i}\,/N\}$ como na Definição 3.1.3, onde N compõe o **corpo da substituição**.

Apresentamos a contração β a seguir, como definida em [3].

Definição 3.1.4: A **contração $\boldsymbol{\beta}$** em λ_{dB} é definida por:

$$(\lambda.M\,N) \to_\beta \{\underline{1}\,/N\}M.$$

Note que o item (iii) da Definição 3.1.3 é o mecanismo que executa a substituição e atualiza os índices livres em M, como consequência da eliminação do abstrator correspondente ao β-redex. A redução β é definida então como a contração β compatível com a estrutura dos termos em λ_{dB}.

Definição 3.1.5: A **redução $\boldsymbol{\beta}$** em λ_{dB} é definida por:

$$\frac{}{(\lambda.M\,N) \to_\beta \{\,\underline{1}\,/N\}M} \qquad \frac{M \to_\beta N}{\lambda.M \to_\beta \lambda.N}$$

$$\frac{M_1 \to_\beta N_1}{(M_1\; M_2) \to_\beta (N_1\; M_2)} \qquad \frac{M_2 \to_\beta N_2}{(M_1\; M_2) \to_\beta (M_1\; N_2)}$$

Um termo é uma **forma $\boldsymbol{\beta}$-normal**, ou simplesmente β-nf, se não há nenhuma redução β possível. A seguir um lema descrevendo as β-nfs.

Lema 3.1.6: Um termo $N \in \Lambda_{dB}$ é uma β-nf se, e somente se, N é um dos seguintes:

- $N \equiv \underline{n}$, para qualquer $n \in \mathbb{N}^*$.

- $N \equiv \lambda.N'$ e N' é uma β-nf.

- $N \equiv (\,\underline{n}\,N_1 \cdots N_m)$, para algum $n \in \mathbb{N}^*$ e $\forall 1 \leq j \leq m$, N_j é uma β-nf.

Demonstração. A prova de *necessidade* é direta, a partir da definição de β-nf. Agora, suponha que N é uma β-nf. A prova de *suficiência* é por indução na estrutura de $N \in \Lambda_{dB}$:

- Se $N \equiv \underline{n}$ então N é uma β-nf.

- Seja $N \equiv \lambda.N'$. Se N' não é uma β-nf então, pela definição de β-nf, N não é uma β-nf. Assim, N' é uma β-nf.

- Seja $N \equiv (N_1\ N_2)$. Tem-se por hipótese que N é uma β-nf, logo ambos N_1 e N_2 são β-nfs. Assim, por HI, $N_1 \equiv \lambda.N'$, para N' uma β-nf, ou $\underline{n}\, N'_1 \cdots N'_m$ para $m \geq 0$ e $\forall 1 \leq j \leq m$, N'_j é uma β-nf. Se $N_1 \equiv \lambda.N'$ então N seria um β-redex. Portanto, $N_1 \equiv \underline{n}\, N'_1 \cdots N'_m$ e $N \equiv (\underline{n}\, N'_1 \cdots N'_m\ N_2)$. □

Em [72], Mauny apresenta um isomorfismo entre o λ_{dB}-calculus e o λ-calculus. Assim, a propriedade de confluência da redução β para o λ_{dB}-calculus segue da confluência da redução β no λ-calculus (*cf.* [6] Teorema 3.2.8) e o fato de serem isomórficos. Para uma demostração de CR para o λ_{dB}-calculus sem o uso do isomorfismo, veja [83].

Contudo, precisamos investigar detalhadamente algumas propriedades sintáticas do λ_{dB} como, *e.g.* o comportamento do conjunto de índices livres após contrações/reduções β, a fim de investigar a relação destas com os sistemas de atribuição de tipos.

Como já mencionado, os índices livres correspondem a noção de variáveis livres do λ-calculus. Assim, a exemplo do que acontece com λ e as variáveis livres, para sabermos se uma substituição β será nula basta saber quais são os índices livres do termo ao qual esta é aplicada. Ao contrário do que ocorre no λ-calculus em uma contração β nula, a atualização de índices faz com que o termo resultante seja sintaticamente diferente do termo original. Para explicar o problema usamos o conceito mais geral de substituição nula. Seja M^{-i} a operação de ***i*-declinação** similar à apresentada em [83], definida apenas quando M não tem ocorrências livres de $\underline{i}$. Intuitivamente, a operação de i-declinação representa o inverso da i-elevação onde índices livres menores que i permanecem inalterados e os maiores que i são decrementados em 1. Assim, para uma substituição nula podemos provar que $\{\underline{i}/N\}M = M^{-i}$, onde $\underline{i}$ não ocorre livre em M. Portanto, para um termo M sem ocorrências livres de $\underline{1}$ temos que $(\lambda.M\ N) \to_\beta M^{-1}$.

Apresentamos uma definição formal para o conjunto dos índices livres de um termo, similar ao introduzido em [61].

Definição 3.1.7: Seja $M \in \Lambda_{dB}$:

1. O **conjunto de índices livres** de M, denotado por $FI(M)$, é definido indutivamente por:

   (i) $FI(\underline{n}) = \{\underline{n}\}$

(ii) $FI(\lambda.M) = FI(M)\backslash 1$

(iii) $FI(M_1\ M_2) = FI(M_1) \cup FI(M_2)$

2. Um termo M é dito **fechado** se $FI(M) = \emptyset$.

3. Seja o índice livre de M com o maior valor absoluto, denotado por $sup(M)$, definido por $sup(M){=}max(FI(M))$ onde $max(\emptyset){=}0$.

Observe que o conjunto de índices livres é isomorfo a um subconjunto de $\mathbb{N}^*$ logo podemos usar a Definição 2.1.2 e a Proposição 2.1.3 com algum abuso de notação.

Para sabermos o número de ocorrências de cada índice livre definimos o multiconjunto associado ao conjunto de índices livres.

Definição 3.1.8: Seja $\overline{FI}(M)$ o **multiconjunto de índices livres** de $M \in \Lambda_{dB}$ definido como acima, trocando a união de conjuntos pela união aditiva:

$$\overline{FI}(M_1\ M_2) = \overline{FI}(M_1) \uplus \overline{FI}(M_2)$$

Apresentamos no restante desta seção as relações entre o conjunto, e multiconjunto, de índices de um termo e do termo resultante após uma i-elevação ou aplicação da substituição β. A seguir, apresentamos alguns lemas auxiliares.

Lema 3.1.9: 1. $\underline{n} \in FI(\lambda.M)$ se, e somente se, $\underline{n{+}1} \in FI(M)$.

2. $\underline{n} \in^m \overline{FI}(\lambda.M)$ se, e somente se, $\underline{n{+}1} \in^m \overline{FI}(M)$.

3. $\underline{n} \in FI(M)$ se, e somente se, $\underline{n} \in^m \overline{FI}(M)$ para $m{>}0$.

Demonstração. 1. Pela Definição 2.1.2.

2. Pela extensão da Definição 2.1.2 para multiconjuntos.

3. Por indução na estrutura de M, com a definição de $\uplus$ e o Lema 3.1.9.1. □

O lema a seguir apresenta as relações de sup e a estrutura de termos em Λ_{dB}.

Lema 3.1.10: 1. $sup(M_1\ M_2) = max(sup(M_1), sup(M_2))$.

2. Se $sup(M)=0$, então $sup(\lambda.M)=0$. Caso contrário, $sup(\lambda.M)=sup(M)-1$.

Demonstração. 1. Se $sup(M_1\ M_2)=0$, nada há provar. Senão, $sup(M_1\ M_2)=n$, onde $\underline{n}=max(FI(M_1\ M_2))=max(FI(M_1)\cup FI(M_2))$. Suponha que $FI(M_1)$,$FI(M_2)\neq\emptyset$ logo $n=max(max(FI(M_1)), max(FI(M_2)))=max(sup(M_1), sup(M_2))$. Suponha que $FI(M_1)=\emptyset$. Observe que $FI(M_1)\cup FI(M_2)=FI(M_2)$ e que $max(FI(M_2))=max(max(FI(M_2)), 0)$.

2. Se $sup(M)=0$, então $FI(\lambda.M)=FI(M)=\emptyset$ logo $sup(\lambda.M)=0$. Suponha que $sup(M)=m>0$. Assim, $\underline{m}=max(FI(M))$. Se $m=1$ então $FI(M)=\{\underline{1}\}$ logo $FI(\lambda.M)=\emptyset$ e $sup(\lambda.M)=0$. Senão, $FI(\lambda.M)=\{\underline{n-1} \mid \underline{n}\in FI(M), n>1\}$. Portanto, $\underline{m-1}=max(FI(\lambda.M))$. □

O lema a seguir apresenta uma relação geral entre a i-elevação e o conjunto de índices livres de um termo.

Lema 3.1.11: 1. Se $i\geq sup(M)$, então $M^{+i}\equiv M$.

2. $FI(M^{+i})=FI(M)_{\leq i}\cup(FI(M)_{>i}+1)$

3. $\overline{FI}(M^{+i})=\overline{FI}(M)_{\leq i}\uplus(\overline{FI}(M)_{>i}+1)$

4. Se $sup(M)>i$, então $sup(M^{+i})=sup(M)+1$.

5. Se $sup(M)\leq i$, então $sup(M^{+i})=sup(M)$.

Demonstração. 1, 2 e 3: Por indução na estrutura de M, com a Proposição 2.1.3 e a observação de que $C_{>k+1}=(C_{>k+1})_{>1}$.
4: Se $sup(M)=m$, então $\underline{m}=max(FI(M))$. Tem-se que $m>i$ logo, pelo Lema 3.1.11.2, $\underline{m+1}\in FI(M^{+i})$ e $\forall\,\underline{j}\in FI(M^{+i})$, $j=n$ ou $j=n+1$ onde $\underline{n}\in FI(M)$. Assim, $m+1\geq j, \forall\,\underline{j}\in FI(M^{+i})$.
5: Pelo item 1 acima, $M^{+i}\equiv M$ logo $sup(M^{+i})=sup(M)$. □

O lema a seguir descreve o conjunto de índices livres de $\{\underline{i}/N\}M$ em função dos conjuntos de índices livres de M e N.

Lema 3.1.12: 1. Se $\underline{i} \notin FI(M)$, então $FI(\{\underline{i}/N\}M) = FI(M)_{<i} \cup (FI(M)_{>i} \backslash 1)$ e $\overline{FI}(\{\underline{i}/N\}M) = \overline{FI}(M)_{<i} \uplus (\overline{FI}(M)_{>i} \backslash 1)$

2. Se $\underline{i} \in FI(M)$, $FI(\{\underline{i}/N\}M) = FI(N) \cup FI(M)_{<i} \cup (FI(M)_{>i} \backslash 1)$ e, para $\underline{i} \in^n \overline{FI}(M)$, $\overline{FI}(\{\underline{i}/N\}M) = \overline{FI}(N)^n \uplus \overline{FI}(M)_{<i} \uplus (\overline{FI}(M)_{>i} \backslash 1)$.

3. Se $i > sup(M)$, então $\{\underline{i}/N\}M \equiv M$.

Demonstração. Por indução na estrutura de M, com os Lemas 3.1.9.1 e 3.1.9.3 e a Proposição 2.1.3. □

Note que, se $FI(M) = \{\underline{i}\}$ então $FI(M)_{<i} = \emptyset$ e $FI(M)_{>i} \backslash 1 = \emptyset$ logo $FI(\{\underline{i}/N\}M) = FI(N)$. Assim, temos o corolário a seguir.

Corolário 3.1.13: Se $\underline{1} \in FI(M)$, então $FI(\{\underline{1}/N\}M) = FI(\lambda.M\ N)$. Caso contrário, $FI(\{\underline{1}/N\}M) = FI(\lambda.M)$.

Note que a mesma propriedade é válida para $\overline{FI}$ quando $\underline{1} \notin FI(M)$, o que não ocorre quando $\underline{1} \in FI(M)$. Como um número positivo de cópias de $\overline{FI}(N)$ é acrescentado no segundo caso, temos que $\overline{FI}(\{\underline{1}/N\}M)$ é maior ou igual a $\overline{FI}(\lambda.M\ N)$.

Lema 3.1.14: Seja $M \in \Lambda_{dB}$ tal que $sup(M) = m$:

1. Se $i < m$ e $\underline{i} \notin FI(M)$, então $sup(\{\underline{i}/N\}M) = m-1$.

2. Se $i > m$, então $sup(\{\underline{i}/N\}M) = m$.

3. Suponha que $\underline{i} \in FI(M)$. Se $FI(M) = \{\underline{i}\}$, então $sup(\{\underline{i}/N\}M) = sup(N)$. Senão, $sup(\{\underline{i}/N\}M) = max(sup(N), m-1)$.

Demonstração. 1. Tem-se que $m \geq n$, $\forall\, \underline{n} \in FI(M)$ e $\underline{m} \in FI(M)$. Pelo Lema 3.1.12.1 tem-se que $\underline{m-1} \in FI(\{\underline{i}/N\}M)$, pois $m > i$, e $\forall\, \underline{j} \in FI(\{\underline{i}/N\}M)$, $j = n < i$ ou $j = n-1$, onde $\underline{n} \in FI(M)$. Assim, $m-1 \geq n-1 \geq i, \forall\, \underline{n} \in FI(M)$ tal que $n > i$ logo $m-1 \geq j, \forall\, \underline{j} \in FI(\{\underline{i}/N\}M)$.

2. Se $i > m$ então, pelo Lema 3.1.12.3, $\{\underline{i}/N\}M \equiv M$ logo $sup(\{\underline{i}/N\}M) = sup(M)$.

3. Pelo Lema 3.1.12.2 tem-se que $FI(\{\underline{i}/N\}M) = FI(N) \cup A$ onde $A = FI(M)_{<i} \cup (FI(M)_{>i} \backslash 1)$. Se $FI(M) = \{\underline{i}\}$, então $A = \emptyset$ logo $FI(\{\underline{i}/N\}M) = FI(N)$. Senão, A não é vazio e, análogo ao caso 1 acima, tem-se que $m-1 \geq j$, $\forall \underline{j} \in A$. □

Lema 3.1.15: $sup(\{\underline{1}/N\}M) \leq sup(\lambda.M\ N)$.

Demonstração. Se $\underline{1} \in FI(M)$, então $sup(\{\underline{1}/N\}M) = sup(\lambda.M\ N)$. Senão existem duas possibilidades. Se $sup(M) = 0$ então pelo Lema 3.1.14.2 tem-se que $sup(\{\underline{1}/N\}M) = 0 \leq max(0, sup(N)) = sup(\lambda.M\ N)$. Se $sup(M) > 1$ então pelo Lema 3.1.14.1 tem-se que $sup(\{\underline{1}/N\}M) = sup(M)-1 = sup(\lambda.M) \leq max(sup(\lambda.M), sup(N))$. □

Finalmente, o teorema a seguir apresenta a propriedade correspondente ao Teorema 2.3.7 para o λ-calculus.

Teorema 3.1.16: Se $M \to_\beta N$ então $FI(N) \subseteq FI(M)$ e $sup(N) \leq sup(M)$.

Demonstração. Por indução na derivação de $M \to_\beta N$.

- Se $M \equiv (\lambda.M_1\ M_2)$ e $N \equiv \{\underline{1}/M_2\}M_1$ então $FI(\{\underline{1}/N\}M_1) \subseteq FI(\lambda.M_1\ M_2)$, pelo Corolário 3.1.13.

- Sejam $M \equiv (M_1\ M_2)$ e $N \equiv (M_1\ N_2)$, onde $M_2 \to_\beta N_2$. Então, por HI, $FI(N_2) \subseteq FI(M_2)$. Assim, $FI(N) = FI(M_1) \cup FI(N_2) \subseteq FI(M_1) \cup FI(M_2) = FI(M)$.

- O caso $M \equiv (M_1\ M_2)$ e $N \equiv (N_1\ M_2)$, onde $M_1 \to_\beta N_1$, é similar ao anterior.

- Se $M \equiv \lambda.M'$, então $N \equiv \lambda.N'$, onde $M' \to_\beta N'$. Por HI, $FI(N') \subseteq FI(M')$ logo $\forall \underline{n} \in FI(N')$, $\underline{n} \in FI(M')$. Assim, $\forall \underline{n-1} \in FI(\lambda.N')$, $\underline{n-1} \in FI(\lambda.M')$. □

Apresentamos a seguir, uma definição da contração η para λ_{dB}, usualmente encontrada na literatura.

Definição 3.1.17: A **contração $\boldsymbol{\eta}$** para λ_{dB} é definida por:

$$\lambda.(M\ \underline{1}) \to_\eta N \qquad \text{se } N^+ = M$$

Assim, a regra é aplicada se existe um termo N tal que a sua elevação resulte no termo M. Em outras palavras, se $\underline{1} \notin FI(M)$. Esta regra não é construtiva, o que levou A. Ríos em [83] a apresentar uma regra construtiva para contração η. Ríos utiliza a declinação de um termo para definir a contração η, apresentada a seguir.

Definição 3.1.18: A **contração construtiva** $\boldsymbol{\eta}$ para λ_{dB} é definida por:

$$\lambda.(M\ \underline{1}) \to_\eta M^- \qquad \text{sempre que } M^- \text{ está bem definido}$$

Temos que se M^- está bem definida então $(M^-)^+ = M$. Portanto, temos um regra construtiva que reduz para o mesmo termo da regra anterior. A regra não construtiva foi utilizada como referência para a adição de uma regra (Eta), com o objetivo de simular a contração η, em dois cálculos de SE: o $\lambda\sigma$ [34] e o λs_e [3]. Em ambos os casos, os cálculos são utilizados no problema de HOU. As regras não construtivas podem ser empregadas em expansões η, utilizadas pelos algoritmos de unificação propostos via cálculos de ES. Porém, no caso de contrações η, as regras não construtivas não possuem a propriedade de SR. Em [99] apresentamos para ambos uma regra (Eta) inspirada na η construtiva de Ríos, que preserva a propriedade de SR. A regra é aplicada em um passo mas tem a decisão da condição de contração η, e a construção do η-redex caso afirmativo, feitas de maneira explícita.

3.1.2 O sistema $\lambda_{dB}^{\to}$, o λ_{dB} com tipos simples

A principal diferença entre um sistema de tipos simples para λ e para λ_{dB} é a estrutura dos contextos de tipo em cada sistema. Enquanto no $\lambda^{\to}$ os contextos se apresentam como conjuntos de designação de tipos para variáveis de termos, estes são sequências de tipos em $\lambda_{dB}^{\to}$. Assim, para sabermos qual o tipo designado para o índice livre $\underline{i}$, basta checarmos o i-ésimo elemento do contexto sequencial. A seguir, apresentamos a definição de contextos sequenciais e algumas operações sobre esta sequência.

Definição 3.1.19 (Contextos sequenciais):

1. Seja $\mathcal{S}$ o conjunto de tipos simples, como apresentado na Definição 2.2.3.2. Os **contextos de tipo sequenciais** são sequências de elementos $\tau \in \mathcal{S}$, definidos por:

$$\Gamma ::= nil \,|\, \tau.\Gamma$$

2. Sejam $n \in \mathbb{N}^*$ e Γ um contexto, então $\Gamma_{<n}$ denota os $n-1$ primeiros tipos da sequência Γ. Similarmente, são definidos $\Gamma_{>n}$, $\Gamma_{\leq n}$ e $\Gamma_{\geq n}$. Observe que, para $\Gamma_{>n}$ e $\Gamma_{\geq n}$ o elemento final *nil* está incluído. Para $n=0$, $\Gamma_{\leq 0}.\Gamma = \Gamma_{<0}.\Gamma = \Gamma_{<1}.\Gamma = \Gamma$.

3. O comprimento de Γ é definido por $|nil|=0$ e $|\Gamma| = 1+|\Gamma_{>1}|$ sempre que $\Gamma \neq nil$.

4. Para qualquer $i > m = |\Gamma|$, seja $\Gamma_{\geq i} = \Gamma_{>i} = \Gamma_{>m} = nil$ e $\Gamma_{\leq i} = \Gamma_{<i} = \Gamma_{\leq m}$.

5. A adição de um tipo τ ao final do contexto Γ é definida por $\Gamma.\tau = \Gamma_{\leq m}.\tau.nil$, onde $|\Gamma| = m$. A extensão para contextos é direta, onde $nil.nil = nil$.

A seguir, apresentamos as regras de inferência para o sistema $\lambda_{dB}^{\rightarrow}$.

Definição 3.1.20: (O sistema $\lambda_{dB}^{\rightarrow}$) As regras de tipagem de $\lambda_{dB}^{\rightarrow}$ são dadas por:

$$\text{(Var)} \quad \frac{}{\underline{1} : \langle \tau.\Gamma \vdash \tau \rangle} \qquad \text{(Varn)} \quad \frac{\underline{n} : \langle \Gamma \vdash \tau \rangle}{\underline{n+1} : \langle \sigma.\Gamma \vdash \tau \rangle}$$

$$\text{(Lambda)} \quad \frac{M : \langle \sigma.\Gamma \vdash \tau \rangle}{\lambda.M : \langle \Gamma \vdash \sigma \rightarrow \tau \rangle} \qquad \text{(App)} \quad \frac{M : \langle \Gamma \vdash \sigma \rightarrow \tau \rangle \qquad N : \langle \Gamma \vdash \sigma \rangle}{(M\ N) : \langle \Gamma \vdash \tau \rangle}$$

A regra $(Varn)$ permite a construção dos contextos sequenciais.

Propriedades como SR e SN para termos tipáveis são geralmente provadas através de um isomorfismo com o sistema $\lambda^{\rightarrow}$ (*cf.* [61]). Apresentamos uma demonstração de SR sem o uso de isomorfismos em [97], para uma versão com tipos anotados no abstratores, onde a adaptação para o sistema $\lambda_{dB}^{\rightarrow}$ apresentado acima é direta.

3.2 O λs_e-calculus

Nesta seção apresentamos o λs-calculus [58] e então o λs_e-calculus [59]. O segundo consiste em uma extensão nas regras de reescrita que compõem o primeiro cálculo, permitindo a composição para substituições. O sistema de tipos simples apresentado infere tipos para os dois cálculos, pois este atribui tipos para o conjunto de termos compartilhado por ambos. Assim, o sistema é denominado $\lambda {s_e}^{\rightarrow}$ e as propriedades, que diferem em cada cálculo, são apresentadas após a introdução das versões sem tipos de cada cálculo.

Apesar do $\lambda\sigma$-calculus [1] ter sido introduzido antes do λs, este representa uma extensão natural do λ_{dB} para a inclusão da substituição na especificação do cálculo. Portanto, optamos por introduzí-lo como um passo intermediário para a introdução do λs_e, de forma que a intuição para substituições explícitas nos cálculos estudados fique mais clara, a partir do entendimento do λ_{dB}-calculus apresentado na seção anterior.

3.2.1 O λs-calculus

O λs-calculus é a extensão natural do λ_{dB}-calculus, onde o cálculo é obtido a partir da orientação das igualdades apresentadas na definição formal da substituição β como uma metaoperação. Os operadores de substituição σ e de atualização φ são introduzidos, para controlar a atomização do processo de substituição através de restrições aritméticas. A sintaxe de λ_{dB} é estendida com termos que têm ocorrências desses operadores na raiz da representação usual de termos em árvore (*cf.* [5]).

Definição 3.2.1: O **conjunto de termos em** λs, denotado por Λs, é definido indutivamente para $n, i, j \in \mathbb{N}^*$ e $k \in \mathbb{N}$ por:

$$M, N \in \Lambda s \quad ::= \quad \underline{n} \,|\, (M\ N) \,|\, \lambda.M \,|\, M\sigma^i N \,|\, \varphi^j_k N$$

A formalização utilizada na obtenção do cálculo tem uma pequena diferença em relação à introduzida na Definição 3.1.3. Na definição de metasubstituição apresentada em [58], a atualização para $FI(N)$ é feita a partir da efetiva substituição de N, ao invés de a cada transição da substituição com λ's. Assim, para a transição com λ tem-se a regra

$$\{\,\underline{i}\,/N\}(\lambda.M) = \lambda.(\{\,\underline{i{+}1}\,/N\}M)$$

e para a troca de índices por termos tem-se a regra

$$\{\,\underline{i}\,/N\}\,\underline{i} = U^i_0(N)$$

onde U^i_k é definido indutivamente para $k \in \mathbb{N}$ e $i \in \mathbb{N}^*$ por

$$(i)\, U^i_k(\lambda.M) = \lambda.(U^i_{k+1}M) \qquad\qquad (ii)\, U^i_k(M_1\ M_2) = (U^i_k M_1\ U^i_k M_2)$$

$$(iii)\, U^i_k\,\underline{n} = \begin{cases} \underline{n{+}i{-}1} & \text{se } n > k \\ \underline{n} & \text{se } n \leq k \end{cases}$$

Portanto, $U^i_k M$ corresponde a $i{-}1$ aplicações da k-elevação ao termo M; *i.e.*, $M^{+k^{(i-1)}}$.

$$
\begin{array}{lrcl}
(\sigma\text{-generation}) & (\lambda.M\ N) & \longrightarrow & M\sigma^1 N \\
(\sigma\text{-}\lambda\text{-transition}) & (\lambda.M)\,\sigma^i N & \longrightarrow & \lambda.(M\sigma^{i+1}N) \\
(\sigma\text{-app-transition}) & (M_1\ M_2)\,\sigma^i N & \longrightarrow & ((M_1\sigma^i N)\ (M_2\,\sigma^i N)) \\
(\sigma\text{-destruction}) & \underline{n}\,\sigma^i N & \longrightarrow & \begin{cases} \underline{n-1} & \text{se } n > i \\ \varphi_0^i N & \text{se } n = i \\ \underline{n} & \text{se } n < i \end{cases} \\
(\varphi\text{-}\lambda\text{-transition}) & \varphi_k^i(\lambda.M) & \longrightarrow & \lambda.(\varphi_{k+1}^i M) \\
(\varphi\text{-app-transition}) & \varphi_k^i(M_1\ M_2) & \longrightarrow & ((\varphi_k^i M_1)\ (\varphi_k^i M_2)) \\
(\varphi\text{-destruction}) & \varphi_k^i\,\underline{n} & \longrightarrow & \begin{cases} \underline{n+i-1} & \text{se } n > k \\ \underline{n} & \text{se } n \leq k \end{cases}
\end{array}
$$

Tabela 3.2.1: O sistema de reescrita para λs

O termo $M\sigma^i N$ representa o início do processo que corresponde ao termo $\{\,\underline{i}\,/N\}M$ de acordo a metasubstituição com as modificações acima ou ao termo $\{\,\underline{i}\,/N^{+(i-1)}\}M$ de acordo a Definição 3.1.3. O termo N em $M\sigma^i N$ é chamado de **corpo da substituição**. O termo $\varphi_k^i M$ representa o início do processo que corresponde ao termo $U_k^i M$.

Definição 3.2.2 (O $\boldsymbol{\lambda s}$-calculus [58]): O λs-calculus é composto pelo conjunto Λs e as regras da Tabela 3.2.1.

Note que $M\sigma^i N$ e $\varphi_k^i M$ representam termos do cálculo, ao invés de denotar apenas o resultado de uma metaoperação. Os termos sem ocorrência dos operadores σ e φ, que são elementos de Λ_{dB}, são chamados **termos puros**.

Definição 3.2.3 ($\boldsymbol{s}$-calculus): O **cálculo de substituição associado a $\boldsymbol{\lambda s}$**, chamado de s-calculus, é composto pelas regras da Tabela 3.2.1 exceto a regra (σ-generation).

O fecho simétrico, transitivo e reflexivo de s é denotado por $=_s$. O teorema a seguir apresenta algumas propriedades tanto de s quanto de λs, como terminação, confluência e simulação da redução β.

Teorema 3.2.4 (Propriedades para $\lambda s/s$-calculi [58]):

1. **(SN e CR para $\boldsymbol{s}$)** O cálculo s é fortemente terminante e confluente.

2. **(Correção)** Sejam $M, N \in \Lambda$. Se $M \to_{\lambda s}^* N$ então $M \to_\beta^* N$.

3. **(Simulação da redução $\boldsymbol{\beta}$)** Sejam $M, N \in \Lambda$. Se $M \to_\beta N$ então $M \to^*_{\lambda s} N$.

4. **(CR para $\boldsymbol{\lambda s}$)** O λs-calculus é confluente.

5. **(PSN para $\boldsymbol{\lambda s}$)** Se um termo puro é SN para β então este é SN em λs.

Pelo Lema 3.2.4.1 acima temos que s é terminante e confluente. Assim, para qualquer $M \in \Lambda s$, M tem uma única forma s-normal correspondente, denotada por $s(M)$.

Lema 3.2.5 (Descrição para $\boldsymbol{s}$-nf [58]): O conjunto de formas s-normais é exatamente o conjunto Λ_{dB}.

O lema a seguir apresenta algumas propriedades de s em relação a estrutura do termo.

Lema 3.2.6 ([58]): Sejam $M, N \in \Lambda s$:

1. $s(M\ N) = (s(M)\ s(N))$.

2. $s(\lambda.M) = \lambda.s(M)$.

3. $s(\varphi^i_k M) = U^i_k(s(M))$.

4. $s(M\sigma^i N) = \{\,\underline{i}\,/U^i_0(s(N))\}s(M)$.

O item 4 foi adaptado de acordo à Definição 3.1.3, para substituição β.

Observe que, para um termo $M\sigma^i N$ onde M e N são s-nfs, os conjuntos $FI(M)$ e $FI(N)$ ainda serão atualizados de acordo com as regras de reescrita de λs. Portanto, o conceito de índices livres em λs não representa uma característica sintática apropriada como para o λ_{dB}, *e.g.* a identificação de uma substituição nula. Como λs não permite composição de substituições, a noção de variável disponível (*available variable*) apresentada para o λx em [69] é uma alternativa pertinente. A idéia é que o conjunto tenha relação com os índices livres da s-nf correspondente. Assim, apresentamos a definição para o conjunto de índices disponíveis.

Definição 3.2.7: Seja $M \in \Lambda s$:

1. O **conjunto dos índices disponíveis** de M, denotado por $AI(M)$, é definido indutivamente por:

(i) $AI(\underline{n}) = \{\underline{n}\}$

(ii) $AI(\lambda.M) = AI(M)\backslash 1$

(iii) $AI(M_1\ M_2) = AI(M_1) \cup AI(M_2)$

(iv) $AI(\varphi_k^i M) = AI(M)_{\leq k} \cup (AI(M)_{>k} + (i-1))$

(v) $AI(M\sigma^i N) = \begin{cases} AI(M^{-i}) \cup AI(\varphi_0^i N), & \text{se } i \in AI(M) \\ AI(M^{-i}), & \text{c.c} \end{cases}$

onde $AI(M^{-i})$ denota $AI(M)_{<i} \cup (AI(M)_{>i})\backslash 1$ e $AI(\varphi_k^i N)$ denota $AI(N)_{\leq k} \cup (AI(N)_{>k} + (i-1))$.

2. Seja o índice disponível de M com o maior valor absoluto, denotado por $sav(M)$, definido por $sav(M) = max(AI(M))$, onde $sav(\emptyset) = 0$.

A seguir, apresentamos a relação entre os conjuntos AI e FI.

Lema 3.2.8: Se $M \in \Lambda s$ então $AI(M) = FI(s(M))$.

Demonstração. Por indução na estrutura de $M \in \Lambda s$.

- Se $M \equiv \underline{n}$, nada há provar.

- Seja $M \equiv \lambda.M'$. Pelo Lema 3.2.6.2 tem-se que $s(\lambda.M') = \lambda.s(M')$. Por HI tem-se que $AI(M') = FI(s(M'))$ logo $AI(\lambda.M') = AI(M')\backslash 1 = FI(s(M'))\backslash 1 = FI(\lambda.s(M'))$.

- Seja $M \equiv (M_1\ M_2)$. Pelo Lema 3.2.6.1 tem-se que $s(M_1\ M_2) = (s(M_1)\ s(M_2))$. Portanto, por HI tem-se que $AI(M_1) = FI(s(M_1))$ e $AI(M_2) = FI(s(M_2))$ logo $AI(M_1\ M_2) = AI(M_1) \cup AI(M_2) = FI(s(M_1)) \cup FI(s(M_2)) = FI(s(M_1)\ s(M_2))$.

- Seja $M \equiv \varphi_k^i N$. Pelo Lema 3.2.6.3 tem-se que $s(\varphi_k^i N) = U_k^i(s(N))$ e por HI tem-se que $AI(N) = FI(s(N))$. Note que $AI(\varphi_k^i N) = AI(N)_{\leq k} \cup (AI(N)_{>k} + (i-1))$ e, por indução em i e pelo Lema 3.1.112, tem-se que $FI(s(N)^{+k^{(i-1)}}) = FI(s(N))_{\leq k} \cup (FI(s(N))_{>k} + (i-1))$. Portanto, $AI(\varphi_k^i N) = FI(s(N)^{+k^{(i-1)}}) = FI(U_k^i(s(N)))$.

- Seja $M \equiv M_1\sigma^i M_2$. Pelo Lema 3.2.6.4 tem-se $s(M_1\sigma^i M_2) = \{\underline{i}/U_0^i(s(M_2))\}s(M_1)$. Por HI tem-se que $AI(M_1) = FI(s(M_1))$ logo $\underline{i} \in AI(M_1)$ sse $\underline{i} \in FI(s(M_1))$. Observe que $AI(M_1^{-i}) = AI(M_1)_{<i} \cup (AI(M_1)_{>i}\backslash 1) = FI(s(M_1))_{<i} \cup (FI(s(M_1))_{>i}\backslash 1)$.

Se $\underline{i} \notin AI(M_1)$, então $AI(M_1\sigma^i M_2) = AI(M_1^{-i})$ e, pelo Lema 3.1.12.1, tem-se que $FI(\{\underline{i}/U_0^i(s(M_2))\}s(M_1)) = FI(s(M_1))_{<i} \cup (FI(s(M_1))_{>i}\backslash 1)$ logo $AI(M_1\sigma^i M_2) = FI(\{\underline{i}/U_0^i(s(M_2))\}s(M_1))$.

Se $\underline{i} \in AI(M_1)$, então $AI(M_1\sigma^i M_2) = AI(M_1^{-i}) \cup AI(\varphi_0^i M_2)$ e, pelo Lema 3.1.12.2, $FI(\{\underline{i}/U_0^i(s(M_2))\}s(M_1)) = FI(U_0^i(s(M_2))) \cup FI(s(M_1))_{<i} \cup (FI(s(M_1))_{>i}\backslash 1)$. Pelo Lema 3.2.6.3 tem-se $s(\varphi_0^i M_2) = U_0^i(s(M_2))$ logo, por HI, $AI(\varphi_0^i M_2) = FI(s(\varphi_0^i M_2)) = FI(U_0^i(s(M_2)))$. Portanto, $AI(M_1\sigma^i M_2) = FI(\{\underline{i}/U_0^i(s(M_2))\}s(M_1))$. □

As propriedades de AI e sav são similares as de FI e sup, apresentadas na Seção 3.1.1. As provas que podem ser obtidas de provas similares para estes dois últimos serão omitidas. A seguir, apresentamos algumas dessas propriedades.

Lema 3.2.9: $\underline{n} \in AI(\lambda.M)$ se, e somente se, $\underline{n+1} \in AI(M)$.

Demonstração. Similar à prova do Lema 3.1.9.1. □

Lema 3.2.10:

1. $sav(M_1\ M_2) = max(sav(M_1), sav(M_2))$.
2. Se $sav(M) = 0$, então $sav(\lambda.M) = 0$. Senão, $sav(\lambda.M) = sav(M) - 1$.
3. Se $sav(N) > k$ então $sav(\varphi_k^i N) = sav(N) + (i-1)$. Se $sav(N) \leq k$ então $sav(\varphi_k^i N) = sav(N)$.
4. Seja $sav(M^{-i}) = max(AI(M^{-i})) = max(AI(M)_{<i} \cup (AI(M)_{>i})\backslash 1)$. Se $sav(M) < i$, então $sav(M^{-i}) = sav(M)$. Se $sav(M) > i$, então $sav(M^{-i}) = sav(M) - 1$.
5. Se $\underline{i} \in sav(M)$, então $sav(M\sigma^i N) = max(sav(M^{-i}), sav(\varphi_0^i N))$. Caso contrário, $sav(M\sigma^i N) = sav(M^{-i})$.

Demonstração.

1. Similar ao Lema 3.1.10.1.
2. Similar ao Lema 3.1.10.2.
3. Se $sav(N) \leq k$ então $AI(\varphi_k^i N) = AI(N)_{\leq k} = AI(N)$. Se $sav(N) > k$, então $AI(N)_{>k} \neq \emptyset$. Logo, $sav(\varphi_k^i N) = max(AI(N)_{>k} + (i-1)) = max(AI(N)_{>k}) + (i-1) = sav(N) + (i-1)$.

4. Se $sav(M) < i$, então $AI(M)_{>i} = \emptyset$. Logo $AI(M)_{<i} \cup (AI(M)_{>i})\backslash 1 = AI(M)_{<i} = AI(M)$. Se $sav(M) > i$, então $AI(M)_{>i} \neq \emptyset$ logo $max(AI(M)_{<i} \cup (AI(M)_{>i})\backslash 1) = max((AI(M)_{>i})\backslash 1) = sav(M) - 1$.

5. Se $\underline{i} \in sav(M)$, então $AI(M\sigma^i N) = AI(M^{-i}) \cup AI(\varphi_0^i N)$. Logo, $max(AI(M\sigma^i N)) = max(AI(M^{-i}) \cup AI(\varphi_0^i N)) = max(max(AI(M^{-i})), max(AI(\varphi_0^i N)))$. Portanto, $sav(M\sigma^i N) = max(sav(M^{-i}), sav(\varphi_0^i N))$. Se $\underline{i} \notin sav(M)$, então $AI(M\sigma^i N) = AI(M^{-i})$ e o resultado é direto. □

3.2.2 O λs_e-calculus

Na aplicação de cálculos de substituições explícitas para o problema de unificação de ordem superior (HOU) [3], um **conjunto de metavariáveis**, denotado por $\mathcal{X}$, é acrescentado a formação de termos. Os termos com ocorrências de metavariáveis são chamados **abertos**. Metavariáveis são substituídas por termos do cálculo com técnicas próprias, que diferem da substituição β. De fato, metavariáveis são definidas de forma que sejam invariantes para a operação de i-elevação da Definição 3.1.2 e para a susbtituição β da Definição 3.1.3.

Assim, ao acrescentarmos o conjunto $\mathcal{X}$ na formação de termos em λs obtemos o conjunto $\Lambda s(\mathcal{X})$. O λs não é confluente quando aplicado aos termos de $\Lambda s(\mathcal{X})$, ou seja, o cálculo com as regras apresentadas na Tabela 3.2.1 não é confluente para termos abertos. Para recuperar a propriedade de confluência para $\Lambda s(\mathcal{X})$, as regras apresentadas na Tabela 3.2.2 são acrescentadas ao λs, obtendo o λs_e-calculus [59].

(σ-σ-transition)	$(M_1\sigma^i M_2)\,\sigma^j N$	$\longrightarrow$	$(M_1\sigma^{j+1}N)\,\sigma^i(M_2\,\sigma^{j-i+1}N)$	se $i \leq j$
(σ-φ-transition 1)	$(\varphi_k^i M)\,\sigma^j N$	$\longrightarrow$	$\varphi_k^{i-1} M$	se $k < j < k+i$
(σ-φ-transition 2)	$(\varphi_k^i M)\,\sigma^j N$	$\longrightarrow$	$\varphi_k^i (M\sigma^{j-i+1}N)$	se $k+i \leq j$
(φ-σ-transition)	$\varphi_k^i (M\sigma^j N)$	$\longrightarrow$	$(\varphi_{k+1}^i M)\,\sigma^j(\varphi_{k+1-j}^i N)$	se $j \leq k+1$
(φ-φ-transition 1)	$\varphi_k^i(\varphi_l^j M)$	$\longrightarrow$	$\varphi_l^j(\varphi_{k+1-j}^i M)$	se $l+j \leq k$
(φ-φ-transition 2)	$\varphi_k^i(\varphi_l^j M)$	$\longrightarrow$	$\varphi_l^{j+i-1} M$	se $l \leq k < l+j$

Tabela 3.2.2: As regras de reescrita de extensão para o λs_e

As regras da Tabela 3.2.2 permitem a composição dos operadores de substituição e atualização, com restrições aritméticas. Assim como para λs, temos um cálculo de substituição associado a λs_e.

Definição 3.2.11 (s_e-calculus): O **cálculo de substituição associado a $\boldsymbol{\lambda s_e}$**, chamado de s_e-calculus, é composto pelas regras das Tabelas 3.2.1 e 3.2.2, exceto a regra (σ-generation).

Em outras palavras s_e é o cálculo s acrescentado das regras da Tabela 3.2.2. O fecho simétrico, transitivo e reflexivo de s_e é denotado por $=_{s_e}$. Assim, o teorema a seguir apresenta propriedades de s_e e λs_e, similar ao Teorema 3.2.4 para s e λs.

Teorema 3.2.12 (Propriedades para $\lambda s_e/s_e$-calculi [59]):

1. **(WN e CR de $\boldsymbol{s_e}$)** O cálculo s_e é fracamente terminante e confluente.

2. **(Correção)** Sejam $M, N \in \Lambda$. Se $M \to^*_{\lambda s_e} N$ então $M \to^*_\beta N$.

3. **(Simulação da redução $\boldsymbol{\beta}$)** Sejam $M, N \in \Lambda$. Se $M \to_\beta N$ então $M \to^*_{\lambda s_e} N$.

4. **(CR para $\boldsymbol{\lambda s_e}$)** O λs_e-calculus é confluente para termos abertos.

Observe que $\Lambda s \subset \Lambda s(\mathcal{X})$ logo as propriedades acima valem para os termos sem ocorrências de metavariáveis. Porém, ao permitir a composição para os operadores σ e φ, a propriedade PSN não é satisfeita para o λs_e-calculus. Em [47] Guillaume apresenta um contraexemplo, onde um termo SN no λ_{dB}-calculus tem uma estratégia de redução infinita em λs_e. De fato, o contra-exemplo é tipável em $\lambda_{\overrightarrow{dB}}$, o que representa um subconjunto próprio dos termos SN em λ_{dB}.

Neste trabalho, estudamos as regras do λs_e aplicadas aos termos de Λs. Assim, definimos o λs_e utilizado neste trabalho a seguir.

Definição 3.2.13 (O $\boldsymbol{\lambda s_e}$-calculus): O λs_e-calculus é definido como o λs-calculus estendido com as regras da Tabela 3.2.2.

O λs_e como definido acima cumpre o papel do cálculo de SE onde a composição para substituições é permitida. A adição de metavariáveis neste estágio traria uma complexidade desnecessária para o estudo de um sistema de IT para o cálculo.

Pelo Lema 3.2.12.1 acima temos que s_e é fracamente terminante e confluente. Assim, para qualquer $M \in \Lambda s$, M tem uma única forma s_e-normal correspondente, denotada por

$s_e(M)$. Note que, quando restrito ao conjunto Λs, as formas s_e-normais são exatamente o conjunto Λ_{dB}.

3.2.3 O sistema $\lambda {s_e}^{\rightarrow}$, o λs_e com tipos simples

Como a sintaxe de λs_e é próxima a do λ_{dB}, para obtermos um sistema de tipos para o primeiro acrescenta-se as regras de inferência ao $\lambda_{dB}^{\rightarrow}$ para as duas novas formas de termos.

Definição 3.2.14 (O sistema $\lambda {s_e}^{\rightarrow}$): O sistema $\lambda {s_e}^{\rightarrow}$ é composto por (Var), (Varn), (App), (Lambda) da Definição 3.1.20 e as regras a seguir:

$$\text{(Sigma)} \quad \frac{N : \langle \Gamma_{\geq i} \vdash \rho \rangle \qquad M : \langle \Gamma_{<i}.\rho.\Gamma_{\geq i} \vdash \tau \rangle}{M\sigma^i N : \langle \Gamma \vdash \tau \rangle}, \quad \text{onde } |\Gamma| \geq i{-}1$$

$$\text{(Phi)} \quad \frac{M : \langle \Gamma_{\leq k}.\Gamma_{\geq k+i} \vdash \tau \rangle}{\varphi_k^i M : \langle \Gamma \vdash \tau \rangle}, \quad \text{onde } |\Gamma| \leq k{+}i{-}1$$

O sistema $\lambda {s_e}^{\rightarrow}$ é uma versão à la Curry do sistema de tipos para o λs_e-calculus a qual foi verificada que possui as mesmas propriedades da versão à la Church apresentada em [3]. Em particular, as propriedades como WN para s_e e CR são herdadas do cálculo sem tipos apresentado acima. A seguir, apresentamos as propriedades dos cálculos tipados.

Lema 3.2.15 (Propriedades para $\boldsymbol{\lambda {s_e}^{\rightarrow}}$ [60]):

1. **(SR para $\boldsymbol{\lambda s_e}$)** Se $M \rightarrow_{\lambda s_e} N$ e $M : \langle \Gamma \vdash_{\lambda {s_e}^{\rightarrow}} \tau \rangle$ então $N : \langle \Gamma \vdash_{\lambda {s_e}^{\rightarrow}} \tau \rangle$.
2. **(WN para $\boldsymbol{\lambda s_e}$)** Se $M : \langle \Gamma \vdash_{\lambda {s_e}^{\rightarrow}} \tau \rangle$ então M é WN.
3. **(SN para $\boldsymbol{\lambda s}$)** Se $M : \langle \Gamma \vdash_{\lambda {s_e}^{\rightarrow}} \tau \rangle$ então M é SN em λs.

Para a propriedade de SR em relação ao λs_e-calculus, foi apresentada uma demonstração para a versão com tipos explícitos em [97] tal que a sua adaptação para essa versão com tipos implícitos é direta.

3.3 O $\lambda\sigma$-calculus

O $\lambda\sigma$-calculus foi introduzido em [1], considerado o trabalho seminal em substituições explícitas. A aplicação deste cálculo para o problema de HOU apresentado em [34] é pioneira no uso de ES para unificação de ordem superior. F. Moura *et al.* apresenta em [32] uma comparação entre o método clássico para o tratamento de HOU no λ-calculus tipado com tipos simples de G. Huet [51] e o realizado via $\lambda\sigma$.

Ao contrário da abordagem adotada no λs, e no λs_e, o $\lambda\sigma$-calculus é composto por um sistema de reescrita de primeira ordem, que explicita a operação de substituição estendendo a linguagem com duas classes de objetos: os **termos** e as **substituições**, chamadas de **expressões em** $\lambda\sigma$.

Definição 3.3.1: O **conjunto de expressões em** $\lambda\sigma$, denotado por $\Lambda\sigma$, é um conjunto formado por duas classes de objetos: a classe dos **termos**, denotada por $\Lambda\sigma^t$, e a classe das **substituições**, denotada por $\Lambda\sigma^s$, definidas por:

$$\begin{aligned} M, N \in \Lambda\sigma^t &::= \underline{1} \,|\, (M\ N) \,|\, \lambda.M \,|\, M[S] \\ S \in \Lambda\sigma^s &::= id \,|\, \uparrow \,|\, M.S \,|\, S \circ S \end{aligned}$$

As substituições em $\lambda\sigma$ representam listas com elementos da forma $N_i/\underline{i}$, indicando que o índice $\underline{i}$ deve ser trocado pelo termo N_i. A expressão id representa a substituição da forma $\{\,\underline{i}\,/\underline{i}\,|\,\forall i\in\mathbb{N}^*\}$ enquanto o **shift**, denotado por $\uparrow$, representa a substituição $\{\,\underline{i+1}\,/\underline{i}\,|\forall i\in\mathbb{N}^*\}$. A expressão $S \circ S$ representa a composição de substituições e usaremos $\uparrow^n$ para denotar a composição de n *shifts*, onde $\uparrow^0$ denota id. O termo $\underline{1}\,[\uparrow^n]$, onde $n \in \mathbb{N}^*$, codifica o índice $\underline{n+1}$ em $\lambda\sigma$ e a expressão $\underline{i}\,[S]$ representa o valor de $\underline{i}$ designado pela substituição S, que pode ser vista informalmente como uma função $S(i)$. A substituição $M.S$, chamada de **cons de** M **em** S, tem a forma $\{M/\underline{1}\,, S(i)/\underline{i+1}\,\}$. O termo $M[N.id]$ inicia o processo de simulação da contração β de $(\lambda.M\ N)$ em $\lambda\sigma$. Portanto, além da substituição das ocorrências livres do índice $\underline{1}$ pelo termo correspondente, as ocorrências livres de índices deve ser decrementada, por causa da eliminação do abstrator. A Tabela 3.3.1 apresenta as regras de reescrita do $\lambda\sigma$-calculus, como apresentadas em [34], porém sem a regra (Eta).

Definição 3.3.2 (O $\lambda\sigma$-calculus): O $\lambda\sigma$-calculus é formado pelo conjunto $\Lambda\sigma$ e as regras da Tabela 3.3.1.

Esse sistema é apresentado em [83] como $\lambda\sigma'$. Esse cálculo é apresentado em [1] como uma variação de $\lambda\sigma$, com a conjectura de que este seja confluente para termos abertos,

$(Beta)$	$(\lambda.M\ N)$	$\longrightarrow$	$M[N.id]$
(App)	$(M\ N)[S]$	$\longrightarrow$	$(M[S]\ N[S])$
$(VarCons)$	$\underline{1}\,[M.S]$	$\longrightarrow$	M
(Id)	$M[id]$	$\longrightarrow$	M
(Abs)	$(\lambda.M)[S]$	$\longrightarrow$	$\lambda.(M[\,\underline{1}.(S\circ\uparrow)])$
$(Clos)$	$(M[S])[S']$	$\longrightarrow$	$M[S\circ S']$
(IdL)	$id\circ S$	$\longrightarrow$	S
$(ShiftCons)$	$\uparrow\circ\,(M.S)$	$\longrightarrow$	S
$(AssEnv)$	$(S_1\circ S_2)\circ S_3$	$\longrightarrow$	$S_1\circ(S_2\circ S_3)$
$(MapEnv)$	$(M.S)\circ S'$	$\longrightarrow$	$M[S'].(S\circ S')$
(IdR)	$S\circ id$	$\longrightarrow$	S
$(VarShift)$	$\underline{1}\,.\!\uparrow$	$\longrightarrow$	id
$(Scons)$	$\underline{1}\,[S].(\uparrow\circ S)$	$\longrightarrow$	S

Tabela 3.3.1: O sistema de reescrita para $\lambda\sigma$

pois o $\lambda\sigma$ apresentado em [1] não possui tal propriedade. Portanto, $\lambda\sigma$ e σ correspondem a $\lambda\sigma'$ e σ' sempre que [83] for referenciado. Assim como para o λs_e, consideramos o $\lambda\sigma$ da Tabela 3.3.1 apenas para o subconjunto de expressões sem metavariáveis, como descrito na Definição 3.3.1, apesar do cálculo ser originalmente considerado para termos abertos.

Definição 3.3.3 ($\boldsymbol{\sigma}$-calculus): O **cálculo de substituição associado a $\boldsymbol{\lambda\sigma}$**, chamado de σ-calculus, é composto por todas as regras da Tabela 3.3.1 exceto a regra $(Beta)$.

O fecho simétrico, transitivo e reflexivo de σ é denotado por $=_\sigma$. O teorema a seguir apresenta as propriedades de σ e $\lambda\sigma$.

Teorema 3.3.4 (Propriedades para $\lambda\sigma/\sigma$-calculi [83]):

1. **(SN e CR para $\boldsymbol{\sigma}$)** O cálculo σ é fortemente terminante e confluente.

2. **(Correção)** Sejam $M, N \in \Lambda$. Se $M \rightarrow^*_{\lambda\sigma} N$ então $M \rightarrow^*_\beta N$.

3. **(Simulação da redução $\boldsymbol{\beta}$)** Sejam $M, N \in \Lambda$. Se $M \rightarrow_\beta N$ então $M \rightarrow^*_{\lambda\sigma} N$.

4. **(CR para $\boldsymbol{\lambda\sigma}$)** O $\lambda\sigma$-calculus é confluente para termos fechados para substituição.

Em [26] foi provado que o $\lambda\sigma$-calculus descrito nesse trabalho não é confluente para expressões abertas, ou seja, quando metavariáveis para substituições são permitidas na formação de expressões do cálculo. Assim, **termos fechados para substituição** são expressões onde apenas metavariáveis para termos são permitidas.

Pelo Lema 3.3.4.1 acima temos que σ é terminante e confluente. Assim, para qualquer $M \in \Lambda\sigma$, M tem uma única forma σ-normal correspondente, denotada por $\sigma(M)$.

Lema 3.3.5 (Descrição para σ-nf): As expressões em forma σ-normal são definidas recursivamente para $n \in \mathbb{N}^*$ por:

$$\begin{array}{lcl} M & ::= & \underline{1} \,|\, (M\ M) \,|\, \lambda.M \,|\, \underline{1}\,[\uparrow^n] \\ S & ::= & id \,|\, \uparrow^n \,|\, M.S \end{array}$$

onde $M.S \neq \underline{1}.\!\uparrow$ e $M.S \neq \underline{1}\,[\uparrow^n].\!\uparrow^{n+1}$.

Demonstração. Pela Proposição 3.2 de [83], restrita a expressões sem ocorrências de metavariáveis. □

Observe que, pelo Lema 3.3.5 acima, para um isomorfismo entre os termos em σ-nf e o conjunto Λ_{dB}, basta que este seja compatível com a estrutura dos termos e que faça a correspondência entre $\underline{1}\,[\uparrow^n]$ e o índice de de Bruijn apropriado.

Definição 3.3.6: Seja $* : \Lambda\sigma^t \to \Lambda_{dB}$ para $n \in \mathbb{N}^*$ tal que:

(i) $\underline{1}^* = \underline{1}$

(ii) $(\underline{1}\,[\uparrow^n])^* = \underline{n+1}$

(iii) $(M\ N)^* = (M^*\ N^*)$

(iv) $(\lambda.M)^* = \lambda.M^*$

Proposição 3.3.7: $*$ é um isomorfismo entre Λ_{dB} e os termos em forma σ-normal.

Demonstração. Por indução na estrutura dos termos de Λ_{dB} e dos termos em σ-nf, descritos no Lema 3.3.5. □

O termo $\underline{1}\,[\uparrow^n]$ será abreviado por $\underline{n+1}$, sempre que não houver confusão. O próximo lema apresenta descrição das formas $\lambda\sigma$-normais.

Lema 3.3.8 (Descrição para $\boldsymbol{\lambda\sigma}$-nf [83]): As expressões em forma $\lambda\sigma$-normal são descritas indutivamente por:

(i) $\lambda.N$, onde N é uma $\lambda\sigma$-nf.

(ii) $(\,\underline{n}\ N_1 \cdots N_p)$, onde $\forall 0 \leq j \leq p$, N_j é uma $\lambda\sigma$-nf.

(iii) $N_1.\cdots.N_p.\!\uparrow^n$, para $n \in \mathbb{N}$ e $\forall 0 \leq j \leq p$, N_j é uma $\lambda\sigma$-nf tal que $N_p \neq \underline{n}$.

Note que a prova do Lema 3.3.8 acima pode ser feita a partir do Lema 3.1.6, o isomorfismo $*$ da Definição 3.3.6 e pela descrição das formas σ-normais no Lema 3.3.5.

Assim como para o λs_e-calculus, a composição de substituições permitida em $\lambda\sigma$ invalida a propriedade de PSN para o cálculo. Em [74], Melliès apresenta um contraexemplo onde um termo que corresponde a um termo SN_β em λ_{dB} tem uma estratégia de redução infinita em $\lambda\sigma$. O contra-exemplo é de fato tipável em $\lambda\sigma^{\rightarrow}$ apresentado na seguinte seção.

3.3.1 O sistema $\lambda\sigma^{\rightarrow}$, o $\lambda\sigma$ com tipos simples

A regras de tipagem do $\lambda\sigma$-calculus designam tipos para objetos da classe de termos como para objetos da classe de substituições. Um objeto da classe de substituições, devido a sua semântica, pode ser visto como uma lista de termos. Consequentemente, seu tipo é também uma sequência de tipos, ou seja, um contexto. Na notação usual para o $\lambda\sigma$ com tipos simples, $S \rhd \Gamma$ denota que a substituição S tem o tipo Γ. Portanto, na notação utilizada no presente trabalho, uma tipagem de objetos da classe de substituições será denotada por $\langle \Gamma \rhd \Gamma' \rangle$.

Definição 3.3.9 (O sistema $\lambda\sigma^{\rightarrow}$): O sistema $\lambda\sigma^{\rightarrow}$ é dado pelas regras de tipagem a seguir:

$$\text{(var)}\quad \frac{}{\underline{1} : \langle \tau.\Gamma \vdash \tau \rangle} \qquad\qquad \text{(lambda)}\quad \frac{M : \langle \sigma.\Gamma \vdash \tau \rangle}{\lambda.M : \langle \Gamma \vdash \sigma \rightarrow \tau \rangle}$$

$$\text{(app)}\quad \frac{M_1 : \langle \Gamma \vdash \sigma \rightarrow \tau \rangle \qquad M_2 : \langle \Gamma \vdash \sigma \rangle}{(M_1\ M_2) : \langle \Gamma \vdash \tau \rangle} \qquad\qquad \text{(clos)}\quad \frac{S : \langle \Gamma \rhd \Gamma' \rangle \qquad M : \langle \Gamma' \vdash \tau \rangle}{M[S] : \langle \Gamma \vdash \tau \rangle}$$

$$\text{(id)}\quad \frac{}{id : \langle \Gamma \rhd \Gamma \rangle} \qquad\qquad \text{(shift)}\quad \frac{}{\uparrow : \langle \tau.\Gamma \rhd \Gamma \rangle}$$

$$\text{(cons)}\quad \frac{M : \langle \Gamma \vdash \tau \rangle \qquad S : \langle \Gamma \rhd \Gamma' \rangle}{M.S : \langle \Gamma \rhd \tau.\Gamma' \rangle} \qquad\qquad \text{(comp)}\quad \frac{S : \langle \Gamma \rhd \Gamma' \rangle \qquad S' : \langle \Gamma' \rhd \Gamma'' \rangle}{S' \circ S : \langle \Gamma \rhd \Gamma'' \rangle}$$

Observe que o nome da regras de tipagem começam com letra minúscula enquanto o das regras de reescrita começam com letra maiúscula. A seguir, apresentamos um lema descrevendo propriedades do cálculo tipado.

Lema 3.3.10 (Propriedades para $\boldsymbol{\lambda\sigma^{\rightarrow}}$):

1. **(SR para $\boldsymbol{\lambda\sigma^{\rightarrow}}$ [1])** Se $M \rightarrow_{\lambda\sigma} N$ e $M : \langle \Gamma \vdash_{\lambda\sigma^{\rightarrow}} \tau \rangle$, então $N : \langle \Gamma \vdash_{\lambda\sigma^{\rightarrow}} \tau \rangle$. Em particular, se $S : \langle \Gamma \rhd_{\lambda\sigma^{\rightarrow}} \Gamma' \rangle$ e $S \rightarrow_{\lambda\sigma} S'$, então $S' : \langle \Gamma \rhd_{\lambda\sigma^{\rightarrow}} \Gamma' \rangle$.

2. **(WN para $\boldsymbol{\lambda\sigma}$ [76])** Se $M : \langle \Gamma \vdash_{\lambda\sigma^{\rightarrow}} \tau \rangle$ então M é WN. Em particular, se $S : \langle \Gamma \rhd_{\lambda\sigma^{\rightarrow}} \Gamma' \rangle$ então S é WN.

Como para o λs_e, a versão com tipos do $\lambda\sigma$-calculus é apresentada à la Curry ao invés da usual versão à la Church de [34]. As propriedades de WN para σ e CR são herdadas do cálculo sem tipos. Para SR, a demonstração apresentada em [97] para a versão tipada é tal que a adaptação para a versão com tipos implícitos é direta.

Capítulo 4

Tipagem principal para sistemas de tipos simples

Neste capítulo introduzimos a noção de tipagem principal para os sistemas de tipos simples apresentados no Capítulo 3. Esse trabalho foi apresentado como parte da dissertação de mestrado, onde apresentamos as definições de PT para cada sistema e provamos que as definições são corretas, de acordo à definição geral de Wells [103], apresentada na Definição 2.2.2. O trabalho estendido com a prova de correspondência entre as definições, ou seja se é PT de acordo à Definição 2.2.2 então é PT de acordo à definição específica no respectivo sistema, foi publicado em [98].

Como visto na Seção 2.3.2, o conceito de tipagem principal em sistemas de tipos simples está ligado à substituição de tipos e à lei de redundância. Estes dois conceitos têm um papel central na definição de PT específicas para os sistemas $\lambda_{dB}^{\rightarrow}$, $\lambda s_e{}^{\rightarrow}$ e $\lambda\sigma^{\rightarrow}$ apresentados neste capítulo. O conjunto de tipos $\mathcal{S}$ para estes sistemas é o apresentado na Definição 2.2.3.2. A substituição de tipos é estendida para contextos sequenciais como definido abaixo.

Definição 4.0.11: Seja $s : \mathcal{A} \rightarrow \mathcal{S}$ como na Definição 2.2.4. A extensão de s para contextos sequenciais é tal que $s(nil) = nil$ e $s(\tau.\Gamma) = s(\tau).s(\Gamma)$. A extensão para tipagens é dada por $s(\Theta) = \langle s(\Gamma) \vdash s(\tau)\rangle$.

Nesse capítulo, variável de tipo e variável de contextos são abreviadas por v.t. e v.c., respectivamente.

4.1 Tipagem principal para o sistema $\lambda_{dB}^{\rightarrow}$

A Figura 4.1.1 contém as regras de inferência de $\lambda_{dB}^{\rightarrow}$, apresentadas na Definição 3.1.20.

$$\text{(Var)} \quad \frac{}{\underline{1}:\langle\tau.\Gamma\vdash\tau\rangle} \qquad \text{(Varn)} \quad \frac{\underline{n}:\langle\Gamma\vdash\tau\rangle}{\underline{n+1}:\langle\sigma.\Gamma\vdash\tau\rangle}$$

$$\text{(Lambda)} \quad \frac{M:\langle\sigma.\Gamma\vdash\tau\rangle}{\lambda.M:\langle\Gamma\vdash\sigma\rightarrow\tau\rangle} \qquad \text{(App)} \quad \frac{M:\langle\Gamma\vdash\sigma\rightarrow\tau\rangle \qquad N:\langle\Gamma\vdash\sigma\rangle}{(M\ N):\langle\Gamma\vdash\tau\rangle}$$

Figura 4.1.1: O sistema $\lambda_{dB}^{\rightarrow}$

O lema a seguir mostra que em uma tipagem de M, pelo menos os índices livres de M têm tipos designados em seu contexto.

Lema 4.1.1: Seja $M:\langle\Gamma\vdash\tau\rangle$:

1. $|\Gamma| \geq sup(M)$
2. $M:\langle\Gamma_{\leq i}.nil\vdash\tau\rangle$, para qualquer $i \geq sup(M)$.

Demonstração. Indução na derivação de $M:\langle\Gamma\vdash\tau\rangle$.

1.
   - Se $\underline{1}:\langle\tau.\Gamma\vdash\tau\rangle$, nada há provar.
   - Seja $\dfrac{\underline{n}:\langle\Gamma\vdash\tau\rangle}{\underline{n+1}:\langle\sigma.\Gamma\vdash\tau\rangle}$. Por IH, $|\Gamma| \geq sup(\underline{n})=n$. Logo, $|\sigma.\Gamma|=1+|\Gamma| \geq n+1$.
   - Seja $\dfrac{M:\langle\sigma.\Gamma\vdash\tau\rangle}{\lambda.M:\langle\Gamma\vdash\sigma\rightarrow\tau\rangle}$. Pelo Lema 3.1.10.2, tem-se que $sup(M)=sup(\lambda.M)=0$ ou $sup(\lambda.M)=sup(M)-1$. No primeiro caso, tem-se $|\Gamma|\geq sup(\lambda.M)=0$ para qualquer contexto Γ. No último, tem-se por IH que $1+|\Gamma|=|\sigma.\Gamma|\geq sup(M)$, logo $|\Gamma|\geq sup(M)-1=sup(\lambda.M)$.
   - Seja $\dfrac{M:\langle\Gamma\vdash\sigma\rightarrow\tau\rangle \qquad N:\langle\Gamma\vdash\sigma\rangle}{(M\ N):\langle\Gamma\vdash\tau\rangle}$. Por HI tem-se que $|\Gamma|\geq sup(M)$ e $|\Gamma|\geq sup(N)$. Logo, $|\Gamma| \geq max(sup(M), sup(N)) = sup(M\ N)$.

2.
   - Se $\underline{1}:\langle\tau.\Gamma\vdash\tau\rangle$, nada há provar.
   - Seja $\dfrac{\underline{n}:\langle\Gamma\vdash\tau\rangle}{\underline{n+1}:\langle\sigma.\Gamma\vdash\tau\rangle}$. Por HI, $\underline{n}:\langle\Gamma_{\leq i}.nil\vdash\tau\rangle$ onde $i\geq n$. Logo, por (Varn), $\underline{n+1}:\langle\sigma.\Gamma_{\leq i}.nil\vdash\tau\rangle$. Note que $(\sigma.\Gamma)_{\leq(i+1)} = \sigma.\Gamma_{\leq i}$.

- Seja $\dfrac{M:\langle\sigma.\Gamma\vdash\tau\rangle}{\lambda.M:\langle\Gamma\vdash\sigma\rightarrow\tau\rangle}$. Por IH, $M:\langle(\sigma.\Gamma)_{\leq i}.nil\vdash\tau\rangle$ onde $i\geq sup(M)$. Pelo Lema 3.1.10.2 tem-se que $sup(M)=sup(\lambda.M)=0$ ou $sup(\lambda.M)=sup(M)-1$. No primeiro caso, para qualquer i dado tem-se que $M:\langle(\sigma.\Gamma)_{\leq(i+1)}.nil\vdash\tau\rangle$. Logo, por (Lambda), $M:\langle\Gamma)_{\leq i}.nil\vdash\sigma\rightarrow\tau\rangle$ onde $i\geq sup(M)=sup(\lambda.M)$. No último caso, por (Lambda), $\lambda.M:\langle\Gamma_{\leq(i-1)}.nil\vdash\sigma\rightarrow\tau\rangle$, onde $i-1\geq sup(M)-1=sup(\lambda.M)$.
- Seja $\dfrac{M:\langle\Gamma\vdash\sigma\rightarrow\tau\rangle \qquad N:\langle\Gamma\vdash\sigma\rangle}{(M\ N):\langle\Gamma\vdash\tau\rangle}$. Por IH, $M:\langle\Gamma_{\leq i}.nil\vdash\sigma\rightarrow\tau\rangle$, $\forall i\geq sup(M)$ e $N:\langle\Gamma_{\leq i'}.nil\vdash\sigma\rangle$, $\forall i'\geq sup(N)$. Dados i e i', seja $j\geq max(i,i')$. Assim, por (App) tem-se $(M\ N):\langle\Gamma_{\leq j}\vdash\tau\rangle$ onde $j\geq max(sup(M),sup(N))=sup(M\ N)$. $\square$

A fim de provar uma lei de redundância para $\lambda_{dB}^{\rightarrow}$, o lema a seguir apresenta uma propriedade mais geral do sistema de tipos.

Lema 4.1.2 (Atualização para $\lambda_{dB}^{\rightarrow}$): Seja $M\in\Lambda_{dB}$ tal que $M:\langle\Gamma\vdash\tau\rangle$. Então, para quaisquer $\sigma\in\mathcal{S}$ e $i\in\mathbb{N}$ tem-se que $M^{+i}:\langle\Gamma_{\leq i}.\sigma.\Gamma_{>i}\vdash\tau\rangle$.

Demonstração. A prova é feita por indução estrutural em M. Note que, se $i\geq|\Gamma|$ então σ é adicionado ao final de Γ.

- $M\equiv\underline{n}$: Suponha que $\underline{n}:\langle\Gamma\vdash\tau\rangle$. Se $n\leq i$, então $\underline{n}^{+i}=\underline{n}$. A inserção de σ na $i{+}1$-ésima posição afeta o tipo apenas de índices maiores do que $\underline{i}$, logo tem-se trivialmente que $\underline{n}:\langle\Gamma_{\leq i}.\sigma.\Gamma_{>i}\vdash\tau\rangle$. Se $n>i$, então $\underline{n}^{+i}=\underline{n{+}1}$. Por $(Varn)$ i vezes tem-se que $\underline{n{-}i}:\langle\Gamma_{>i}\vdash\tau\rangle$. Assim, por $(Varn)$ aplicada $i+1$ vezes, tem-se que $\underline{n{+}1}:\langle\Gamma_{\leq i}.\sigma.\Gamma_{>i}\vdash\tau\rangle$.

- $M\equiv(M_1\ M_2)$: Suponha que $(M_1\ M_2):\langle\Gamma\vdash\tau\rangle$. Por (App), $M_1:\langle\Gamma\vdash\tau'\rightarrow\tau\rangle$ e $M_2:\langle\Gamma\vdash\tau'\rangle$. Por HI, $M_1^{+i}:\langle\Gamma_{\leq i}.\sigma.\Gamma_{>i}\vdash\tau'\rightarrow\tau\rangle$ e $M_2^{+i}:\langle\Gamma_{\leq i}.\sigma.\Gamma_{>i}\vdash\tau'\rangle$. Logo, por (App), $(M_1^{+i}\ M_2^{+i}):\langle\Gamma_{\leq i}.\sigma.\Gamma_{>i}\vdash\tau\rangle$.

- $M\equiv\lambda.M'$: Suponha que $\lambda.M':\langle\Gamma\vdash\tau\rangle$. Por $(Lambda)$, $M':\langle\tau_1.\Gamma\vdash\tau_2\rangle$, onde $\tau=\tau_1\rightarrow\tau_2$. Por HI tem-se que $M'^{+(i+1)}:\langle\tau_1.\Gamma_{\leq i}.\sigma.\Gamma_{>i}\vdash\tau_2\rangle$. Assim, por $(Lambda)$, $\lambda.M'^{+(i+1)}:\langle\Gamma_{\leq i}.\sigma.\Gamma_{>i}\vdash\tau_1\rightarrow\tau_2\rangle$. $\square$

Lema 4.1.3 (Redundância para $\lambda_{dB}^{\rightarrow}$): Seja M um termo em λ_{dB}. Se $M : \langle \Gamma \vdash \tau \rangle$, então $M : \langle \Gamma.\sigma \vdash \tau \rangle$. Assim, a regra ($\lambda_{dB}$-weak) abaixo é dedutível no sistema $\lambda_{dB}^{\rightarrow}$.

$$\frac{M : \langle \Gamma \vdash \tau \rangle}{M : \langle \Gamma.\sigma \vdash \tau \rangle} (\lambda_{dB}\text{-weak})$$

Demonstração. Pelo Lema 4.1.1.1, $|\Gamma| \geq sup(M)$. Assim, pelo Lema 3.1.11.1, $M^{+j} = M$ para qualquer $j \geq |\Gamma|$. Portanto, pelo Lema 4.1.2 para $i = m$, tem-se que $M : \langle \Gamma.\sigma \vdash \tau \rangle$, para qualquer $\sigma \in \mathcal{S}$. $\square$

Note que, caso a adição de σ no contexto Γ seja em uma posição menor do que $sup(M)$, então M^{+i} corresponderia a uma função distinta da representada pelo termo M. De fato, uma lei de redundância mais geral poderia ser introduzida tendo o valor $sup(M)$ com uma premissa adicional. Porém, a regra (λ_{dB}-weak) introduzida no Lema 4.1.3, onde os tipos são adicionados apenas ao final do contexto, é suficiente como a operação sintática relacionada a definição de tipagem principal para o $\lambda_{dB}^{\rightarrow}$.

Assim, com a regra (λdB-weak) e substituições de tipos, apresentamos uma definição para tipagem principal para o λ_{dB}-calculus com tipos simples, similar a definição de [103] para a tipagem principal de Hindley.

Definição 4.1.4 (Tipagem principal em $\lambda_{dB}^{\rightarrow}$): Uma **tipagem principal em** $\lambda_{dB}^{\rightarrow}$ de um termo M é uma tipagem $\Theta = \langle \Gamma \vdash \tau \rangle$ tal que:

1. $\lambda_{dB}^{\rightarrow} \Vdash M : \Theta$

2. Se $\lambda_{dB}^{\rightarrow} \Vdash M : \Theta'$ para alguma tipagem $\Theta' = \langle \Gamma' \vdash \tau' \rangle$, então existe uma substituição s tal que $s(\Gamma) = \Gamma'_{\leq|\Gamma|}.nil$ e $s(\tau) = \tau'$.

Observe que, dada uma tipagem principal $\langle \Gamma \vdash \tau \rangle$ de M, o contexto Γ é o contexto mais curto tal que M pode ser tipado ($|\Gamma| = sup(M)$). Em contraste com o λ-calculus com nomes, onde o contexto de uma tipagem principal de um termo M é o menor conjunto porquê atribui tipos para todas e apenas as variáveis livres de M, o contexto de uma tipagem principal em $\lambda_{dB}^{\rightarrow}$ pode ter declarações de tipos para índices que não ocorrem livremente no respectivo termo. Essas declarações extra são necessárias para manter a estrutura sequencial do contexto. Por exemplo, uma tipagem principal para $\underline{2}$ é $\langle \tau_1.\tau_2.nil \vdash \tau_2 \rangle$.

Como no caso do λ-calculus com tipos simples, a melhor maneira de assegurar que a Definição 4.1.4 é a tradução correta do conceito de PT é verificando se esta corresponde a Definição 2.2.2.

Teorema 4.1.5 (Correspondência para $\lambda_{dB}^{\to}$): Uma tipagem Θ é principal em $\lambda_{dB}^{\to}$ de acordo com a Definição 4.1.4 se, e somente se, Θ é principal em $\lambda_{dB}^{\to}$ de acordo com a Definição 2.2.2.

A prova de correspondência é similar a de [103] e será apresentada na Seção 4.4. A prova de *suficiência* usa um lema de substituição de tipos como em [49] 3A2.1(ii) e a lei de redundância do Lema 4.1.3. A prova de *necessidade* é construtiva por contradição, apresentando um contra-exemplo: dado um termo M com PT Θ, toma-se uma tipagem Θ' que não é PT de acordo com a Definição 4.1.4. A partir de M e da relação entre Θ e Θ' dada pela Definição 4.1.4, um novo termo N é construído tal que Θ' é uma tipagem de N mas Θ não a é. A principal diferença entre a prova em [103] e a apresentada aqui é a função recursiva usada para dar a N uma estrutura que explora particularidades de Θ', que precisou ser dividida em dois casos considerando a ordem dos índices a serem ligados por uma abstração durante a construção recursiva do contra-exemplo.

As seguir apresentamos um algoritmo de inferência de tipos para termos em λ_{dB}, similar ao de [4] para o λs_e, com o intuito de verificar se o sistema $\lambda_{dB}^{\to}$ tem PT de acordo com a Definição 4.1.4. Dado um $M \in \Lambda_{dB}$ qualquer, decore cada subtermo de M tendo uma nova variável de tipo subescrita e uma nova variável de contextos superescrita, obtendo um novo termo denotado por M'. Como exemplo, para o termo $\lambda.(\underline{2}\ \underline{1})$ tem-se o termo decorado $(\lambda.(\underline{2}^{\Gamma_1}_{\alpha_1}\ \underline{1}^{\Gamma_2}_{\alpha_2})^{\Gamma_3}_{\alpha_3})^{\Gamma_4}_{\alpha_4}$. Então, as regras da Tabela 4.1.1 são aplicadas a pares da forma $\langle\!\langle R, E\rangle\!\rangle$, onde R é um conjunto de termos decorados e E um conjunto de equações sobre as variáveis de tipo e contexto.

As regras de inferência na Tabela 4.1.1 estão de acordo como as regras de tipagem de $\lambda_{dB}^{\to}$. A inferência de tipo para M começa com $\langle\!\langle R_0, \varnothing\rangle\!\rangle$, onde R_0 é o conjunto dos subtermos de M'. As regras da Tabela 4.1.1 são aplicadas até obtermos o par $\langle\!\langle \varnothing, E_f\rangle\!\rangle$, onde E_f é o conjunto de equações de primeira ordem sobre as v.c.'s e as v.t.'s.

Exemplo 4.1.1: Seja $M = \lambda.(\underline{2}\ \underline{1})$. Então $M' = (\lambda.(\underline{2}^{\Gamma_1}_{\alpha_1}\ \underline{1}^{\Gamma_2}_{\alpha_2})^{\Gamma_3}_{\alpha_3})^{\Gamma_4}_{\alpha_4}$ e $R_0 = \{\underline{2}^{\Gamma_1}_{\alpha_1}, \underline{1}^{\Gamma_2}_{\alpha_2}, (\underline{2}^{\Gamma_1}_{\alpha_1}\ \underline{1}^{\Gamma_2}_{\alpha_2})^{\Gamma_3}_{\alpha_3}, (\lambda.(\underline{2}^{\Gamma_1}_{\alpha_1}\ \underline{1}^{\Gamma_2}_{\alpha_2})^{\Gamma_3}_{\alpha_3})^{\Gamma_4}_{\alpha_4}\}$. Usando as regras da Tabela 4.1.1 tem-se a redução:

(Var)	$\langle\!\langle R \cup \{\, \underline{1}_\alpha^\Gamma \,\}, E \rangle\!\rangle$	$\rightarrow$	$\langle\!\langle R, E \cup \{\Gamma = \alpha.\Gamma'\}\rangle\!\rangle$, onde Γ' é uma v.c. nova;
(Varn)	$\langle\!\langle R \cup \{\, \underline{n}_\alpha^\Gamma \,\}, E \rangle\!\rangle$	$\rightarrow$	$\langle\!\langle R, E \cup \{\Gamma = \alpha'_1.\cdots.\alpha'_{n-1}.\tau.\Gamma'\}\rangle\!\rangle$, onde $\alpha'_1, \ldots, \alpha'_{n-1}$ e Γ' são v.t.s e v.c. novas
(Lambda)	$\langle\!\langle R \cup \{(\lambda.M_{\alpha_1}^{\Gamma_1})_{\alpha_2}^{\Gamma_2}\}, E \rangle\!\rangle$	$\rightarrow$	$\langle\!\langle R, E \cup \{\alpha_2 = \alpha^* \rightarrow \alpha_1, \Gamma_1 = \alpha^*.\Gamma_2\}\rangle\!\rangle$, onde α^* é uma v.t. nova;
(App)	$\langle\!\langle R \cup \{(M_{\alpha_1}^{\Gamma_1}\ N_{\alpha_2}^{\Gamma_2})_{\alpha_3}^{\Gamma_3}\}, E \rangle\!\rangle$	$\rightarrow$	$\langle\!\langle R, E \cup \{\Gamma_1 = \Gamma_2, \Gamma_2 = \Gamma_3, \alpha_1 = \alpha_2 \rightarrow \alpha_3\}\rangle\!\rangle$

Tabela 4.1.1: Regras de inferência de tipos para o λ_{dB}-calculus

$$\begin{aligned}
&\langle\!\langle R_0, \emptyset \rangle\!\rangle \rightarrow_{\text{Varn}} \\
&\langle\!\langle R_1 = R_0 \setminus \{\, \underline{2}_{\alpha_1}^{\Gamma_1}\}, E_1 = \{\Gamma_1 = \alpha'_1.\alpha_1.\Gamma'_1\}\rangle\!\rangle \rightarrow_{\text{Var}} \\
&\langle\!\langle R_2 = R_1 \setminus \{\, \underline{1}_{\alpha_2}^{\Gamma_2}\}, E_2 = E_1 \cup \{\Gamma_2 = \alpha_2.\Gamma'_2\}\rangle\!\rangle \rightarrow_{\text{App}} \\
&\langle\!\langle R_3 = R_2 \setminus \{(\, \underline{2}_{\alpha_1}^{\Gamma_1}\ \ \underline{1}_{\alpha_2}^{\Gamma_2})_{\alpha_3}^{\Gamma_3}\}, E_3 = E_2 \cup \{\Gamma_1 = \Gamma_2, \Gamma_2 = \Gamma_3, \alpha_1 = \alpha_2 {\rightarrow} \alpha_3\}\rangle\!\rangle \rightarrow_{\text{Lambda}} \\
&\langle\!\langle \emptyset = R_3 \setminus \{(\lambda.(\, \underline{2}_{\alpha_1}^{\Gamma_1}\ \ \underline{1}_{\alpha_2}^{\Gamma_2})_{\alpha_3}^{\Gamma_3})_{\alpha_4}^{\Gamma_4}\}, E_4 = E_3 \cup \{\alpha_4 = \alpha_1^* {\rightarrow} \alpha_3, \Gamma_3 = \alpha_1^*.\Gamma_4\}\rangle\!\rangle
\end{aligned}$$

Logo, $E_4 = E_f$. Resolvendo as equações triviais sobre v.c., i.e. $\Gamma_1 = \Gamma_2 = \Gamma_3$, e tomando as variáveis com o menor subindice, tem-se $\{\alpha_1 = \alpha_2 {\rightarrow} \alpha_3, \alpha_4 = \alpha_1^* {\rightarrow} \alpha_3, \Gamma_1 = \alpha'_1.\alpha_1.\Gamma'_1, \Gamma_1 = \alpha_2.\Gamma'_2, \Gamma_1 = \alpha_1^*.\Gamma_4\}$. Simplificando, $\{\alpha_1 = \alpha_2 {\rightarrow} \alpha_3, \alpha_4 = \alpha_1^* {\rightarrow} \alpha_3, \alpha'_1.\alpha_1.\Gamma'_1 = \alpha_2.\Gamma'_2 = \alpha_1^*.\Gamma_4\}$. A partir dessas equações obtém-se o unificador mais geral $\alpha_4 = \alpha_2 {\rightarrow} \alpha_3$ e $\Gamma_4 = (\alpha_2 {\rightarrow} \alpha_3).\Gamma'_1$, para as variáveis de interesse. Como o contexto deve ser o mais curto possível, $\Gamma'_1 = nil$ e $\langle (\alpha_2 {\rightarrow} \tau_3).nil \vdash \alpha_2 {\rightarrow} \tau_3 \rangle$ é uma tipagem principal de M.

A partir da Definição 4.1.4 e pela unicidade das soluções retornadas pelo algoritmo de inferência de tipos, deduz-se que $\lambda_{dB}^{\rightarrow}$ satisfaz a propriedade de PT. O teorema a seguir afirma que todo termo tipável em $\lambda_{dB}^{\rightarrow}$ tem uma tipagem principal.

Teorema 4.1.6 (PT para o $\lambda_{dB}^{\rightarrow}$): O sistema $\lambda_{dB}^{\rightarrow}$ satisfaz a propriedade de tipagem principal.

Demonstração. Seja $M \in \Lambda_{dB}$ qualquer e M' sua versão decorada.

Seja R_0 o conjunto de todos os subtermos de M'. Iniciando com o par $\langle\!\langle R_0, \emptyset \rangle\!\rangle$ e aplicando as regras de inferência da Tabela 4.1.1 obtém-se um par final após um número finito de passos, pois após cada aplicação o número de elementos do conjunto de subtermos decorados é decrementado. Pela unicidade na decomposição em subtermos, só é possível aplicar uma única regra em cada elemento de R_0. Logo, o processo termina com um par $\langle\!\langle \emptyset, E_f \rangle\!\rangle$, onde E_f é um conjunto de equações de primeira ordem sobre variáveis de contexto e de tipo, que estão de acordo com a regras de tipagem de $\lambda_{dB}^{\rightarrow}$. Um algoritmo de unificação de primeira ordem adequado, *e.g.* veja [85], é então aplicado. Pela correção, completitude e unicidade da unificação de primeira ordem, tem-se que o algoritmo vai retornar um mgu se M é tipável. Caso contrário, o algoritmo retorna que não existe um unificador. Consequentemente, o sistema $\lambda_{dB}^{\rightarrow}$ satisfaz PT. $\square$

4.2 Tipagem principal para o $\lambda s_e{}^{\rightarrow}$, o λs_e com tipos simples

A Figura 4.2.1 contém as regras de $\lambda s_e{}^{\rightarrow}$, introduzidas na Definição 3.2.14.

$$
\begin{array}{llll}
\text{(Var)} & \dfrac{}{\underline{1} : \langle \tau.\Gamma \vdash \tau \rangle} & \text{(Varn)} & \dfrac{\underline{n} : \langle \Gamma \vdash \tau \rangle}{\underline{n+1} : \langle \sigma.\Gamma \vdash \tau \rangle} \\
\text{(Lambda)} & \dfrac{M : \langle \sigma.\Gamma \vdash \tau \rangle}{\lambda.M : \langle \Gamma \vdash \sigma \rightarrow \tau \rangle} & \text{(App)} & \dfrac{M : \langle \Gamma \vdash \sigma \rightarrow \tau \rangle \qquad N : \langle \Gamma \vdash \sigma \rangle}{(M\ N) : \langle \Gamma \vdash \tau \rangle}
\end{array}
$$

$$
\text{(Phi)} \qquad \frac{M : \langle \Gamma_{\leq k}.\Gamma_{\geq k+i} \vdash \tau \rangle}{\varphi_k^i M : \langle \Gamma \vdash \tau \rangle}, \qquad \text{onde } |\Gamma| \leq k+i-1
$$

$$
\text{(Sigma)} \quad \frac{N : \langle \Gamma_{\geq i} \vdash \rho \rangle \qquad M : \langle \Gamma_{<i}.\rho.\Gamma_{\geq i} \vdash \tau \rangle}{M \sigma^i N : \langle \Gamma \vdash \tau \rangle}, \quad \text{onde } |\Gamma| \geq i-1
$$

Figura 4.2.1: O sistema $\lambda s_e{}^{\rightarrow}$

A lei de redundância para o $\lambda s_e{}^{\rightarrow}$ é da mesma forma que para o $\lambda_{dB}^{\rightarrow}$, adicionando tipos ao final do contexto, de onde tem-se o seguinte lema.

Lema 4.2.1 (Redundância para $\lambda s_e{}^{\rightarrow}$): A regra ($\lambda s_e$-weak) abaixo é dedutível no sistema $\lambda s_e{}^{\rightarrow}$.

$$
\frac{M : \langle \Gamma \vdash \tau \rangle}{M : \langle \Gamma.\sigma \vdash \tau \rangle} (\lambda s_e\text{-weak}).
$$

Demonstração. Indução na estrutura de M. Seja $\sigma \in \mathcal{S}$, qualquer.

- $M \equiv \underline{n}$: Seja $\underline{n} : \langle \Gamma \vdash \tau \rangle$. Como a adição de tipos ao final de Γ não altera a designação de tipo de nenhum índice livre, tem-se trivialmente que $\underline{n} : \langle \Gamma.\sigma \vdash \tau \rangle$.

- $M \equiv (M_1\ M_2)$: Seja $(M_1\ M_2) : \langle \Gamma \vdash \tau \rangle$. Por (App), $M_1 : \langle \Gamma \vdash \rho \rightarrow \tau \rangle$ e $M_2 : \langle \Gamma \vdash \rho \rangle$, para algum $\rho \in \mathcal{S}$. Por HI, $M_1 : \langle \Gamma.\sigma \vdash \rho \rightarrow \tau \rangle$ e $M_2 : \langle \Gamma.\sigma \vdash \rho \rangle$, logo, por (App),

$(M_1\ M_2) : \langle \Gamma.\sigma \vdash \tau \rangle$.

- $M \equiv \lambda.N$: Seja $\lambda.N : \langle \Gamma \vdash \tau \rangle$. Por (Lambda), $N : \langle \tau_1.\Gamma \vdash \tau_2 \rangle$, onde $\tau = \tau_1 \rightarrow \tau_2$. Por HI, $N : \langle \tau_1.\Gamma.\sigma \vdash \tau_2 \rangle$, logo, por (Lambda), $\lambda.N : \langle \Gamma.\sigma \vdash \tau \rangle$

- $M \equiv M_1 \sigma^i M_2$: Seja $M_1 \sigma^i M_2 : \langle \Gamma \vdash \tau \rangle$. Por (Sigma), $M_2 : \langle \Gamma_{\geq i} \vdash \rho \rangle$ e $M_1 : \langle \Gamma_{<i}.\rho.\Gamma_{\geq i} \vdash \tau \rangle$, para algum $\rho \in \mathcal{S}$. Por HI, $M_2 : \langle \Gamma_{\geq i}.\sigma \vdash \rho \rangle$ e $M_1 : \langle \Gamma_{<i}.\rho.\Gamma_{\geq i}.\sigma \vdash \tau \rangle$, logo, por (Sigma), $M_1 \sigma^i M_2 : \langle \Gamma.\sigma \vdash \tau \rangle$.

- $M \equiv \varphi^i_k N$: Seja $\varphi^i_k N : \langle \Gamma \vdash \tau \rangle$. Por (Phi), $N : \langle \Gamma_{\leq k}.\Gamma_{\geq k+i} \vdash \tau \rangle$. Por HI, $N : \langle \Gamma_{\leq k}.\Gamma_{\geq k+i}.\sigma \vdash \tau \rangle$, logo, por (Phi), $\varphi^i_k N : \langle \Gamma.\sigma \vdash \tau \rangle$. □

Assim, a definição de tipagem principal em $\lambda s_e{}^{\rightarrow}$ é a mesma de $\lambda_{dB}^{\rightarrow}$.

Definição 4.2.2 (Tipagem principal em $\lambda s_e{}^{\rightarrow}$): Uma **tipagem principal** de um termo M em $\lambda s_e{}^{\rightarrow}$ é uma tipagem $\Theta = \langle \Gamma \vdash \tau \rangle$ tal que:

1. $\lambda s_e{}^{\rightarrow} \Vdash M : \Theta$

2. Se $\lambda s_e{}^{\rightarrow} \Vdash M : \Theta'$ para alguma tipagem $\Theta' = \langle \Gamma' \vdash \tau' \rangle$, então existe uma substituição s tal que $s(\Gamma) = \Gamma'_{\leq|\Gamma|}.nil$ e $s(\tau) = \tau'$.

Teorema 4.2.3 (Correspondência para $\lambda s_e{}^{\rightarrow}$): Uma tipagem Θ é principal em $\lambda s_e{}^{\rightarrow}$ de acordo com a Definição 4.2.2 se, e somente se, Θ é principal em $\lambda s_e{}^{\rightarrow}$ de acordo com a Definição 2.2.2.

A prova do Teorema 4.2.3 é uma extensão direta da prova do Teorema 4.1.5, sendo apresentada na Seção 4.4.

A seguir apresentamos o algoritmo de inferência de tipos para o λs_e-calculus, similar ao introduzido em [4]. O algoritmo é composto das regras da Tabela 4.1.1 acrescentado das regras da Tabela 4.2.1.

Analogamente ao algoritmo anterior, as regras da Tabela 4.2.1 foram desenvolvidas de acordo com as regras de tipagem da Figura 3.2.14. O termo decorado associado ao termo M, denotado por M', tem uma sintaxe próxima a dos termo decorados em λ_{dB}: todo subtermo de M é decorado com uma variável de tipo e de contexto. As regras são então aplicadas a pares na forma $\langle\!\langle R, E \rangle\!\rangle$, iniciando de $\langle\!\langle R_0, \varnothing \rangle\!\rangle$, como feito para o $\lambda_{dB}^{\rightarrow}$.

(Sigma)	$\langle\!\langle R \cup \{(M^{\Gamma_1}_{\alpha_1} \sigma^i N^{\Gamma_2}_{\alpha_2})^{\Gamma_3}_{\alpha_3}\}, E\rangle\!\rangle \rightarrow$ $\langle\!\langle R, E \cup \{\alpha_1{=}\alpha_3, \Gamma_1{=}\alpha'_1.\cdots.\alpha'_{i-1}.\alpha_2.\Gamma_2, \Gamma_3{=}\alpha'_1.\cdots.\alpha'_{i-1}.\Gamma_2\}\rangle\!\rangle$, onde $\alpha'_1, \ldots, \alpha'_{i-1}$ são v.t. novas e a sequência é vazia se $i = 1$;
(Phi)	$\langle\!\langle R \cup \{(\varphi^i_k M^{\Gamma_1}_{\alpha_1})^{\Gamma_2}_{\alpha_2}\}, E\rangle\!\rangle \rightarrow$ $\langle\!\langle R, E \cup \{\alpha_1 = \alpha_2, \Gamma_2 = \alpha'_1.\cdots.\alpha'_{k+i-1}.\Gamma', \Gamma_1 = \alpha'_1.\cdots.\alpha'_k.\Gamma'\}\rangle\!\rangle$, onde Γ' e $\alpha'_1, \ldots, \alpha'_{k+i-1}$ são v.c. e v.t.s novas. Se $k{+}i{-}1{=}0$ ou $k{=}0$, então as sequências $\alpha'_1, \ldots, \alpha'_{k+i-1}$ e $\alpha'_1, \ldots, \alpha'_k$, respectivamente, são vazias

Tabela 4.2.1: Regras de inferência de tipos para o λs_e-calculus

Exemplo 4.2.1: Para $M = \lambda.((\underline{1}\,\sigma^2\,\underline{2})\ (\varphi^2_0\ \underline{2}))$, obtém-se o R_0 correspondente ao $M' = (\lambda.((\underline{1}^{\Gamma_1}_{\alpha_1}\sigma^2\,\underline{2}^{\Gamma_2}_{\alpha_2})^{\Gamma_3}_{\alpha_3}\ (\varphi^2_0\,\underline{2}^{\Gamma_4}_{\alpha_4})^{\Gamma_5}_{\alpha_5})^{\Gamma_6}_{\alpha_6})^{\Gamma_7}_{\alpha_7}$. Então, aplicando as regras das Tabelas 4.1.1 e 4.2.1 ao par $\langle\!\langle R_0, \emptyset\rangle\!\rangle$, obtém-se o par $\langle\!\langle \emptyset, E_f\rangle\!\rangle$. Simplificando E_f, similarmente ao executado no Exemplo 4.1.1, tem-se o sistema de equações a partir do qual o m.g.u $\alpha_7 = (\alpha_2 \rightarrow \alpha_6) \rightarrow \alpha_6$ e $\Gamma_7 = \alpha'_1.\alpha_2.\Gamma'_2$ são obtidos, para as variáveis de interesse.

Teorema 4.2.4 (PT para o $\lambda s_e{}^{\rightarrow}$): O sistema $\lambda s_e{}^{\rightarrow}$ satisfaz a propriedade de tipagem principal.

Demonstração. Análoga a prova do Teorema 4.1.6, com aplicação das Tabelas 4.1.1 e 4.2.1. □

4.3 Tipagem principal para o $\lambda\sigma^{\rightarrow}$, o $\lambda\sigma$ com tipos simples

A Figura 4.3.1 contém as regras de inferência de λs_e, como apresentas na Definição 3.3.9.

O conceito de tipagem para $\lambda\sigma^{\rightarrow}$ tem que ser adaptado porque a expressão da classe de substituições é decorada com variáveis de contexto como seus respectivos tipos e contextos. Assim, diz-se que $\Theta = \langle\Gamma \blacktriangleright \mathbb{T}\rangle$ é uma tipagem de uma expressão em $\lambda\sigma^{\rightarrow}$, onde $\mathbb{T}$ pode ser tanto um tipo quanto um contexto. Se a expressão analisada corresponde a um termo do λ-calculus, a noção de tipagem corresponde a noção em $\lambda^{\rightarrow}_{dB}$.

Para a introdução de uma lei de redundância para o $\lambda\sigma^{\rightarrow}$, algumas definições e lemas auxiliares são necessários.

$$
\begin{array}{llll}
\text{(var)} & \dfrac{}{\underline{1}:\langle\tau.\Gamma\vdash\tau\rangle} & \text{(lambda)} & \dfrac{M:\langle\sigma.\Gamma\vdash\tau\rangle}{\lambda.M:\langle\Gamma\vdash\sigma\rightarrow\tau\rangle} \\[2ex]
\text{(app)} & \dfrac{M_1:\langle\Gamma\vdash\sigma\rightarrow\tau\rangle \qquad M_2:\langle\Gamma\vdash\sigma\rangle}{(M_1\ M_2):\langle\Gamma\vdash\tau\rangle} & \text{(clos)} & \dfrac{S:\langle\Gamma\triangleright\Gamma'\rangle \qquad M:\langle\Gamma'\vdash\tau\rangle}{M[S]:\langle\Gamma\vdash\tau\rangle} \\[2ex]
\text{(id)} & \dfrac{}{id:\langle\Gamma\triangleright\Gamma\rangle} & \text{(shift)} & \dfrac{}{\uparrow:\langle\tau.\Gamma\triangleright\Gamma\rangle} \\[2ex]
\text{(cons)} & \dfrac{M:\langle\Gamma\vdash\tau\rangle \qquad S:\langle\Gamma\triangleright\Gamma'\rangle}{M.S:\langle\Gamma\triangleright\tau.\Gamma'\rangle} & \text{(comp)} & \dfrac{S:\langle\Gamma\triangleright\Gamma'\rangle \qquad S':\langle\Gamma'\triangleright\Gamma''\rangle}{S'\circ S:\langle\Gamma\triangleright\Gamma''\rangle}
\end{array}
$$

Figura 4.3.1: O sistema $\lambda\sigma^{\rightarrow}$

Definição 4.3.1: Seja M uma expressão em $\lambda\sigma$. Seja $\|\cdot\|:\Lambda_\sigma\rightarrow\mathbb{N}$ tal que:

$$
\begin{array}{rcl@{\qquad\qquad}rcl}
\|(M\ N)\| & = & \|M\|+\|N\| & \|\underline{1}\| & = & 0 \\
\|\lambda.M\| & = & \|M\| & \|id\| & = & 0 \\
\|M[S]\| & = & \|M\|+\|S\| & \|\uparrow\| & = & 0 \\
\|S\circ T\| & = & \|S\|+\|T\| & \|M.S\| & = & 1+\|M\|+\|S\|
\end{array}
$$

Lema 4.3.2: Seja S uma substituição em $\lambda\sigma$. Se $\|S\|=0$ e $S:\langle\Gamma\triangleright\Gamma'\rangle$, então $S:\langle\Gamma.\sigma\triangleright\Gamma'.\sigma\rangle$.

Demonstração. Por indução na estrutura de S, onde $\|S\|=0$.

- $S\equiv id$: Por (id) tem-se trivialmente que $id:\langle\Gamma.\sigma\triangleright\Gamma'.\sigma\rangle$.
- $S\equiv\uparrow$: Seja $\uparrow:\langle\Gamma\triangleright\Gamma'\rangle$ onde, por (shift), $\Gamma=\tau.\Gamma'$. Assim, $\uparrow:\langle\Gamma.\sigma\triangleright\Gamma'.\sigma\rangle$.
- $S\equiv S_1\circ S_2$: Seja $S_1\circ S_2:\langle\Gamma\triangleright\Gamma'\rangle$. Por (comp) tem-se que $S_2:\langle\Gamma\triangleright\Gamma''\rangle$ e $S_1:\langle\Gamma''\triangleright\Gamma'\rangle$, para algum contexto Γ''. Por HI tem-se $S_2:\langle\Gamma.\sigma\triangleright\Gamma''.\sigma\rangle$ e $S_1:\langle\Gamma''.\sigma\triangleright\Gamma'.\sigma\rangle$, logo, por (comp), $S_1\circ S_2:\langle\Gamma.\sigma\triangleright\Gamma'.\sigma\rangle$. □

Lema 4.3.3: Seja M um termo em $\lambda\sigma$. Se $\|M\|=0$ e $M:\langle\Gamma\vdash\tau\rangle$, então $M:\langle\Gamma.\sigma\vdash\tau\rangle$.

Demonstração. Por indução na estrutura de M, onde $\|M\|=0$.

- $M\equiv\underline{1}$: Seja $\underline{1}:\langle\Gamma\vdash\tau\rangle$. Por (var) tem-se que $\Gamma=\tau.\Gamma'$, para qualquer contexto Γ'. Logo, tem-se trivialmente que $\underline{1}:\langle\Gamma.\sigma\vdash\tau\rangle$.

- $M \equiv (M_1\ M_2)$: Seja $(M_1\ M_2):\langle\Gamma \vdash \tau\rangle$. Por (app), $M_1:\langle\Gamma \vdash \rho\to\tau\rangle$ e $M_2:\langle\Gamma \vdash \rho\rangle$ para algum $\rho\in\mathcal{S}$. Por HI, $M_1:\langle\Gamma.\sigma \vdash \rho\to\tau\rangle$ e $M_2:\langle\Gamma.\sigma \vdash \rho\rangle$ logo, por (app), $(M_1\ M_2):\langle\Gamma.\sigma \vdash \tau\rangle$.

- $M \equiv \lambda.N$: Seja $\lambda.N:\langle\Gamma \vdash \tau\rangle$. Por (lambda), $N:\langle\rho.\Gamma \vdash \tau'\rangle$, onde $\tau=\rho\to\tau'$. Por HI, $N:\langle\rho.\Gamma.\sigma \vdash \tau'\rangle$ logo, por (lambda), $\lambda.N:\langle\Gamma.\sigma \vdash \tau\rangle$.

- $M \equiv N[S]$: Seja $N[S]:\langle\Gamma \vdash \tau\rangle$. Por (clos), $S:\langle\Gamma \rhd \Gamma'\rangle$ e $N:\langle\Gamma' \vdash \tau\rangle$, para algum contexto Γ'. Como $\|N[S]\| = \|N\| + \|S\| = 0$, pelo Lema 4.3.2, $S:\langle\Gamma.\sigma \rhd \Gamma'.\sigma\rangle$ e, por HI, $N:\langle\Gamma'.\sigma \vdash \tau\rangle$. Assim, por (clos), $N[S]:\langle\Gamma.\sigma \vdash \tau\rangle$. □

Lema 4.3.4 (Redundância para $\lambda\sigma^{\to}$): Sejam M um termo e S uma substituição, ambos em $\lambda\sigma$, e $\sigma\in\mathcal{S}$, qualquer. Se $M:\langle\Gamma \vdash \tau\rangle$, então $M:\langle\Gamma.\sigma \vdash \tau\rangle$. Similarmente, se $S:\langle\Gamma \rhd \Gamma'\rangle$, então $S:\langle\Gamma.\sigma \rhd \Gamma'.\sigma\rangle$. Portanto, as regras ($\lambda\sigma$-tweak) e ($\lambda\sigma$-sweak) abaixo são dedutíveis no sistema $\lambda\sigma^{\to}$.

$$\frac{M:\langle\Gamma \vdash \tau\rangle}{M:\langle\Gamma.\sigma \vdash \tau\rangle}(\lambda\sigma\text{-tweak}) \qquad\qquad \frac{S:\langle\Gamma \rhd \Gamma'\rangle}{S:\langle\Gamma.\sigma \rhd \Gamma'.\sigma\rangle}(\lambda\sigma\text{-sweak})$$

Demonstração. Por indução na estrutura de M, com subindução em $\|\cdot\|$, tendo os Lemas 4.3.2 e 4.3.3 como base de indução (BI).

- $M \equiv \underline{1}$: Seja $\underline{1}:\langle\Gamma \vdash \tau\rangle$. Pelo Lema 4.3.3, $\underline{1}:\langle\Gamma.\sigma \vdash \tau\rangle$.

- $M \equiv (M_1\ M_2)$: Seja $(M_1\ M_2):\langle\Gamma \vdash \tau\rangle$. Por (app) tem-se que $M_1:\langle\Gamma \vdash \rho\to\tau\rangle$ e $M_2:\langle\Gamma \vdash \rho\rangle$, para algum $\rho\in\mathcal{S}$. Por IH na estrutura tem-se $M_1:\langle\Gamma.\sigma \vdash \rho\to\tau\rangle$ e $M_2:\langle\Gamma.\sigma \vdash \rho\rangle$ logo, por (app), $(M_1\ M_2):\langle\Gamma.\sigma \vdash \tau\rangle$.

- $M \equiv \lambda.N$: Seja $\lambda.N:\langle\Gamma \vdash \tau\rangle$. Por (lambda), $N:\langle\rho.\Gamma \vdash \tau'\rangle$, onde $\tau=\rho\to\tau'$. Por HI, $N:\langle\rho.\Gamma.\sigma \vdash \tau'\rangle$ logo, por (lambda), $\lambda.N:\langle\Gamma.\sigma \vdash \tau\rangle$.

- $M \equiv N[S]$: Seja $N[S]:\langle\Gamma \vdash \tau\rangle$. Por (clos), $S:\langle\Gamma \rhd \Gamma'\rangle$ e $N:\langle\Gamma' \vdash \tau\rangle$, para algum contexto Γ'. Por HI, $N:\langle\Gamma'.\sigma \vdash \tau\rangle$. A substituição S deve ser analisada. Se $\|N\| > 0$, então $\|N[S]\| > \|S\|$, logo por IH em $\|\cdot\|$ tem-se que $S:\langle\Gamma.\sigma \rhd \Gamma'.\sigma\rangle$. Senão, para $\|N\| = 0$ tem-se:

  - Se $\|S\| = 0$, então o Lema 4.3.2 pode ser aplicado.

- Caso contrário, $S \equiv N'.S'$ ou $S \equiv S_1 \circ S_2$. Se $S \equiv N'.S'$ então por (cons) tem-se que $N' : \langle \Gamma \vdash \rho \rangle$ e $S' : \langle \Gamma \rhd \Gamma'' \rangle$, onde $\Gamma' = \rho.\Gamma''$. Como $\|N'\|, \|S'\| < \|S\| = \|N[S]\|$, por HI em $\| \cdot \|$ tem-se que $N' : \langle \Gamma.\sigma \vdash \rho \rangle$ e $S' : \langle \Gamma.\sigma \rhd \Gamma''.\sigma \rangle$. Assim, por (cons), $N'.S' : \langle \Gamma.\sigma \rhd \Gamma'.\sigma \rangle$. Se $S \equiv S_1 \circ S_2$, então por (comp) tem-se que $S_2 : \langle \Gamma \rhd \Gamma'' \rangle$ e $S_1 : \langle \Gamma'' \rhd \Gamma' \rangle$, para algum contexto Γ''. Se $\|S_1\|, \|S_2\| > 0$ então o resultado vale, por indução em $\| \cdot \|$. Caso contrário, pelo menos uma das substituições tem $\| \cdot \|$ maior do que 0. Usando indução na estrutura de S, onde $\|S\| > 0$, o resultado é valido. Assim, $S_2 : \langle \Gamma.\sigma \rhd \Gamma''.\sigma \rangle$ e $S_1 : \langle \Gamma''.\sigma \rhd \Gamma'.\sigma \rangle$ logo, por (comp), $S_1 \circ S_2 : \langle \Gamma.\sigma \rhd \Gamma'.\sigma \rangle$.

Finalmente, por (clos) tem-se que $N[S] : \langle \Gamma.\sigma \vdash \tau \rangle$. □

O Lema 4.3.4 acima e as substituições de tipos nos permitem apresentar uma definição de PT em $\lambda\sigma^{\rightarrow}$.

Definição 4.3.5 (Tipagem principal em $\lambda\sigma^{\rightarrow}$): Uma **tipagem principal** de uma expressão M em $\lambda\sigma^{\rightarrow}$ é uma tipagem $\Theta = \langle \Gamma \blacktriangleright \mathbb{T} \rangle$ tal que:

1. $\lambda\sigma^{\rightarrow} \Vdash M : \Theta$
2. Se $\lambda\sigma^{\rightarrow} \Vdash M : \Theta'$ para alguma tipagem $\Theta' = \langle \Gamma' \blacktriangleright \mathbb{T}' \rangle$, então existe uma substituição s tal que $s(\Gamma) = \Gamma'_{\leq|\Gamma|}.nil$ e se $\mathbb{T}$ é um tipo, $s(\mathbb{T}) = \mathbb{T}'$, senão $s(\mathbb{T}) = \mathbb{T}'_{\leq|\mathbb{T}|}.nil$.

Deve-se então verificar se esta definição de PT para $\lambda\sigma^{\rightarrow}$ tem correspondência com a definição independente de sistemas de Wells em [103].

Teorema 4.3.6 (Correspondência para $\lambda\sigma^{\rightarrow}$): Uma tipagem Θ é principal em $\lambda\sigma^{\rightarrow}$ de acordo com a Definição 4.3.5 se, e somente se, Θ é principal em $\lambda\sigma^{\rightarrow}$ de acordo com a Definição 2.2.2.

A despeito do conceito de tipagem ter sido estendido a fim de incluir a classe de substituições, a técnica usada para provar o Teorema 4.3.6 coincide com as técnicas aplicadas nas provas dos Teoremas 4.1.5 e 4.2.3. Assim, as três provas são amalgamadas e apresentadas na Seção 4.4.

Apresentamos a seguir o algoritmo de inferência de tipos, para verificar se $\lambda\sigma^{\rightarrow}$ tem PT de acordo com a Definição 4.3.5. Assim, dado uma expressão M, trabalharemos com sua versão decorada M' onde o tipo das substituições também é um contexto. A mesma sintaxe de [14] para expressões decoradas é adotada.

(Var)	$\langle\!\langle R \cup \{\, \underline{1}^{\Gamma}_{\alpha} \,\}, E \rangle\!\rangle$	$\rightarrow$	$\langle\!\langle R, E \cup \{\Gamma = \alpha.\Gamma'\} \rangle\!\rangle$, onde Γ' é v.c. nova;
(Lambda)	$\langle\!\langle R \cup \{(\lambda.M^{\Gamma_1}_{\alpha_1})^{\Gamma_2}_{\alpha_2}\}, E \rangle\!\rangle$	$\rightarrow$	$\langle\!\langle R, E \cup \{\alpha_2 = \alpha^* \rightarrow \alpha_1, \Gamma_1 = \alpha^*.\Gamma_2\} \rangle\!\rangle$, onde α^* é v.t. nova;
(App)	$\langle\!\langle R \cup \{(M^{\Gamma_1}_{\alpha_1}\ N^{\Gamma_2}_{\alpha_2})^{\Gamma_3}_{\alpha_3}\}, E \rangle\!\rangle$	$\rightarrow$	$\langle\!\langle R, E \cup \{\Gamma_1 = \Gamma_2, \Gamma_2 = \Gamma_3, \alpha_1 = \alpha_2 \rightarrow \alpha_3\} \rangle\!\rangle$
(Clos)	$\langle\!\langle R \cup \{(M^{\Gamma_1}_{\alpha_1}[S^{\Gamma_2}_{\Gamma_3}])^{\Gamma_4}_{\alpha_2}\}, E \rangle\!\rangle$	$\rightarrow$	$\langle\!\langle R, E \cup \{\Gamma_1 = \Gamma_3, \Gamma_2 = \Gamma_4, \alpha_1 = \alpha_2\} \rangle\!\rangle$
(Id)	$\langle\!\langle R \cup \{id^{\Gamma_1}_{\Gamma_2}\}, E \rangle\!\rangle$	$\rightarrow$	$\langle\!\langle R, E \cup \{\Gamma_1 = \Gamma_2\} \rangle\!\rangle$
(Shift)	$\langle\!\langle R \cup \{\uparrow^{\Gamma_1}_{\Gamma_2}\}, E \rangle\!\rangle$	$\rightarrow$	$\langle\!\langle R, E \cup \{\Gamma_1 = \alpha'.\Gamma_2\} \rangle\!\rangle$, onde α' é v.t. nova;
(Cons)	$\langle\!\langle R \cup \{(M^{\Gamma_1}_{\alpha_1}.S^{\Gamma_2}_{\Gamma_3})^{\Gamma_4}_{\Gamma_5}\}, E \rangle\!\rangle$	$\rightarrow$	$\langle\!\langle R, E \cup \{\Gamma_1 = \Gamma_2, \Gamma_2 = \Gamma_4, \Gamma_5 = \alpha_1.\Gamma_3\} \rangle\!\rangle$
(Comp)	$\langle\!\langle R \cup \{(S^{\Gamma_1}_{\Gamma_2} \circ S'^{\Gamma_3}_{\Gamma_4})^{\Gamma_5}_{\Gamma_6}\}, E \rangle\!\rangle$	$\rightarrow$	$\langle\!\langle R, E \cup \{\Gamma_1 = \Gamma_4, \Gamma_2 = \Gamma_6, \Gamma_3 = \Gamma_5\} \rangle\!\rangle$

Tabela 4.3.1: Regras de inferência de tipo para o $\lambda\sigma$-calculus

As regras de inferência na Tabela 4.3.1 estão de acordo com as regras de tipagem do sistema $\lambda\sigma^{\rightarrow}$, apresentadas na Figura 3.3.9.

Analogamente aos algoritmos anteriores, as regras da Tabela 4.3.1 são aplicadas a pares $\langle\!\langle R, E \rangle\!\rangle$, onde R é um conjunto de subexpressões de M' e E um conjunto de equações sobre as variáveis de tipo e contexto.

Exemplo 4.3.1: Para $M \equiv (\,\underline{2}\,.id) \circ \uparrow$ tem-se que $M' = (((\,\underline{1}^{\Gamma_1}_{\alpha_1}[\uparrow^{\Gamma_2}_{\Gamma_3}])^{\Gamma_4}_{\alpha_2}.id^{\Gamma_5}_{\Gamma_6})^{\Gamma_7}_{\Gamma_8} \circ \uparrow^{\Gamma_9}_{\Gamma_{10}})^{\Gamma_{11}}_{\Gamma_{12}}$. Então, $R_0 = \{(\,\underline{1}^{\Gamma_1}_{\alpha_1}[\uparrow^{\Gamma_2}_{\Gamma_3}])^{\Gamma_4}_{\alpha_2}, ((\,\underline{1}^{\Gamma_1}_{\alpha_1}[\uparrow^{\Gamma_2}_{\Gamma_3}])^{\Gamma_4}_{\alpha_2}.id^{\Gamma_5}_{\Gamma_6})^{\Gamma_7}_{\Gamma_8}, (((\,\underline{1}^{\Gamma_1}_{\alpha_1}[\uparrow^{\Gamma_2}_{\Gamma_3}])^{\Gamma_4}_{\alpha_2}.id^{\Gamma_5}_{\Gamma_6})^{\Gamma_7}_{\Gamma_8} \circ \uparrow^{\Gamma_9}_{\Gamma_{10}})^{\Gamma_{11}}_{\Gamma_{12}}, \underline{1}^{\Gamma_1}_{\alpha_1}, \uparrow^{\Gamma_2}_{\Gamma_3}, \uparrow^{\Gamma_9}_{\Gamma_{10}}$ $, id^{\Gamma_5}_{\Gamma_6}\}$. Aplica-se as regras da Tabela 4.3.1 ao par $\langle\!\langle R_0, \emptyset \rangle\!\rangle$ até obter o par $\langle\!\langle \emptyset, E_f \rangle\!\rangle$. Simplificando E_f a exemplo do feito no Exemplo 4.1.1, obtém-se o conjunto de equações $\{\alpha_1 = \alpha_2, \Gamma_{11} = \Gamma_{12} = \alpha_2.\Gamma_2, \Gamma_2 = \alpha'_1.\Gamma_1, \Gamma_1 = \alpha_1.\Gamma'_1\}$. A partir deste sistema de equações tem-se o m.g.u. $\Gamma_{11} = \Gamma_{12} = \alpha_1.\alpha'_1.\alpha_1.\Gamma'_1$, para as variáveis de interesse. Assim, $\langle \alpha_1.\alpha'_1.\alpha_1.nil \rhd \alpha_1.\alpha'_1.\alpha_1.nil \rangle$ é uma tipagem principal de M em $\lambda\sigma^{\rightarrow}$.

Teorema 4.3.7 (PT para o $\lambda\sigma^{\rightarrow}$): O sistema $\lambda\sigma^{\rightarrow}$ satisfaz a propriedade de tipagem principal.

Demonstração. Análoga à prova do Teorema 4.1.6, com aplicação da Tabela 4.3.1. □

4.4 Prova de correspondência

Nesta seção será apresentada uma prova que combina as provas de correspondência entres as definições de tipagem principal específicas dos sistemas $\lambda_{dB}^{\rightarrow}$, $\lambda s_e{}^{\rightarrow}$ e $\lambda\sigma^{\rightarrow}$ com a definição geral de Wells, como afirmado nos Teoremas 4.1.5, 4.2.3 e 4.3.6, respectivamente.

Demonstração. As provas utilizam a mesma técnica da prova apresentada por Wells em [103] para a correspondência das Definições 2.2.2 e 2.3.14. A prova para o $\lambda\sigma^{\rightarrow}$ tem uma adaptação para tratar também de substituições. Seja $\mathsf{u} \in \{\lambda_{dB}, \lambda s_e, \lambda\sigma\}$ e $\mathcal{O}_{\mathsf{u}}$ o operador para atualização de índices de cada cálculo. Em outras palavras, $\mathcal{O}_{\lambda dB}(M) = M^{+}$, $\mathcal{O}_{\lambda s_e}(M) = \varphi_0^2 M$ e $\mathcal{O}_{\lambda\sigma}(M) = M[\uparrow]$. Seja $\mathcal{O}_{\mathsf{u}}^1 = \mathcal{O}_{\mathsf{u}}$ e $\mathcal{O}_{\mathsf{u}}^{n+1}(M) = \mathcal{O}_{\mathsf{u}}(\mathcal{O}_{\mathsf{u}}^n(M))$. Para $\tau \in \mathcal{S}$, seja $TV(\tau)$ o conjunto das variáveis de tipo que ocorrem em τ. Abreviaremos $\underline{1}\,[\uparrow^n]$ com $\underline{n{+}1}$.

Demonstração de $\Rightarrow$: Seja $\Theta_{\mathsf{u}} = \langle \Gamma_{\mathsf{u}} \vdash \tau_{\mathsf{u}} \rangle$ uma PT de um termo M_{u}, de acordo às Definições 4.1.4, 4.2.2 e 4.3.5 e $\Theta'_{\mathsf{u}} = \langle \Gamma'_{\mathsf{u}} \vdash \tau'_{\mathsf{u}} \rangle$ uma tipagem de M_{u} em $\mathsf{u}^{\rightarrow}$. A partir da definição de PT para cada sistema, existe uma substituição de tipos s tal que $s(\Gamma_{\mathsf{u}}) = (\Gamma'_{\mathsf{u}})_{\leq|\Gamma_{\mathsf{u}}|}.nil$ e $s(\tau_{\mathsf{u}}) = \tau'_{\mathsf{u}}$. Tem-se que $u^{\rightarrow} \Vdash M : \Theta_{\mathsf{u}}$ implica $u^{\rightarrow} \Vdash M : s(\Theta_{\mathsf{u}})$, para qualquer substituição de tipos s. Portanto, $\Theta_{\mathsf{u}} \leq_{u^{\rightarrow}} s(\Theta_{\mathsf{u}})$. Com as leis de redundância admissíveis em cada sistema ((λdB-weak), (λs_e-weak) e ($\lambda\sigma$-tweak)), tem-se que $s(\Theta_{\mathsf{u}}) \leq_{u^{\rightarrow}} \Theta'_{\mathsf{u}}$. Assim, Θ_{u} é PT de M_{u} em $\mathsf{u}^{\rightarrow}$ de acordo com a Definição 2.2.2.

A demonstração para uma substituição S em $\lambda\sigma$ com PT $\Theta = \langle \Gamma \rhd \Delta \rangle$ de acordo a Definição 4.3.5 e uma tipagem $\Theta' = \langle \Gamma' \rhd \Delta' \rangle$ é similar a demonstração para termos em $\lambda\sigma$, aplicando a lei de redundância apropriada, ($\lambda\sigma$-sweak).

Demonstração de $\Leftarrow$: Seja $\Theta_{\mathsf{u}} = \langle \Gamma_{\mathsf{u}} \vdash \tau_{\mathsf{u}} \rangle$ uma PT de um termo M_{u}, de acordo às Definições 4.1.4, 4.2.2 e 4.3.5 e $\Theta'_{\mathsf{u}} = \langle \Gamma'_{\mathsf{u}} \vdash \tau'_{\mathsf{u}} \rangle$ uma tipagem de M_{u} em $\mathsf{u}^{\rightarrow}$ que não seja PT de acordo com tais definições. Então, existe uma substituição de tipos s tal que $s(\Gamma_{\mathsf{u}}) = (\Gamma'_{\mathsf{u}})_{\leq|\Gamma_{\mathsf{u}}|}.nil$ e $s(\tau_{\mathsf{u}}) = \tau'_{\mathsf{u}}$ e não existe uma substituição de tipos s' tal que $s'(\Gamma'_{\mathsf{u}}) = (\Gamma_{\mathsf{u}})_{\leq|\Gamma'_{\mathsf{u}}|}.nil$ e $s'(\tau'_{\mathsf{u}}) = \tau_{\mathsf{u}}$.

1. Se $s(\Gamma_{\mathsf{u}}) \neq \Gamma'_{\mathsf{u}}$, então $m_{\mathsf{u}} = |\Gamma_{\mathsf{u}}| < |\Gamma'_{\mathsf{u}}|$. Seja $N_{\mathsf{u}} = (\lambda.\mathcal{O}_{\mathsf{u}}(M_{\mathsf{u}})\ \underline{m_{\mathsf{u}}{+}1}\,)$.

2. Se $s(\Gamma_{\mathsf{u}}) = \Gamma'_{\mathsf{u}}$, seja $\alpha \in \mathcal{A}$. Defina as funções ϕ_1^{u} e ϕ_2^{u} por:

$$
\begin{array}{rcl}
\phi_1^{\mathsf{u}}(\alpha,\alpha) & = & \lambda.\lambda.\big(\,\underline{1}\;(\underline{2}\;\underline{4})\;(\underline{2}\;\underline{3})\big)\\
\phi_1^{\mathsf{u}}(\sigma\to\tau,\alpha) & = & \begin{cases} \lambda.\lambda.\big(\,\underline{1}\;(\underline{3}\;\underline{2})\;(\mathcal{O}_{\mathsf{u}}^3(\lambda.\phi_1^{\mathsf{u}}(\sigma,\alpha))\;\underline{2})\big), & \text{se } \alpha \in TV(\sigma)\\ \lambda.\big(\mathcal{O}_{\mathsf{u}}^2(\lambda.\phi_1^{\mathsf{u}}(\tau,\alpha))\;(\underline{2}\;\underline{1})\big), & \text{c.c.}\end{cases}\\
\phi_2^{\mathsf{u}}(\alpha,\alpha) & = & \lambda.\lambda.\big(\,\underline{1}\;(\underline{2}\;\underline{3})\;(\underline{2}\;\underline{4})\big)\\
\phi_2^{\mathsf{u}}(\sigma\to\tau,\alpha) & = & \begin{cases} \lambda.\lambda.\big(\,\underline{1}\;(\underline{4}\;\underline{2})\;(\mathcal{O}_{\mathsf{u}}^2(\lambda.\phi_1^{\mathsf{u}}(\sigma,\alpha))\;\underline{2})\big), & \text{se } \alpha \in TV(\sigma)\\ \lambda.\big(\mathcal{O}_{\mathsf{u}}(\lambda.\phi_1^{\mathsf{u}}(\tau,\alpha))\;(\underline{3}\;\underline{1})\big), & \text{c.c.}\end{cases}
\end{array}
$$

(a) Suponha que $s(\alpha_{\mathsf{u}})$ não seja uma variável de tipo, para $\alpha_{\mathsf{u}} \in TV(\Theta_{\mathsf{u}})$

i. Suponha que $\alpha_{\mathsf{u}} \in TV(\tau_{\mathsf{u}})$. Seja

$$N_{\mathsf{u}} = \Big(\lambda.\big(\lambda.\underline{2}\;\lambda.(\mathcal{O}_{\mathsf{u}}(\lambda.\phi_2^{\mathsf{u}}(\tau_{\mathsf{u}},\alpha_{\mathsf{u}}))\;\lambda.\underline{2}\,)\big)\;M_{\mathsf{u}}\Big)$$

ii. Suponha que $\alpha_{\mathsf{u}} \in TV((\Gamma_{\mathsf{u}})_{i_{\mathsf{u}}})$. Seja

$$N_{\mathsf{u}} = \big(\lambda.\mathcal{O}_{\mathsf{u}}(M_{\mathsf{u}})\;\lambda.(\lambda.\lambda.\phi_2^{\mathsf{u}}((\Gamma_{\mathsf{u}})_{i_{\mathsf{u}}},\alpha_{\mathsf{u}})\;\underline{i_{\mathsf{u}}+1}\;\lambda.\underline{2}\,)\big)$$

(b) Suponha que $s(\alpha_{\mathsf{u}}^1) = s(\alpha_{\mathsf{u}}^2) = \beta$ para $\alpha_{\mathsf{u}}^1, \alpha_{\mathsf{u}}^2 \in TV(\Theta_{\mathsf{u}})$, distintas.

i. Suponha que $\alpha_{\mathsf{u}}^j \in TV((\Gamma_{\mathsf{u}})_{i_{\mathsf{u},j}})$ para $j \in \{1,2\}$. Seja

$$P_{\mathsf{u}} = \lambda.\lambda.\big(\,\underline{1}\;\mathcal{O}_{\mathsf{u}}(P_{\mathsf{u}}^1)\;\mathcal{O}_{\mathsf{u}}(P_{\mathsf{u}}^2)\big)$$

onde $P_{\mathsf{u}}^j = \big(\lambda.\phi_1^{\mathsf{u}}((\Gamma_{\mathsf{u}})_{i_{\mathsf{u},j}},\alpha_{\mathsf{u}}^j)\;\;\underline{i_{\mathsf{u},j}+1}\big)$. Seja então $N_{\mathsf{u}} = \big(\lambda.\lambda.\underline{2}\;M_{\mathsf{u}}\,P_{\mathsf{u}}\big)$

ii. Suponha que $\alpha_{\mathsf{u}}^1 \in TV((\Gamma_{\mathsf{u}})_{i_{\mathsf{u}}})$ e $\alpha_{\mathsf{u}}^2 \in TV(\tau_{\mathsf{u}})$. Seja

$$P_{\mathsf{u}} = \lambda.\lambda.\big(\,\underline{1}\;\big(\mathcal{O}_{\mathsf{u}}(\lambda.\phi_1^{\mathsf{u}}((\Gamma_{\mathsf{u}})_{i_{\mathsf{u}}},\alpha_{\mathsf{u}}^1))\;\;\underline{i_{\mathsf{u}}+3}\big)\;\mathcal{O}_{\mathsf{u}}(\phi_2^{\mathsf{u}}(\tau_{\mathsf{u}},\alpha_{\mathsf{u}}^2))\big)$$

e $N_{\mathsf{u}} = \big(\lambda.(\lambda.\underline{2}\;P_{\mathsf{u}})\;M_{\mathsf{u}}\big)$

iii. Suponha que $\alpha_{\mathsf{u}}^i \in TV(\tau_{\mathsf{u}})$ para $i \in \{1,2\}$. Seja

$$P_{\mathsf{u}} = \lambda.\lambda.\big(\,\underline{1}\;\mathcal{O}_{\mathsf{u}}(\phi_2^{\mathsf{u}}(\tau_{\mathsf{u}},\alpha_{\mathsf{u}}^1))\;\mathcal{O}_{\mathsf{u}}(\phi_2^{\mathsf{u}}(\tau_{\mathsf{u}},\alpha_{\mathsf{u}}^2))\big)$$

e $N_{\mathsf{u}} = \big(\lambda.(\lambda.\underline{2}\;P_{\mathsf{u}})\;M_{\mathsf{u}}\big)$

Então, $N_{\mathsf{u}} \in Termos_{u\to}(\Theta'_{\mathsf{u}}) \setminus Termos_{u\to}(\Theta_{\mathsf{u}})$. Portanto, $\Theta'_{\mathsf{u}} \not\leq_{u\to} \Theta_{\mathsf{u}}$.

Consequentemente, se Θ'_{u} não é PT de acordo com as Definições 4.1.4, 4.2.2 e 4.3.5, então Θ'_{u} não é PT de acordo com a Definição 2.2.2.

Seja S uma substituição em $\lambda\sigma$ e $\Theta = \langle \Gamma \rhd \Delta \rangle$ uma PT de S de acordo à Definição 4.3.5, e $\Theta' = \langle \Gamma' \rhd \Delta' \rangle$ uma tipagem de S em $\lambda\sigma^{\rightarrow}$ que não é PT de acordo com essa definição. Então existe uma substituição de tipos s tal que $s(\Gamma) = \Gamma'_{\leq|\Gamma|}.nil$ e $s(\Delta) = \Delta'_{\leq|\Delta|}.nil$ e não existe uma substituição s' tal que $s'(\Gamma') = \Gamma_{\leq|\Gamma'|}.nil$ e $s'(\Delta') = \Delta_{\leq|\Delta'|}.nil$.

1. Suponha que $s(\Gamma) \neq \Gamma'$. Então, $m = |\Gamma| < |\Gamma'|$. Sejam $S_i = (\,\underline{1}.\,\underline{2}.\cdots.\,\underline{m{+}1}.\uparrow^{m+1})$ e $T = S \circ S_i$

2. Caso contrário, $s(\Gamma) = \Gamma'$. Sejam ϕ_1 e ϕ_2 as funções $\phi_1^{\lambda\sigma}$ e $\phi_2^{\lambda\sigma}$ definidas acima, respectivamente.

   (a) Suponha que $s(\alpha)$ não é uma variável de tipo, para $\alpha \in TV(\Theta)$

      i. Suponha que $\alpha \in TV(\Delta_i)$. Seja $N = \Big(\lambda.\big(\lambda.\underline{2}\ \lambda.((\lambda.\phi_2(\Delta_i,\alpha))[\uparrow]\ \lambda.\underline{2}\,)\big)\ \underline{i}\Big)$ e seja $S'_i = (\,\underline{1}.\,\underline{2}.\cdots.\,\underline{i{-}1}.N.\uparrow^{i})$. Seja então, $T = S'_i \circ S$.

      ii. Suponha que $\alpha \in TV(\Gamma_i)$. Sejam N e S'_i como acima. Então, seja $T = S \circ S'_i$.

   (b) Suponha que $s(\alpha_1) = s(\alpha_2) = \beta$ para $\alpha_1, \alpha_2 \in TV(\Theta)$, distintas.

      i. Suponha que $\alpha_j \in TV(\Gamma_{i_j})$ para $j \in \{1,2\}$. Seja $P_j = \big(\lambda.\phi_1(\Gamma_{i_j},\alpha_j)\ \underline{i_j{+}1}\big)$ e $P = \lambda.\lambda.\big(\,\underline{1}\ P_1[\uparrow]\,P_2[\uparrow]\big)$. Seja $N_j = \big(\lambda.\lambda.\underline{2}\ \ \underline{i_j}\,P\big)$, onde $j \in \{1,2\}$ e seja $S_{i_j} = (\,\underline{1}.\,\underline{2}.\cdots.\,\underline{i_j{-}1}.N_j.\uparrow^{i_j})$. Seja então $T = S \circ S_{i_1}$ ou $T = S \circ S_{i_2}$.

      ii. Suponha que $\alpha_j \in TV(\Delta_{i_j})$, $j \in \{1,2\}$. Seja $P_j = \big(\lambda.\phi_1(\Delta_{i_j},\alpha_j)\ \underline{i_j{+}1}\big)$. Então, para P, N_j e S_{i_j} como definidos acima, seja $T = S_{i_j} \circ S$.

      iii. Suponha que $\alpha_1 \in TV(\Gamma_i)$ e $\alpha_2 \in TV(\Delta_j)$. Seja $N = \big(\lambda.(\lambda.\underline{2}\ P)\ \underline{j}[S]\big)$, onde

$$P = \lambda.\lambda.\big(\,\underline{1}\ \big((\lambda.\phi_1(\Gamma_i,\alpha_1))[\uparrow]\ \ \underline{i{+}3}\big)\ \phi_2(\Delta_j,\alpha_2)[\uparrow]\big)$$

      Então, seja $T = \big(\,\underline{1}[S].\,\underline{2}[S].\cdots.\,\underline{j-1}[S].N.(\uparrow^{j} \circ S)\big)$.

   Assim, $T \in Termos_{\lambda\sigma^{\rightarrow}}(\Theta') \setminus Termos_{\lambda\sigma^{\rightarrow}}(\Theta)$. Portanto, $\Theta' \not\leq_{\lambda\sigma^{\rightarrow}} \Theta$.

Consequentemente, se uma tipagem Θ' de uma substituição em $\lambda\sigma$ não é PT de acordo à Definição 4.3.5, então Θ' não é PT de acordo à Definição 2.2.2. $\square$

Capítulo 5

Tipos com interseção para o λ_{dB}-calculus

Apresentamos neste capítulo a versão à la de Bruijn de dois sistemas de IT. Na Seção 5.1 apresentamos o sistema baseado no sistema de E. Sayag e M. Mauny em [37]. O sistema de tipos é apresentado como em [38], onde os autores tentam estender os resultados para formas β-normais de [37] para qualquer termo que possua formas β-normais. Na Seção 5.2 apresentamos o sistema baseado no sistema de F. Kamareddine e K. Nour em [56]. O sistema da segunda seção, juntamente com a propriedade de SR, foi o primeiro sistema de IT proposto para o λ_{dB}-calculus [98]. Porém, devido a presença de uma relação de subtipos, o sistema perde a propriedade de relevância satisfeita no sistema de Kamareddine e Nour. Assim, estudamos a versão à la de Bruijn do sistema em [37], sendo este sistema a base para os sistemas de IT propostos aos cálculos de ES estudados.

O sistema de [37] foi originalmente introduzido para caracterizar sintaticamente PT de formas β-normais e, por ser um versão mais restrita dos sistemas apresentados em [95], é o primeiro passo na direção a proposta de sistemas de IT para $\lambda\sigma$ e λs_e. A caracterização de PT para formas β-normais em λ_{dB} é apresentada no Capítulo 6.

5.1 O sistema λ_{dB}^{SM}

O conjunto $\mathcal{T}$ de tipos atribuídos aos termos nesse sistema é o mesmo conjunto do sistema λ^{SM}, descrito na Seção 2.5.1. A seguir, definimos este conjunto de uma maneira similar aos conjuntos $\mathbb{T}$ e $\mathbb{U}$ de $\lambda^{\sqcap}$.

Definição 5.1.1: 1. Seja $\mathcal{A}$ o conjunto de variáveis de tipo. O conjunto $\mathcal{T}$ de **tipos com interseção restrita** é definido por:

$$\tau, \sigma \in \mathcal{T} ::= \mathcal{A} \,|\, \mathcal{U} \to \mathcal{T} \qquad u \in \mathcal{U} ::= \omega \,|\, \mathcal{U} \wedge \mathcal{U} \,|\, \mathcal{T}$$

Os tipos são considerados módulo comutatividade, associatividade e ω o elemento neutro de $\wedge$.

2. Os contextos de tipo são sequênciais, definidos para $u \in \mathcal{U}$ da forma: $\Gamma ::= nil \,|\, u.\Gamma$.

   O **contexto omega** $\omega^{\underline{n}}$ denota a sequência $\omega.\omega.\cdots.\omega.nil$ de comprimento n. Seja $\omega^{\underline{0}}.\Gamma = \Gamma$.

   A extensão de $\wedge$ para contextos é tal que nil é o elemento neutro e $(u_1.\Gamma) \wedge (u_2.\Delta) = (u_1 \wedge u_2).(\Gamma \wedge \Delta)$. Assim, $\wedge$ é comutativa e associativa para contextos.

3. Seja $u' \sqsubseteq u$ se existe v tal que $u = u' \wedge v$ e $u' \sqsubset u$ se $v \neq \omega$.

   Seja $\Gamma' \sqsubseteq \Gamma$ se existe um contexto Δ tal que $\Gamma = \Gamma' \wedge \Delta$, onde tanto Γ' quanto Δ não são contextos omegas, e $\Gamma' \sqsubset \Gamma$ se $\Delta \neq nil$. Seja tal Δ denotado por Γ / Γ'.

O lema a seguir mostra que os conjuntos $\mathcal{T}$ e $\mathcal{U}$ definidos acima são equivalentes aos conjuntos correspondentes definidos em [37], descritos na Seção 2.5.1.

Lema 5.1.2: 1. Se $u \in \mathcal{U}$, então $u = \omega$ ou $u = \wedge_{i=1}^{n} \tau_i$ $n > 0$ e $\forall\, 1 \leq i \leq n$, $\tau_i \in \mathcal{T}$.

2. Se $\tau \in \mathcal{T}$, então $\tau = \alpha$, $\tau = \omega \to \sigma$ ou $\tau = \wedge_{i=1}^{n} \tau_i \to \sigma$, onde $n > 0$ e $\sigma, \tau_1, \ldots, \tau_n \in \mathcal{T}$.

Demonstração. 1. Por indução em $u \in \mathcal{U}$.

2. Por indução em $\tau \in \mathcal{T}$, com o Lema 5.1.2.1. □

Algumas propriedades de contextos e a respectiva extensão de $\wedge$ seguem de suas definições.

Lema 5.1.3: Sejam $\Gamma^1, \ldots, \Gamma^m$ contextos diferentes de nil:

1. $\Gamma^1 \wedge \cdots \wedge \Gamma^m = (\Gamma^1_1 \wedge \cdots \wedge \Gamma^m_1).(\Gamma^1_{>1} \wedge \cdots \wedge \Gamma^m_{>1})$.

2. Se $i \leq min(|\Gamma^1|, \ldots, |\Gamma^m|)$, então $(\Gamma^1 \wedge \cdots \wedge \Gamma^m)_i = \Gamma^1_i \wedge \cdots \wedge \Gamma^m_i$.

   Senão, $(\Gamma^1 \wedge \cdots \wedge \Gamma^m)_i = \Gamma^{j_1}_i \wedge \cdots \wedge \Gamma^{j_k}_i$, onde $k \leq m$ e $\forall 1 \leq l \leq k$, $\Gamma^{j_l}_i \in \mathcal{U}$.

3. $(\Gamma^1 \wedge \cdots \wedge \Gamma^m)_{<i} = \Gamma^1_{<i} \wedge \cdots \wedge \Gamma^m_{<i}$. Observe que se $i \geq |\Gamma^j|$ então $(\Gamma^1 \wedge \cdots \wedge \Gamma^m)_{<i} = \Gamma^1_{<i} \wedge \cdots \wedge \Gamma^j \wedge \cdots \wedge \Gamma^m_{<i}$. Tem-se propriedades similares para $(\Gamma \wedge \Delta)_{\leq i}$.

4. $(\Gamma^1 \wedge \cdots \wedge \Gamma^m)_{>i} = \Gamma^1_{>i} \wedge \cdots \wedge \Gamma^m_{>i}$. Observe que se $i \geq |\Gamma^j|$ então $(\Gamma^1 \wedge \cdots \wedge \Gamma^m)_{>i} = \Gamma^1_{>i} \wedge \cdots \wedge \Gamma^{j-1}_{>i} \wedge \Gamma^{j+1}_{>i} \wedge \cdots \wedge \Gamma^m_{>i}$. Tem-se propriedades similares para $(\Gamma \wedge \Delta)_{\geq i}$.

5. $(\Gamma^1 \wedge \cdots \wedge \Gamma^m)_{<i}.(\Gamma^1 \wedge \cdots \wedge \Gamma^m)_{>i} = (\Gamma^1_{<i}.\Gamma^1_{>i}) \wedge \cdots \wedge (\Gamma^m_{<i}.\Gamma^m_{>i})$.

6. $|\Gamma^1 \wedge \cdots \wedge \Gamma^m| = max(|\Gamma^1|, \ldots, |\Gamma^m|)$.

Definição 5.1.4 (Os sistemas $\lambda_{dB}^{SM}/\lambda_{dB}^{SM_r}$):

1. As regras de tipagem do sistema λ_{dB}^{SM} são dadas a seguir:

$$\frac{}{\underline{1}:\langle \tau.nil \vdash \tau\rangle}\ \text{var} \qquad \frac{M:\langle u.\Gamma \vdash \tau\rangle}{\lambda.M:\langle \Gamma \vdash u \rightarrow \tau\rangle}\ \rightarrow_i$$

$$\frac{\underline{n}:\langle \Gamma \vdash \tau\rangle}{\underline{n+1}:\langle \omega.\Gamma \vdash \tau\rangle}\ \text{varn} \qquad \frac{M:\langle nil \vdash \tau\rangle}{\lambda.M:\langle nil \vdash \omega \rightarrow \tau\rangle}\ \rightarrow_i'$$

$$\frac{M_1:\langle \Gamma \vdash \omega \rightarrow \tau\rangle \qquad M_2:\langle \Delta \vdash \sigma\rangle}{(M_1\ M_2):\langle \Gamma \wedge \Delta \vdash \tau\rangle}\ \rightarrow_e'$$

$$\frac{M_1:\langle \Gamma \vdash \wedge_{i=1}^n \sigma_i \rightarrow \tau\rangle \qquad M_2:\langle \Delta^1 \vdash \sigma_1\rangle \ \ldots\ M_2:\langle \Delta^n \vdash \sigma_n\rangle}{(M_1\ M_2):\langle \Gamma \wedge \Delta^1 \wedge \cdots \wedge \Delta^n \vdash \tau\rangle}\ \rightarrow_e$$

2. O sistema $\lambda_{dB}^{SM_r}$ é obtido de λ_{dB}^{SM}, trocando a regra var pela regra:

$$\underline{1}:\langle \sigma_1 \rightarrow \cdots \rightarrow \sigma_n \rightarrow \alpha.nil \vdash \sigma_1 \rightarrow \cdots \rightarrow \sigma_n \rightarrow \alpha\rangle\ (n \geq 0) \quad \text{var}_r$$

A regra de inferência varn permite a construção dos contextos sequenciais. Observe que, ao invés da introdução de tipos quaisquer como na regra para o sistema $\lambda_{dB}^{\rightarrow}$, a introdução é restrita ao ω, que nesse sistema pode ser interpretado como a interseção vazia. Dessa forma, restringimos as designações de tipos somente aos índices que ocorrem livre no termo correpondente. Essa é a característica principal de sistemas relevantes, como descrito em [29]. Provamos a relevância para o sistema mais adiante.

Observe que o axioma var é restrito aos tipos em $\mathcal{T}$ e a introdução da interseção ocorre nos contextos de tipos, através das regra $\rightarrow_e$ e $\rightarrow_e'$. Para os tipos atribuídos a termos, apenas as regras $\rightarrow_i$ e $\rightarrow_i'$ introduzem os elementos do conjunto $\mathcal{U}$, sempre a esquerda de

$\rightarrow$. Para o sistema $\lambda_{dB}^{SM_r}$, os subtermos de tipos atribuídos a termos que são elementos de $\mathcal{U}$ são todos, e apenas, introduzidos pelas regras $\rightarrow_i$ e $\rightarrow_i'$. O lema a seguir estabelece que os tipos atribuídos aos termos são os elementos de $\mathcal{T}$, enquanto os tipos designados aos índices são elementos de $\mathcal{U}$.

Lema 5.1.5: Se $M:\langle\Gamma\vdash_{\lambda_{dB}^{SM}} u\rangle$, então $u\in\mathcal{T}$, e $\forall 1\leq i\leq|\Gamma|$, $\Gamma_i\in\mathcal{U}$ sempre que $\Gamma\neq nil$.

Demonstração. Por indução na derivação $M:\langle\Gamma\vdash u\rangle$.

- Se $\underline{1}:\langle\tau.nil\vdash\tau\rangle$, nada há provar.
- Seja $\dfrac{\underline{n}:\langle\Gamma\vdash u\rangle}{\underline{n+1}:\langle\omega.\Gamma\vdash u\rangle}$. Por HI tem-se que $u\in\mathcal{T}$ e $\forall 1\leq i\leq|\Gamma|$, $\Gamma_i\in\mathcal{U}$. Note que $|\omega.\Gamma|=n{+}1$, $(\omega.\Gamma)_1=\omega$ e $\forall 1\leq i\leq n$, $(\omega.\Gamma)_{i+1}=\Gamma_i$.
- Seja $\dfrac{M:\langle u.\Gamma\vdash v\rangle}{\lambda.M:\langle\Gamma\vdash u\rightarrow v\rangle}$. Por HI tem-se que $v\in\mathcal{T}$ e $u\in\mathcal{U}$, então $u\rightarrow v\in\mathcal{T}$. Note que $\Gamma\equiv nil$ ou, por HI, $\forall 1\leq i\leq|\Gamma|$, $(u.\Gamma)_{i+1}=\Gamma_i\in\mathcal{U}$.
- Seja $\dfrac{M:\langle nil\vdash v\rangle}{\lambda.M:\langle nil\vdash\omega\rightarrow v\rangle}$. Por HI tem-se que $v\in\mathcal{T}$ logo $\omega\rightarrow v\in\mathcal{T}$.
- Seja $\dfrac{M_1:\langle\Gamma\vdash\omega\rightarrow u\rangle\qquad M_2:\langle\Delta\vdash v\rangle}{(M_1\ M_2):\langle\Gamma\wedge\Delta\vdash u\rangle}$. Por HI, $\omega\rightarrow u\in\mathcal{T}$, assim pelo Lema 5.1.2.2, $u\in\mathcal{T}$. Se $\Gamma\wedge\Delta\equiv nil$, nada há provar. Senão, seja $1\leq i\leq|\Gamma\wedge\Delta|$ e suponha s.p.d.g. que $i\leq|\Gamma|,|\Delta|$. Assim, $(\Gamma\wedge\Delta)_i=\Gamma_i\wedge\Delta_i$ tal que, por HI, $\Gamma_i,\Delta_i\in\mathcal{U}$, então $\Gamma_i\wedge\Delta_i\in\mathcal{U}$.
- Seja $\dfrac{M_1:\langle\Gamma\vdash\wedge_{k=1}^n v_k\rightarrow u\rangle\qquad M_2:\langle\Delta^1\vdash v_1\rangle\ \ldots\ M_2:\langle\Delta^n\vdash v_n\rangle}{(M_1\ M_2):\langle\Gamma\wedge\Delta^1\wedge\cdots\wedge\Delta^n\vdash u\rangle}$. Por HI, $\wedge_{k=1}^n v_k\rightarrow u\in\mathcal{T}$, assim, pelo Lema 5.1.2.2, $u\in\mathcal{T}$. Se $\Gamma\wedge\Delta^1\wedge\cdots\wedge\Delta^n\equiv nil$, nada há provar. Senão, se $\forall 1\leq k\leq n$, $\Delta^k\equiv nil$ então $\Gamma\wedge\Delta^1\wedge\cdots\wedge\Delta^n=\Gamma$ e, por HI, o resultado vale. Se $|\Delta^k|>0$ para algum $1\leq k\leq n$ então seja $\Delta'=\Delta^1\wedge\cdots\wedge\Delta^n$. Suponha s.p.d.g. que $\forall 1\leq k\leq n$, $|\Delta^k|>0$ e que possuem o mesmo comprimento. Assim, para qualquer $1\leq j\leq|\Delta'|$, $\Delta'_j=\Delta^1_j\wedge\cdots\wedge\Delta^n_j$ e, por HI, $\Delta^k_j\in\mathcal{U}$ para cada $1\leq k\leq n$ logo $\Delta'_j\in\mathcal{U}$. Se $\Gamma\equiv nil$, então $\Gamma\wedge\Delta'=\Delta'$ e o resultado vale, caso contrário a demonstração é análoga a anterior. $\square$

Lema 5.1.6: λ_{dB}^{SM} é uma extensão própria de $\lambda_{dB}^{SM_r}$.

Demonstração. Se $M : \langle \Gamma \vdash_{\lambda_{dB}^{SM_r}} \tau \rangle$ então tem-se $M : \langle \Gamma \vdash_{\lambda_{dB}^{SM}} \tau \rangle$, trivialmente. Como um contra-exemplo para a equivalência, tome $M \equiv (\underline{1}\ \lambda.\underline{1})$. Então tem-se que $M : \langle \tau \wedge (\tau \to \tau) \to \beta.nil \vdash_{\lambda_{dB}^{SM}} \beta \rangle$, para $\tau = \alpha \to \alpha$, e $(\tau \wedge (\tau \to \tau) \to \beta.nil, \beta)$ não é uma tipagem de M em $\lambda_{dB}^{SM_r}$. $\square$

Portanto, as propriedades provadas para λ_{dB}^{SM} também são válidas para $\lambda_{dB}^{SM_r}$.

5.1.1 Propriedades

O lema a seguir afirma que λ_{dB}^{SM} é relevante de acordo à [29].

Lema 5.1.7 (Relevância para λ_{dB}^{SM}): Se $M : \langle \Gamma \vdash_{\lambda_{dB}^{SM}} \tau \rangle$, então:

i) $|\Gamma| = sup(M)$.

ii) $\forall 1 \leq i \leq |\Gamma|$, $\Gamma_i \neq \omega$ sse $\underline{i} \in FI(M)$.

Demonstração. Por indução na derivação $M : \langle \Gamma \vdash u \rangle$.

- Se $\underline{1} : \langle \tau.nil \vdash \tau \rangle$, então $|\Gamma| = 1 = sup(\underline{1})$. Note que $FI(\underline{1}) = \{\underline{1}\}$ e $\Gamma_1 = \tau$.

- Se $\dfrac{\underline{n} : \langle \Gamma \vdash \tau \rangle}{\underline{n+1} : \langle \omega.\Gamma \vdash \tau \rangle}$, então por HI tem-se $|\Gamma| = sup(\underline{n}) = n$, $\Gamma_n \neq \omega$ e $\forall 1 \leq i < n$, $\Gamma_i = \omega$. Assim, $|\omega.\Gamma| = 1 + |\Gamma| = n{+}1 = sup(\underline{n{+}1})$, $(\omega.\Gamma)_{n+1} = \Gamma_n \neq \omega$, $(\omega.\Gamma)_1 = \omega$ e $\forall 1 \leq i < n$, $(\omega.\Gamma)_{i+1} = \Gamma_i = \omega$.

- Seja $\dfrac{M : \langle u.\Gamma \vdash \sigma \rangle}{\lambda.M : \langle \Gamma \vdash u \to \sigma \rangle}$. Por HI, $|u.\Gamma| = sup(M)$ e $\forall 0 \leq i \leq sup(M){-}1$, $(u.\Gamma)_{i+1} \neq \omega$ sse $\underline{i+1} \in FI(M)$. Assim, $sup(M) = 1 + |\Gamma| > 0$ e, pelo Lema 3.1.10.2, $sup(\lambda.M) = sup(M) - 1 = |\Gamma|$. Pelo Lema 3.1.9.1, $\forall 1 \leq i \leq sup(\lambda.M)$, $\underline{i} \in FI(\lambda.M)$ sse $\underline{i+1} \in FI(M)$. Assim, $(u.\Gamma)_{i+1} = \Gamma_i \neq \omega$ sse $\underline{i} \in FI(\lambda.M)$.

- Seja $\dfrac{M : \langle nil \vdash \sigma \rangle}{\lambda.M : \langle nil \vdash \omega \to \sigma \rangle}$. Por HI tem-se que $|nil| = sup(M) = 0$. Assim, pelo Lema 3.1.10.2, $sup(\lambda.M) = sup(M) = |nil|$. Note que $FI(M) = FI(\lambda.M) = \emptyset$.

- Seja $\dfrac{M_1 : \langle \Gamma \vdash \omega \to \tau \rangle \qquad M_2 : \langle \Delta \vdash \sigma \rangle}{(M_1\ M_2) : \langle \Gamma \wedge \Delta \vdash \tau \rangle}$. Por HI, $|\Gamma| = sup(M_1)$ onde $\forall 1 \leq i \leq |\Gamma|$ tem-se $\Gamma_i \neq \omega$ sse $\underline{i} \in FI(M_1)$, $|\Delta| = sup(M_2)$ onde $\forall 1 \leq j \leq |\Delta|$ tem-se $\Delta_j \neq \omega$ sse $\underline{j} \in FI(M_2)$. Pelo Lema 3.1.10.1 tem-se que $sup(M_1\ M_2) = max(sup(M_1), sup(M_2)) =$

$max(|\Gamma|,|\Delta|) = |\Gamma \wedge \Delta|$. Seja $1 \leq l \leq |\Gamma \wedge \Delta|$ e suponha s.p.d.g. que $l \leq |\Gamma|, |\Delta|$. Assim, $(\Gamma \wedge \Delta)_l = \Gamma_l \wedge \Delta_l \neq \omega$ *sse* $\Gamma_l \neq \omega$ ou $\Delta_l \neq \omega$ *sse* $\underline{l} \in FI(M_1)$ ou $\underline{l} \in FI(M_2)$ *sse* $\underline{l} \in FI(M_1) \cup FI(M_2) = FI(M_1\ M_2)$.

- Seja $\dfrac{M_1 : \langle \Gamma \vdash \wedge_{k=1}^{n} \sigma_k \to \tau \rangle \qquad M_2 : \langle \Delta^1 \vdash \sigma_1 \rangle \ \ldots \ M_2 : \langle \Delta^n \vdash \sigma_n \rangle}{(M_1\ M_2) : \langle \Gamma \wedge \Delta^1 \wedge \cdots \wedge \Delta^n \vdash \tau \rangle}$. Por HI tem-se que $|\Gamma| = sup(M_1)$ onde $\forall 1 \leq i \leq |\Gamma|$, $\Gamma_i \neq \omega$ sse $\underline{i} \in FI(M_1)$ e $|\Delta^k| = sup(M_2)$ onde $\forall 1 \leq j \leq |\Delta^k|$, $\Delta^k_j \neq \omega$ sse $\underline{j} \in FI(M_2)$. Seja $\Delta' = \Delta^1 \wedge \cdots \wedge \Delta^n$. Assim, $|\Delta'| = sup(M_2)$ e $\forall 1 \leq j \leq |\Delta'|$, $\Delta'_j \neq \omega$ *sse* $\underline{j} \in FI(M_2)$. A demonstração então é análoga a anterior. $\square$

Observe que, pelo Lema 5.1.7 acima o sistema λ_{dB}^{SM} não é apenas relevante mas existe uma relação estreita entre os índices livres do termo e o comprimento do contexto em sua tipagem. A seguir, lemas de geração para tipagens em λ_{dB}^{SM}, e alguns para tipagens em $\lambda_{dB}^{SM_r}$, são apresentados.

Lema 5.1.8 (Geração para λ_{dB}^{SM}):

1. Se $\underline{n} : \langle \Gamma \vdash_{\lambda_{dB}^{SM}} \tau \rangle$, então $\Gamma_n = \tau$.
2. Se $\underline{n} : \langle \Gamma \vdash_{\lambda_{dB}^{SM_r}} \tau \rangle$, então $\tau = \sigma_1 \to \cdots \to \sigma_k \to \alpha$ para $k \geq 0$.
3. Se $\lambda.M : \langle nil \vdash_{\lambda_{dB}^{SM}} \tau \rangle$, então $\tau = \omega \to \sigma$ e $M : \langle nil \vdash_{\lambda_{dB}^{SM}} \sigma \rangle$ ou $\tau = \wedge_{i=1}^{n} \sigma_i \to \sigma$, $n > 0$, e $M : \langle \wedge_{i=1}^{n} \sigma_i . nil \vdash_{\lambda_{dB}^{SM}} \sigma \rangle$ para $\sigma, \sigma_1, \ldots, \sigma_n \in \mathcal{T}$.
4. Se $\lambda.M : \langle \Gamma \vdash_{\lambda_{dB}^{SM}} \tau \rangle$ e $|\Gamma| > 0$, então $\tau = u \to \sigma$ para algum $u \in \mathcal{U}$ e $\sigma \in \mathcal{T}$, onde $M : \langle u.\Gamma \vdash_{\lambda_{dB}^{SM}} \sigma \rangle$.
5. Se $(\underline{n}\ M_1 \cdots M_m) : \langle \Gamma \vdash_{\lambda_{dB}^{SM_r}} \tau \rangle$ então, para $\tau = \sigma_{m+1} \to \cdots \to \sigma_{m+k} \to \alpha$, $\Gamma = (\omega^{\underline{n-1}} . \sigma_1 \to \cdots \to \sigma_m \to \tau . nil) \wedge \Gamma^1 \wedge \cdots \wedge \Gamma^m$ onde $\forall 1 \leq i \leq m$, $M_i : \langle \Gamma^i \vdash_{\lambda_{dB}^{SM_r}} \sigma_i \rangle$.

Demonstração. 1. Por indução na derivação $\underline{n} : \langle \Gamma \vdash_{\lambda_{dB}^{SM}} \tau \rangle$. Note que $(\omega.\Gamma)_{n+1} = \Gamma_n$.

2. Por indução na derivação $\underline{n} : \langle \Gamma \vdash_{\lambda_{dB}^{SM_r}} \tau \rangle$.

3. Por análise de casos na derivação $\lambda.M : \langle nil \vdash_{\lambda_{dB}^{SM}} \tau \rangle$.

4. Por análise de casos na derivação $\lambda.M : \langle \Gamma \vdash_{\lambda_{dB}^{SM}} \tau \rangle$, para $|\Gamma| > 0$.

5. Por indução em m.

   Se $m = 0$ então, pelo Lema 5.1.8.2, $\tau = \sigma_1 \to \cdots \to \sigma_k \to \alpha$. Assim, pelos Lemas 5.1.7 e 5.1.8.1, $\Gamma = \underline{\omega^{n-1}}.\tau.nil$.

   Se $m = m' + 1$, então por análise de casos o último passo na derivação é

$$\frac{\underline{n}\; M_1 \cdots M_{m'} : \langle \Gamma \vdash \wedge_{j=1}^{l} \tau_j \to \tau \rangle \qquad M_{m'+1} : \langle \Delta^1 \vdash \tau_1 \rangle \;\ldots\; M_{m'+1} : \langle \Delta^l \vdash \tau_l \rangle}{(\underline{n}\; M_1 \cdots M_{m'}\; M_{m'+1}) : \langle \Gamma \wedge \Delta^1 \wedge \cdots \wedge \Delta^l \vdash \tau \rangle}$$

   Por HI tem-se que $\Gamma = (\underline{\omega^{n-1}}.\sigma_1 \to \cdots \to \sigma_{m'} \to (\wedge_{j=1}^{l} \tau_j \to \tau).nil) \wedge \Gamma^1 \wedge \cdots \wedge \Gamma^{m'}$, onde $\wedge_{j=1}^{l} \tau_j \to \tau = \sigma_{m'+1} \to \cdots \to \sigma_{m'+k} \to \alpha$ e $\forall 1 \leq i \leq m'$, $M_i : \langle \Gamma^i \vdash_{\lambda_{dB}^{SMr}} \sigma_i \rangle$. Assim, $l = 1$, $\tau_1 = \sigma_{m'+1}$ e $\tau = \sigma_{m'+2} \to \cdots \to \sigma_{m+k} \to \alpha$. Logo, tomando $\Gamma^{m'+1} = \Delta^1$ e $\sigma_{m'+1} = \tau_1$, o resultado é válido. $\square$

5.1.2 Redução de sujeito

Por causa da inclusão da informação de tipos sobre o argumento da aplicação relacionada ao ω no contexto final na regra $\to_e'$, esse sistema não possui as propriedades de expansão/redução de sujeito em seu sentido usual. A seguir, um contra-exemplo para a propriedade de expansão de sujeito.

Exemplo 5.1.1: A afirmação que deve ser provada, de forma que a propriedade de expansão de sujeito seja válida, é: Se $\{\underline{1}/N\}M : \langle \Gamma \vdash \tau \rangle$ então $(\lambda.M\; N) : \langle \Gamma \vdash \tau \rangle$.

Sejam $M \equiv \lambda.\underline{1}$ e $N \equiv \underline{3}$ logo $\{\underline{1}/\underline{3}\}(\lambda.\underline{1}) = \lambda.\underline{1}$. Temos, pelo lema de geração, $\lambda.\underline{1} : \langle nil \vdash \alpha \to \alpha \rangle$. Assim, $\lambda.\lambda.\underline{1} : \langle nil \vdash \omega \to \alpha \to \alpha \rangle$ e $\underline{3} : \langle \omega.\omega.\beta.nil \vdash \beta \rangle$, então $(\lambda.\lambda.\underline{1}\; \underline{3}) : \langle \omega.\omega.\beta.nil \vdash \alpha \to \alpha \rangle$.

No caso da propriedade de redução de sujeito precisamos da afirmação: Se $(\lambda.M\; N) : \langle \Gamma \vdash \tau \rangle$ então $\{\underline{1}/N\}M : \langle \Gamma \vdash \tau \rangle$. Observe que tomando M e N como no exemplo acima temos o mesmo problema, mas na direção oposta. Em outras palavras, tem-se uma restrição do contexto original depois da redução β, pois a informação de tipos referente a $N \equiv \underline{3}$ é perdida.

Uma possível solução para estes problemas seria a substituição da regra $\to_e'$ pela regra $\to_e^{\omega}$ abaixo:

$$\frac{M : \langle \Gamma \vdash \omega \to \tau \rangle}{(M\; N) : \langle \Gamma \vdash \tau \rangle}$$

Essa abordagem foi originalmente apresentada em [38]. Porém, uma noção apropriada deve ser introduzida no lugar dos índices livres, pois não teríamos informação de tipos para todo índice livre do termo correspondente. A seguir, apresentamos um exemplo que ilustra o problema no uso de índices livres em um sistema com a regra $\rightarrow^{\omega}_{e}$.

Exemplo 5.1.2: A derivação de $M \equiv (\lambda.\underline{2}\ \underline{3})$ abaixo usa a regra $\rightarrow^{\omega}_{e}$.

$$\dfrac{\dfrac{\underline{2}:\langle \omega.\tau.nil \vdash \tau\rangle}{\lambda.\underline{2}:\langle \tau.nil \vdash \omega\rightarrow\tau\rangle}}{(\lambda.\underline{2}\ \underline{3}):\langle \tau.nil \vdash \tau\rangle}$$

Observe que, apesar de $\underline{3} \in FI(M)$, a tipagem $\langle \tau.nil \vdash \tau\rangle$ acima não designa nenhum tipo para este índice.

Em [38], e em [39], nenhuma noção nova foi apresenta além das usuais variáveis livres. Assim, a hipótese de que se $x \in FV(M)$ então x tem uma designação de tipo no contexto de qualquer tipagem de M, é usada no Lemas 1,2 e 4. Estes lemas são usados na demonstração dos Teoremas 1 e 2 que afirmam a expansão e redução de sujeito, respectivamente. Apesar disso, acreditamos que as propriedades possam ser demonstradas, com um lema de geração adequado para o sistema de tipos .

Para um sistema que caracterize SN para λ_{dB}, a regra $\rightarrow'_{e}$ deve incluir no contexto final essa informação do contexto sobre o termo que corresponde ao argumento da aplicação a ω. Caso contrário, se a regra tem a premissa da tipabilidade do termo, mas a informação do contexto não é incluída, temos o exemplo a seguir.

Exemplo 5.1.3: Seja $I \equiv \lambda.\underline{1}$, a função identidade. A autoaplicação $A \equiv (\underline{1}\ \underline{1})$ tem a tipagem $\langle(\alpha\rightarrow\beta)\wedge\alpha.nil \vdash \beta\rangle$ em λ^{SM}_{dB}. Assim, apresentamos a seguinta derivação, usando uma variação da regra $\rightarrow'_{e}$:

$$\dfrac{\dfrac{\dfrac{\lambda.I:\langle nil \vdash \omega\rightarrow\tau\rangle \qquad A:\langle(\alpha\rightarrow\beta)\wedge\alpha.nil \vdash \beta\rangle}{(\lambda.I\ A):\langle nil \vdash \tau\rangle}}{\lambda.(\lambda.I\ A):\langle nil \vdash \omega\rightarrow\tau\rangle} \qquad \dfrac{A:\langle(\alpha\rightarrow\beta)\wedge\alpha.nil \vdash \beta\rangle}{\lambda.A:\langle nil \vdash (\alpha\rightarrow\beta)\wedge\alpha\rightarrow\beta\rangle}}{(\lambda.(\lambda.I\ A)\ \lambda.A):\langle nil \vdash \tau\rangle}$$

Observe que $(\lambda.(\lambda.I\ A)\ \lambda.A) \rightarrow_{\beta} (\lambda.I\ R)$, onde R é o autorreprodutor $(\lambda.A\ \lambda.A)$.

Uma outra maneira de reobter a redução de sujeito é a de pensar no significado da propriedade a fim de reformulá-la de maneira adequada. As noções de expansão e restrição em contextos de tipos para o λ-calculus foram introduzida em [56], para falar de expansão e redução de sujeito, respectivamente, em um sistema de IT relevante. O Lema 5.1.7 estabelece que o sistema λ_{dB}^{SM} é relevante, logo introduzimos o conceito de restrição para contextos sequenciais a fim de verificar se a propriedade de SR pode ser reobtida.

Antes da introdução de um conceito apropriado de restrição de contextos, apresentamos alguns lemas sobre as propriedades relativas às tipagens e à substituição β. O lema a seguir apresenta a relação entre tipagens e o mecanismo para atualização de índices, sendo auxiliar ao lema de substituição apresentado mais adiante.

Lema 5.1.9 (Atualização para λ_{dB}^{SM}): Seja $M:\langle\Gamma\vdash\tau\rangle$:

1. Se $i\geq|\Gamma|$ então $M^{+i}:\langle\Gamma\vdash\tau\rangle$.

2. Se $0\leq i<|\Gamma|$, então $M^{+i}:\langle\Gamma_{\leq i}.\omega.\Gamma_{>i}\vdash\tau\rangle$.

Demonstração. Por indução na derivação de $M:\langle\Gamma\vdash\tau\rangle$:

1. Seja $i\geq|\Gamma|$. Pelo Lema 5.1.7 tem-se que $|\Gamma| = sup(M)$ logo, pelo Lema 3.1.11.1, $M^{+i}\equiv M$.

2. Seja $0\leq i < |\Gamma|$.

   - Seja $\underline{1}:\langle\tau.nil\vdash\tau\rangle$. Para $i{=}0$, $\underline{1}^{+}=\underline{2}$. Pela regra varn, $\underline{2}:\langle\omega.\tau.nil\vdash\tau\rangle$.
   - Seja $\dfrac{\underline{n}:\langle\Gamma\vdash\tau\rangle}{\underline{n{+}1}:\langle\omega.\Gamma\vdash\tau\rangle}$. Se $i{=}0$, então pela regra varn $\underline{n{+}2}:\langle\omega.\omega.\Gamma\vdash\tau\rangle$. Senão, note que $\underline{n^{+i}+1}=\underline{n{+}1}^{+(i+1)}=\underline{n{+}2}$. Assim, por HI, $\underline{n}^{+i}:\langle\Gamma_{\leq i}.\omega.\Gamma_{>i}\vdash\tau\rangle$ e, pela regra varn, $\underline{n{+}2}:\langle\omega.\Gamma_{\leq i}.\omega.\Gamma_{>i}\vdash\tau\rangle$.
   - Seja $\dfrac{M:\langle u.\Gamma\vdash\tau\rangle}{\lambda.M:\langle\Gamma\vdash u\to\tau\rangle}$. Se $0\leq i<|\Gamma|$ então $1\leq i{+}1<|\Gamma|{+}1$ logo, por HI, $M^{+(i+1)}:\langle u.\Gamma_{\leq i}.\omega.\Gamma_{>i}\vdash\tau\rangle$. Assim, pela regra $\to_i$ e definição de i-elevação, $(\lambda.M)^{+i}:\langle\Gamma_{\leq i}.\omega.\Gamma_{>i}\vdash u\to\tau\rangle$.
   - Seja $\dfrac{M_1:\langle\Gamma\vdash\omega\to\tau\rangle \qquad M_2:\langle\Delta\vdash\sigma\rangle}{(M_1\ M_2):\langle\Gamma\wedge\Delta\vdash\tau\rangle}$. Suponha s.p.d.g. que $i<|\Gamma|$ e $i\geq|\Delta|$. Assim, por HI, $M_1^{+i}:\langle\Gamma_{\leq i}.\omega.\Gamma_{>i}\vdash\omega\to\tau\rangle$ e, pelo Lema 5.1.9.1, $M_2^{+i}:\langle\Delta\vdash\sigma\rangle$. Assim, pela regra $\to_e'$, $(M_1^{+i}\ M_2^{+i}):\langle\Gamma_{\leq i}.\omega.\Gamma_{>i}\wedge\Delta\vdash\tau\rangle$. Note que $(\Gamma\wedge\Delta)_{\leq i}=\Gamma_{\leq i}\wedge\Delta$ e que $(\Gamma\wedge\Delta)_{>i}=\Gamma_{>i}$.

- Seja $\dfrac{M_1 : \langle \Gamma \vdash \wedge^n_{j=1}\sigma_j \rightarrow \tau\rangle \qquad M_2 : \langle \Delta^1 \vdash \sigma_1\rangle \ \ldots \ M_2 : \langle \Delta^n \vdash \sigma_n\rangle}{(M_1\ M_2) : \langle \Gamma \wedge \Delta^1 \wedge \cdots \wedge \Delta^n \vdash \tau\rangle}$. Pelo Lema 5.1.7, $\forall 1 \leq j \leq n$, $|\Delta^j| = sup(M_2)$. Seja $\Delta' = \Delta^1 \wedge \cdots \wedge \Delta^n$ logo $|\Delta'| = sup(M_2)$. Suponha s.p.d.g. que $i \geq |\Gamma|$ e $i < |\Delta'|$. Pelo Lema 5.1.9.1, $M_1^{+i} : \langle \Gamma \vdash \wedge^n_{j=1}\sigma_j \rightarrow \tau\rangle$ e, por HI, $M_2^{+i} : \langle \Delta^j_{\leq i}.\omega.\Delta^j_{>i} \vdash \sigma_j\rangle$, $\forall 1 \leq j \leq n$. Note que $(\Delta^1_{\leq i}.\omega.\Delta^1_{>i}) \wedge \cdots \wedge (\Delta^n_{\leq i}.\omega.\Delta^n_{>i}) = \Delta'_{\leq i}.\omega.\Delta'_{>i}$. Portanto, a demonstração é análoga a anterior. □

Lema 5.1.10 (Substituição para λ^{SM}_{dB}): Seja $M : \langle \Gamma \vdash \tau\rangle$:

1. Se $i > |\Gamma|$ então $\{\underline{i}/N\}M : \langle \Gamma \vdash \tau\rangle$, para qualquer $N \in \Lambda_{dB}$.

2. Se $\Gamma_i = \omega$ para $0 < i < |\Gamma|$, então $\{\underline{i}/N\}M : \langle \Gamma_{<i}.\Gamma_{>i} \vdash \tau\rangle$.

3. Seja $\Gamma_i = \wedge^m_{j=1}\sigma_j$ para $0 < i \leq |\Gamma|$ e $\forall 1 \leq j \leq m$, $N : \langle nil \vdash \sigma_j\rangle$. Se $sup(M) = i$, então $\{\underline{i}/N\}M : \langle \Gamma_{<i'}.nil \vdash \tau\rangle$ onde $i' = sup(\{\underline{i}/N\}M)$. Caso contrário, $\{\underline{i}/N\}M : \langle \Gamma_{<i}.\Gamma_{>i} \vdash \tau\rangle$.

4. Seja $\Gamma_i = \wedge^m_{j=1}\sigma_j$ para $0 < i \leq |\Gamma|$ e $N \in \Lambda_{dB}$ tal que $sup(N) \geq i$. Se $\forall 1 \leq j \leq m$, $N : \langle \Delta^j \vdash \sigma_j\rangle$ então $\{\underline{i}/N\}M : \langle (\Gamma_{<i}.\Gamma_{>i}) \wedge \Delta^1 \wedge \cdots \wedge \Delta^m \vdash \tau\rangle$.

Demonstração.

1. Seja $i > |\Gamma|$. Pelo Lema 5.1.7 tem-se que $|\Gamma| = sup(M)$ logo, pelo Lema 3.1.12.3, $\{\underline{i}/N\}M \equiv M$.

2. Por indução na derivação de $M : \langle \Gamma \vdash \tau\rangle$

   - Se $\underline{1} : \langle \tau.nil \vdash \tau\rangle$, nada há provar.
   - Seja $\dfrac{\underline{n} : \langle \Gamma \vdash \tau\rangle}{\underline{n+1} : \langle \omega.\Gamma \vdash \tau\rangle}$. Tem-se por hipótese que $i < |\Gamma| + 1$ e, pelo Lema 5.1.7, $|\Gamma| = n$. Logo, pela Definição 3.1.3, $\{\underline{i}/N\}\,\underline{n+1} = \underline{n}$. Se $i = 1$, nada há provar. Senão, note que $(\omega.\Gamma)_i = \Gamma_{i-1} = \omega$ e que $i - 1 < n$. Assim, por HI, $\{\underline{i-1}/N\}\,\underline{n} : \langle \Gamma_{<(i-1)}.\Gamma_{>(i-1)} \vdash \tau\rangle$ e, pela Definição 3.1.3, $\{\underline{i-1}/N\}\,\underline{n} = \underline{n-1}$. Portanto, pela regra varn, $\underline{n} : \langle \omega.\Gamma_{<(i-1)}.\Gamma_{>(i-1)} \vdash \tau\rangle$.
   - Seja $\dfrac{M : \langle u.\Gamma \vdash \tau\rangle}{\lambda.M : \langle \Gamma \vdash u \rightarrow \tau\rangle}$. Por HI, $\{\underline{i+1}/N^+\}M : \langle u.\Gamma_{<i}.\Gamma_{>i} \vdash \tau\rangle$. Assim, pela regra $\rightarrow_i$, $\lambda.\{\underline{i+1}/N^+\}M : \langle \Gamma_{<i}.\Gamma_{>i} \vdash u \rightarrow \tau\rangle$.

- Seja $\dfrac{M_1\!:\!\langle\Delta \vdash \wedge_{j=1}^{n}\sigma_j \!\rightarrow\! \tau\rangle \qquad M_2\!:\!\langle\Delta^1 \vdash \sigma_1\rangle\ \ldots\ M_2\!:\!\langle\Delta^n \vdash \sigma_n\rangle}{(M_1\ M_2)\!:\!\langle\Delta \wedge \Delta^1 \wedge \cdots \wedge \Delta^n \vdash \tau\rangle}$.
  Seja $\Delta' = \Delta^1 \wedge \cdots \wedge \Delta^n$. Observe que, pelo Lema 5.1.7, $i \neq |\Delta|$ e $i \neq |\Delta'|$. Suponha s.p.d.g. que $i > |\Delta|$ e $i < |\Delta'| = |\Delta^j|$, $\forall 1 \leq j \leq n$. Note que $\Delta'_i = \omega = \Delta^j_i$. Pelo Lema 5.1.10.1, $\{\,\underline{i}\,/N\}M_1 : \langle\Delta \vdash \wedge_{j=1}^{n}\sigma_j \!\rightarrow\! \tau\rangle$ e $\forall 1 \leq j \leq n$, por HI, $\{\,\underline{i}\,/N\}M_2\!:\!\langle\Delta^j_{<i}.\Delta^j_{>i} \vdash \sigma_j\rangle$. Assim, pela regra $\rightarrow_e$, $(\{\,\underline{i}\,/N\}M_1\ \{\,\underline{i}\,/N\}M_2)$: $\langle\Delta \wedge (\Delta'_{<i}.\Delta'_{>i}) \vdash \tau\rangle$.

- Se a última regra aplicada é $\rightarrow'_e$, a demonstração é análoga a anterior.

3. Por indução na derivação de $M\!:\!\langle\Gamma \vdash \tau\rangle$

- Se $\underline{1}\!:\!\langle\tau.nil \vdash \tau\rangle$, então $i = 1$ e, por hipótese, $\tau = \wedge_{j=1}^{m}\sigma_j$ logo $m = 1$. Pela Definição 3.1.3 tem-se $\{\,\underline{1}\,/N\}\,\underline{1} = N$, logo o resultado vale para $N\!:\!\langle nil \vdash \sigma_1\rangle$.
- Seja $\dfrac{\underline{n}\!:\!\langle\Gamma \vdash \tau\rangle}{\underline{n{+}1}\!:\!\langle\omega.\Gamma \vdash \tau\rangle}$. Pelo Lema 5.1.7 apenas $\Gamma_n \neq \omega$ e, pelo Lema 5.1.8.1, $\Gamma_n = \tau$. Então, a única possibilidade é $i = n{+}1$ logo $\tau = \wedge_{j=1}^{m}\sigma_j$. Pelo Lema 5.1.5, $\tau \in \mathcal{T}$ logo $m = 1$. Pela Definição 3.1.3, $\{\,\underline{i{+}1}\,/N\}\,\underline{i{+}1} = N$, assim, se $N\!:\!\langle nil \vdash \sigma_1\rangle$ então o resultado é válido.
- Seja $\dfrac{M\!:\!\langle u.\Gamma \vdash \tau\rangle}{\lambda.M\!:\!\langle\Gamma \vdash u \!\rightarrow\! \tau\rangle}$. Assim, para algum $\Gamma_i = \wedge_{j=1}^{m}\sigma_j$, se $\forall 1 \leq j \leq m$, N : $\langle nil \vdash \sigma_j\rangle$ então, pelo Lema 5.1.9.1, $\forall 1 \leq j \leq m$, $N^+ : \langle nil \vdash \sigma_j\rangle$. Note que $1 < i{+}1 \leq |\Gamma|{+}1$.

  Se $sup(\lambda.M) = i$ então, pelo Lema 3.1.10.2, $sup(M) = i{+}1$. Por HI tem-se $\{\,\underline{i{+}1}\,/N^+\}M : \langle(u.\Gamma)_{<i'}.nil \vdash \tau\rangle$ onde $i' = sup(\{\,\underline{i{+}1}\,/N^+\}M)$. Note que, pela descrição do conjunto $FI(\{i{+}1/N^+\}M)$, $i' = 0$ sse $FI(M) = \{\,\underline{i}\,\}$. Portanto, se $i' = 0$ então $u = \omega$, $(\omega.\Gamma)_{<i'}.nil = nil$ e, pelo Lema 3.1.10.2, $sup(\lambda.\{\,\underline{i{+}1}\,/N^+\}M) = 0$. Assim, pela regra $\rightarrow'_i$ tem-se que $\lambda.\{\,\underline{i{+}1}\,/N^+\}M$: $\langle nil \vdash \omega \!\rightarrow\! \tau\rangle$. Se $i' > 0$ então $(u.\Gamma)_{<i'} = u.\Gamma_{<(i'-1)}$ e, pelo Lema 3.1.10.2, $sup(\lambda.\{\,\underline{i{+}1}\,/N^+\}M) = i'-1$. Assim, pela regra $\rightarrow_i$ tem-se $\lambda.\{\,\underline{i{+}1}\,/N^+\}M$: $\langle\Gamma_{<(i'-1)}.nil \vdash u \!\rightarrow\! \tau\rangle$.

  Caso contrário, por HI tem-se $\{\,\underline{i{+}1}\,/N^+\}M : \langle u.\Gamma_{<i}.\Gamma_{>i} \vdash \tau\rangle$. Assim, pela regra $\rightarrow_i$, $\lambda.\{\,\underline{i{+}1}\,/N^+\}M\!:\!\langle\Gamma_{<i}.\Gamma_{>i} \vdash u \!\rightarrow\! \tau\rangle$.
- Seja $\dfrac{M_1\!:\!\langle\Delta \vdash \omega \!\rightarrow\! \tau\rangle \qquad M_2\!:\!\langle\Delta' \vdash \tau'\rangle}{(M_1\ M_2)\!:\!\langle\Delta \wedge \Delta' \vdash \tau\rangle}$. Suponha s.p.d.g. que $\Delta'_i = \omega$ logo, pela relevância do sistema, $sup(M_2) \neq i$. Assim, pelo Lema 5.1.10.2, M_2 :

$\langle \Delta'_{<i}.\Delta'_{>i} \vdash \tau' \rangle$.

Se $sup(M_1) = i$, então por HI tem-se que $\{\underline{i}/N\}M_1 : \langle \Delta_{<i'}.nil \vdash \omega \to \tau \rangle$ onde $i' = sup(\{\underline{i}/N\}M_1)$. Assim, pela regra $\to'_e$ tem-se que $\{\underline{i}/N\}(M_1\ M_2)$: $\langle (\Delta_{<i'}.nil) \wedge (\Delta'_{<i}.\Delta'_{>i}) \vdash \tau \rangle$. Note que $sup(M_1\ M_2) = sup(M_2) > i$ e $\Delta_{<i'} \wedge \Delta'_{<i} = \Delta_{<i} \wedge \Delta'_{<i}$.

Caso contrário, por HI tem-se $\{\underline{i}/N\}M_1 : \langle \Delta_{<i}.\Delta_{>i} \vdash \omega \to \tau \rangle$. Assim, pela regra $\to'_e$, $\{\underline{i}/N\}(M_1\ M_2) : \langle (\Delta \wedge \Delta')_{<i}.(\Delta \wedge \Delta')_{>i} \vdash \tau \rangle$.

- Seja $\dfrac{M_1 : \langle \Delta \vdash \wedge_{k=1}^{l} \tau_k \to \tau \rangle \qquad M_2 : \langle \Delta^1 \vdash \tau_1 \rangle \ \ldots\ M_2 : \langle \Delta^l \vdash \tau_l \rangle}{(M_1\ M_2) : \langle \Delta \wedge \Delta^1 \wedge \cdots \wedge \Delta^l \vdash \tau \rangle}$.
Seja $(\Delta \wedge \Delta^1 \wedge \cdots \wedge \Delta^l)_i = \wedge_{j=1}^{m} \sigma_j$. Suponha s.p.d.g. que $\Delta_i = \omega$, $l = m$ e $\forall 1 \leq j \leq m$, $\Delta^j_i = \sigma_j$. Assim, pelo Lema 5.1.10.2, $\{\underline{i}/N\}M_1 : \langle \Delta_{<i}.\Delta_{>i} \vdash \wedge_{k=1}^{l} \tau_k \to \tau \rangle$. Se $\forall 1 \leq j \leq m$, $N : \langle nil \vdash \sigma_j \rangle$, então por HI $\{\underline{i}/N\}M_2 : \langle \Delta^j_{<i'}.nil \vdash \tau_j \rangle$ onde $i' = sup(\{\underline{i}/N\}M_2)$ caso $sup(M_2) = i$ ou $\{\underline{i}/N\}M_2 : \langle (\Delta^j_{<i}.\Delta^j_{>i}) \vdash \tau_j \rangle$ caso contrário. A demonstração é então análoga a anterior.

4. Por indução na derivação de $M : \langle \Gamma \vdash \tau \rangle$

- Se $\underline{1} : \langle \tau.nil \vdash \tau \rangle$, então $i = 1$ e, por hipótese, $\tau = \wedge_{j=1}^{m} \sigma_j$ logo $m = 1$. Pela Definição 3.1.3 tem-se que $\{\underline{1}/N\}\,\underline{1} = N$ logo se $N : \langle \Delta^1 \vdash \sigma_1 \rangle$ então o resultado é válido. Observe que $(\sigma_1.nil)_{<1}.(\sigma_1.nil)_{>1} = nil$.
- Seja $\dfrac{\underline{n} : \langle \Gamma \vdash \tau \rangle}{\underline{n+1} : \langle \omega.\Gamma \vdash \tau \rangle}$. Pelo Lema 5.1.7 apenas $\Gamma_n \neq \omega$ e, pelo Lema 5.1.8.1, $\Gamma_n = \tau$. Assim, $i = n+1$ logo $\tau = \wedge_{j=1}^{m} \sigma_j$. Pelo Lema 5.1.5, $\tau \in \mathcal{T}$ logo $m = 1$. Pela Definição 3.1.3 tem-se $\{\underline{i+1}/N\}\,\underline{i+1} = N$. Assim, se $N : \langle \Delta^1 \vdash \sigma_1 \rangle$ tal que $|\Delta^1| \geq n+1$ então o resultado é válido. Note que $(\omega.\Gamma)_{<(n+1)}.(\omega.\Gamma)_{>(n+1)} = \omega^{\underline{n}}$ e que $\omega^{\underline{n}} \wedge \Delta^1 = \Delta^1$.
- Seja $\dfrac{M : \langle u.\Gamma \vdash \tau \rangle}{\lambda.M : \langle \Gamma \vdash u \to \tau \rangle}$. Dado $\Gamma_i = \wedge_{j=1}^{m} \sigma_j$, se $\forall 1 \leq j \leq m$, $N : \langle \Delta^j \vdash \sigma_j \rangle$ tal que $|\Delta^j| \geq i$ então, pelo Lema 5.1.9.2, $\forall 1 \leq j \leq m$, $N^+ : \langle \omega.\Delta^j \vdash \sigma_j \rangle$. Note que $\forall 1 \leq j \leq m$, $|\omega.\Delta^j| = |\Delta^j| + 1 \geq i+1$. Então, por HI tem-se que $\{\underline{i+1}/N^+\}M : \langle (u.\Gamma_{<i}.\Gamma_{>i}) \wedge (\omega.\Delta^1) \wedge \cdots \wedge (\omega.\Delta^n) \vdash \tau \rangle$. Assim, pela regra $\to_i$, $\lambda.\{\underline{i+1}/N^+\}M : \langle (\Gamma_{<i}.\Gamma_{>i}) \wedge \Delta^1 \wedge \cdots \wedge \Delta^n \vdash u \to \tau \rangle$. Note que $(u.\Gamma_{<i}.\Gamma_{>i}) \wedge (\omega.\Delta^1) \wedge \cdots \wedge (\omega.\Delta^n) = u.\big((\Gamma_{<i}.\Gamma_{>i}) \wedge \Delta^1 \wedge \cdots \wedge \Delta^n\big)$.

- Seja $\dfrac{M_1\!:\!\langle\Delta\vdash\omega\!\to\!\tau\rangle \qquad M_2\!:\!\langle\Delta'\vdash\tau'\rangle}{(M_1\ M_2)\!:\!\langle\Delta\wedge\Delta'\vdash\tau\rangle}$. Sejam $(\Delta\!\wedge\!\Delta')_i\!=\!\wedge_{j=1}^{m}\sigma_j$ e $\forall 1\!\le\! j\!\le\! m$, $N\!:\!\langle\Delta^j\vdash\sigma_j\rangle$ tal que $|\Delta^j|\!\ge\! i$. Note que $i\!\le\!|\Delta\wedge\Delta'| = max(|\Delta|,|\Delta'|)$.

  Se $i>|\Delta|$ então, pelo Lema 5.1.10.1, $\{\underline{i}/N\}M_1:\langle\Delta\vdash\omega\!\to\!\tau\rangle$ e, por HI, $\{\underline{i}/N\}M_2\!:\!\langle(\Delta'_{<i}.\Delta'_{>i})\wedge\Delta^1\wedge\cdots\wedge\Delta^n\vdash\tau'\rangle$. Pela regra $\to'_e$, $\{\underline{i}/N\}(M_1\ M_2)$: $\langle(\Delta\wedge\Delta'_{<i}.\Delta'_{>i})\wedge\Delta^1\wedge\cdots\wedge\Delta^n\vdash\tau\rangle$. Observe que $(\Delta\wedge\Delta')_{<i}\!=\!\Delta\wedge\Delta'_{<i}$ e que $(\Delta\wedge\Delta')_{>i}\!=\!\Delta'_{>i}$. A prova para $i>|\Delta'|$ é similar.

  Sejam $|\Delta|,|\Delta'|\ge i$. Se $\Delta'_i=\omega$ ou $\Delta_i=\omega$ então a prova é similar à acima, usando o Lema 5.1.10.2. Senão, sejam $\Delta_i=\wedge_{k=1}^{l}\sigma_{j_k}$ e $\Delta'_k=\wedge_{k=1}^{l'}\sigma_{j'_k}$. Por HI tem-se que $\{\underline{i}/N\}M_1\!:\!\langle(\Delta_{<i}.\Delta_{>i})\wedge\Delta^{j_1}\wedge\cdots\wedge\Delta^{j_l}\vdash\omega\!\to\!\tau\rangle$ e $\{\underline{i}/N\}M_2$: $\langle(\Delta'_{<i}.\Delta'_{>i})\wedge\Delta^{j'_1}\wedge\cdots\wedge\Delta^{j'_{l'}}\vdash\tau'\rangle$. Note que j_k e j'_k percorrem juntos cada $1\le j\!\le\! m$. Assim, $\{\underline{i}/N\}(M_1\ M_2)\!:\!\langle\big((\Delta\wedge\Delta')_{<i}.(\Delta\wedge\Delta')_{>i}\big)\wedge\Delta^1\wedge\cdots\wedge\Delta^m\vdash\tau\rangle$ pela regra $\to'_e$.

- Seja $\dfrac{M_1\!:\!\langle\Delta\vdash\wedge_{k=1}^{l}\tau_k\!\to\!\tau\rangle \qquad M_2\!:\!\langle\Delta^1\vdash\tau_1\rangle\ \ldots\ M_2\!:\!\langle\Delta^l\vdash\tau_l\rangle}{(M_1\ M_2)\!:\!\langle\Delta\wedge\Delta^1\wedge\cdots\wedge\Delta^l\vdash\tau\rangle}$.
  A demonstração é análoga a anterior. Note que $\Delta^j=\omega$ para algum $1\le j\le l$ sse $\Delta^j\!=\!\omega$ para todo j. □

Observe que, para uma substituição proveniente de uma redução β, a condição presente no item 4 acima, de que $sup(N)\ge i$, é satisfeita para qualquer N tal que $sup(N)\!>\!0$. A seguir, definimos a restrição para contextos, usada para verificar se o sistema possui SR.

Definição 5.1.11 (Restrição para contextos): Seja $\Gamma\!\downharpoonright_M$ um contexto $\Gamma'\sqsubseteq\Gamma$ tal que $|\Gamma'|=sup(M)$ e que $\forall 1\!\le\! i\!\le\!|\Gamma'|$, $\Gamma'_i\neq\omega$ sse $\underline{i}\!\in\! FI(M)$.

Observe que a restrição não é unicamente definida, porque a maneira como as interseções são particionadas pode variar. Por exemplo, $(\alpha\wedge\beta.nil)\!\downharpoonright_{\underline{1}}=\alpha.nil,\beta.nil$ ou ao próprio contexto. Propriedades dessa restrição são apresentadas a seguir.

Lema 5.1.12: 1. $\Gamma\!\downharpoonright_M= nil$, para qualquer contexto Γ e termo fechado M.

2. Sejam M e M' termos tais que $FI(M)=FI(M')$. Se $M\!:\!\langle\Gamma\vdash\tau\rangle$, então $(\Gamma\!\wedge\!\Delta)\!\downharpoonright_{M'}=\Gamma$, para qualquer contexto Δ.

Demonstração. Em ambos os casos, as propriedades seguem da Definição 5.1.11. □

Teorema 5.1.13 (SR para contração $\boldsymbol{\beta}$ em λ^{SM}_{dB}): Se $(\lambda.M\ M'):\langle\Gamma\vdash_{\lambda^{SM}_{dB}}\tau\rangle$, então $\{\underline{1}/M'\}M:\langle\Gamma\lfloor_{\{\underline{1}/M'\}M}\vdash_{\lambda^{SM}_{dB}}\tau\rangle$.

Demonstração. Por indução na derivação de $(\lambda.M\ M'):\langle\Gamma\vdash\tau\rangle$:

- Seja $\dfrac{\lambda.M:\langle\Gamma\vdash\omega\to\tau\rangle \qquad M':\langle\Delta\vdash\sigma\rangle}{(\lambda.M\ M'):\langle\Gamma\wedge\Delta\vdash\tau\rangle}$.

  Se $\Gamma=nil$, então pelo Lema 5.1.8.3 tem-se que $M:\langle nil\vdash\tau\rangle$. Assim, pelo Lema 5.1.10.1, $\{\underline{1}/M'\}M:\langle nil\vdash\tau\rangle$. Observe que $FI(\{\underline{1}/M'\}M)=FI(M)=\emptyset$ logo $(nil\wedge\Delta)\lfloor_{\{\underline{1}/M'\}M}=nil$.

  Caso contrário, pelo Lema 5.1.8.4 tem-se que $M:\langle\omega.\Gamma\vdash\tau\rangle$. Assim, pelo Lema 5.1.10.2, $\{\underline{1}/M'\}M:\langle\Gamma\vdash\tau\rangle$. Note que $FI(\{\underline{1}/M'\}M)=FI(\lambda.M)$ logo $(\Gamma\wedge\Delta)\lfloor_{\{\underline{1}/M'\}M}=\Gamma$.

- Seja $\dfrac{\lambda.M:\langle\Gamma\vdash\wedge^n_{i=1}\sigma_i\to\tau\rangle \qquad M':\langle\Delta^1\vdash\sigma_1\rangle\ \ldots\ M':\langle\Delta^n\vdash\sigma_n\rangle}{(\lambda.M\ M'):\langle\Gamma\wedge\Delta^1\wedge\cdots\wedge\Delta^n\vdash\tau\rangle}$.

  Pelo Lema 5.1.8.4 tem-se que $M:\langle\wedge^n_{i=1}\sigma_i.\Gamma\vdash\tau\rangle$.

  Se $sup(M')=0$ então $\forall 1\le i\le n$, $\Delta^i=nil$. Se $sup(M)=1$ então pelo Lema 3.1.10.2 $sup(\lambda.M)=0$ e, pelo Corolário 3.1.13, $FI(\{\underline{1}/M'\}M)=FI(\lambda.M\ M')=\emptyset$. Assim, pelo Lema 5.1.10.3, $\{\underline{1}/M'\}M:\langle nil\vdash\tau\rangle$. Senão, pelo Lema 5.1.10.3, $\{\underline{1}/M'\}M:\langle\Gamma\vdash\tau\rangle$. Note que $FI(\{\underline{1}/M'\}M)=FI(\lambda.M)$ logo $(\Gamma\wedge nil)\lfloor_{\{\underline{1}/M'\}M}=\Gamma$.

  Se $sup(M')>0$ então, pelo Lema 5.1.10.4, $\{\underline{1}/M'\}M:\langle\Gamma\wedge\Delta^1\wedge\cdots\wedge\Delta^n\vdash\tau\rangle$. $\square$

Observe que a informação perdida do contexto é toda e apenas a informação de tipo sobre os índices livres do termo que desaparece após a contração β. Essa informação perdida reflete no tipo do termo, quando a redução β é analisada.

Exemplo 5.1.4: Sejam $M\equiv\lambda.(\lambda.\underline{2}\ \underline{1})$ e $M'\equiv\lambda.\underline{1}$. Temos que $M:\langle nil\vdash(\alpha\wedge\beta)\to\alpha\rangle$, $M\to_\beta M'$ e que $M':\langle nil\vdash\alpha\to\alpha\rangle$.

5.2 O sistema $\lambda^{\sqcap}_{dB}$

Nesta seção apresentamos a versão à la de Bruijn do sistema de tipos com interseção de Kamareddine e Nour introduzido em [56]. Para ser completo em relação a semântica de realização proposta, esse sistema tem a interseção como um construtor geral para tipos e uma relação de subtipos. Além disso, o ω é um tipo universal. Apesar das definições para os conjuntos $\mathcal{T}'$ e $\mathcal{U}'$ serem semelhantes às apresentadas na Definição 5.1.1.1, a interseção, denotada por $\sqcap$ no lugar do $\wedge$, é idempotente. Assim, definimos a seguir o conjunto de tipos com interseção restrita idempotente.

Definição 5.2.1: 1. Seja $\mathcal{A}$ o conjunto de variáveis de tipo. O conjunto de **tipos com interseção restrita idempotente** é definido por:

$$\tau, \sigma \in \mathcal{T}' ::= \mathcal{A} \,|\, \mathcal{U}' \to \mathcal{T}' \qquad\qquad u, v \in \mathcal{U}' ::= \omega \,|\, \mathcal{U}' \sqcap \mathcal{U}' \,|\, \mathcal{T}'$$

Os tipos são considerados módulo comutatividade, associatividade, idempotência e ω o elemento neutro de $\sqcap$.

2. Os contextos de tipos são sequências de elementos $u \in \mathcal{U}'$.

Dado um termo M, ω^M denota o contexto omega $\omega^{\underline{m}}$ tal que $m = sup(M)$.

A extensão de $\sqcap$ para contextos é direta, tendo nil como elemento neutro. Assim, $\sqcap$ é comutativa, associativa e idempotente para contextos.

Análogo ao sistema λ^{SM}_{dB}, a regra var é restrita a tipos em $\mathcal{T}'$ e a regra $\to_e$ introduz a interseção para tipos nos contextos. Diferentemente desse sistema, uma regra $\to'_e$ não é necessária, pois o ω pode ser atribuído a qualquer termo através da regra (ω) de inferência. Além disso, tipos com interseção podem ser atribuídos a termos, através da regra $\sqcap_i$. A eliminação da interseção é feita através da relação de ordem $\leqslant$ para os tipos. A seguir, apresentamos a definição do sistema $\lambda^{\sqcap}_{dB}$.

Definição 5.2.2 (O sistema $\lambda^{\sqcap}_{dB}$): As regras de tipagem do sistema $\lambda^{\sqcap}_{dB}$ são dadas por:

$$\frac{\tau \in \mathcal{T}'}{\underline{1} : \langle \tau.nil \vdash \tau \rangle}\ \text{var} \qquad\qquad \frac{M : \langle nil \vdash \tau \rangle}{\lambda.M : \langle nil \vdash \omega \to \tau \rangle}\ \to'_i$$

$$\frac{\underline{n} : \langle \Gamma \vdash u \rangle}{\underline{n+1} : \langle \omega.\Gamma \vdash u \rangle}\ \text{varn} \qquad\qquad \frac{M_1 : \langle \Gamma \vdash u \to \tau \rangle \qquad M_2 : \langle \Gamma' \vdash u \rangle}{(M_1\ M_2) : \langle \Gamma \sqcap \Gamma' \vdash \tau \rangle}\ \to_e$$

$$\frac{}{M : \langle \omega^M \vdash \omega \rangle}\ \omega \qquad\qquad \frac{M : \langle \Gamma \vdash u_1 \rangle \qquad M : \langle \Gamma \vdash u_2 \rangle}{M : \langle \Gamma \vdash u_1 \sqcap u_2 \rangle}\ \sqcap_i$$

$$\frac{M : \langle u.\Gamma \vdash \tau \rangle}{\lambda.M : \langle \Gamma \vdash u \to \tau \rangle}\ \to_i \qquad\qquad \frac{M : \langle \Gamma \vdash u \rangle \qquad \langle \Gamma \vdash u \rangle \leqslant \langle \Gamma' \vdash u' \rangle}{M : \langle \Gamma' \vdash u' \rangle}\ \leqslant$$

onde a relação binária $\leqslant$ é definida como segue:

$$\frac{}{\Phi \leqslant \Phi}\ \text{ref} \qquad \frac{\Phi_1 \leqslant \Phi_2 \qquad \Phi_2 \leqslant \Phi_3}{\Phi_1 \leqslant \Phi_3}\ \text{tr}$$

$$\frac{}{u_1 \sqcap u_2 \leqslant u_1}\ \sqcap_e \qquad \frac{u_1 \leqslant v_1 \qquad u_2 \leqslant v_2}{u_1 \sqcap u_2 \leqslant v_1 \sqcap v_2}\ \sqcap$$

$$\frac{u_2 \leqslant u_1 \qquad \tau_1 \leqslant \tau_2}{u_1 \rightarrow \tau_1 \leqslant u_2 \rightarrow \tau_2}\ \rightarrow \qquad \frac{u_1 \leqslant u_2}{\Gamma_{\leq i}.u_1.\Gamma_{>i} \leqslant \Gamma_{\leq i}.u_2.\Gamma_{>i}}\ \leqslant_c$$

$$\frac{u_1 \leqslant u_2 \qquad \Gamma' \leqslant \Gamma}{\langle \Gamma \vdash u_1 \rangle \leqslant \langle \Gamma' \vdash u_2 \rangle}\ \leqslant_{\langle\rangle}$$

Φ, Φ', $\Phi_1, \ldots$ são usados para denotar $u \in \mathcal{U}'$, contextos Γ ou tipagens $\langle \Gamma \vdash u \rangle$. Observe que em $\Phi \leqslant \Phi'$, Φ e Φ' pertencem a mesma classe de objetos.

Similar aos contextos com tipos em $\mathcal{U}$, algumas propriedades sobre os contextos com tipos em $\mathcal{U}'$ seguem das respectivas definições.

Lema 5.2.3: Sejam Γ e Δ contextos diferentes de *nil*:

1. Se $|\Gamma| \geq sup(M)$, então $\Gamma \sqcap env^M_\omega = \Gamma$
2. $\Gamma \sqcap \Delta = (\Gamma_1 \sqcap \Delta_1).(\Gamma_{>1} \sqcap \Delta_{>1})$
3. Se $i \leq |\Gamma|, |\Delta|$, então $(\Gamma \sqcap \Delta)_i = \Gamma_i \sqcap \Delta_i$.
4. $(\Gamma \sqcap \Delta)_{<i} = \Gamma_{<i} \sqcap \Delta_{<i}$ e $(\Gamma \sqcap \Delta)_{>i} = \Gamma_{>i} \sqcap \Delta_{>i}$. O mesmo para $(\Gamma \sqcap \Delta)_{\leq i}$ e $(\Gamma \sqcap \Delta)_{\geq i}$.
5. $|\Gamma \sqcap \Delta| = max(|\Gamma|, |\Delta|)$.

Os próximos lemas descrevem propriedades sobre a forma dos tipos e contextos, e suas ligações com a relação de subtipos definidas por $\leqslant$. Como os conjunto de tipos e a relação de ordem são idênticos aos do sistema $\lambda^{\sqcap}$, descritos na Seção 2.5.2, as demonstrações são exatamente as mesmas das apresentadas em [56].

Lema 5.2.4: 1. Se $u \in \mathcal{U}'$, então $u = \omega$ ou $u = \sqcap_{i=1}^n \tau_i$ onde $n \geq 1$ e $\forall\, 1 \leq i \leq n$, $\tau_i \in \mathcal{T}'$.

2. $u \leqslant \omega$.

3. Se $\omega \leqslant u$, então $u = \omega$.

Demonstração. Veja [56]. □

Observe que, a partir da tipagem $\underline{2} : \langle \omega.\tau.nil \vdash \tau\rangle$ e da relação $\leqslant$ temos que, para qualquer $u \in \mathcal{U}'$, $\underline{2}:\langle u.\tau.nil \vdash \tau\rangle$. Portanto, uma forma de lei de redundância é permitida no sistema de tipos , o que não acontece no sistema original introduzido em [56]. Isso se deve ao fato de ω's extras serem necessários para a construção do contexto sequencial apropriado para a designação correta de um tipo a um índice livre $\underline{i}$. Apesar disso, no Lema 5.2.7 prova-se que essa redundância é limitada pelo próprio termo a ser tipado, através de seu *sup*.

Lema 5.2.5: Seja $v \in \mathcal{U}'$ tal que $v \neq \omega$.

1. Se $u \leqslant v$, então $u = \sqcap_{j=1}^{k}\tau_j$, $v = \sqcap_{i=1}^{p}\tau'_i$ onde $p, k \geq 1$, $\forall 1 \leq j \leq k$, $1 \leq i \leq p$, $\tau_j, \tau'_i \in \mathcal{T}'$, e $\forall\, 1 \leq i \leq p$, $\exists 1 \leq j \leq k$ tal que $\tau_j \leqslant \tau'_i$.
2. Se $u \leqslant v' \sqcap \alpha$, então $u = u' \sqcap \alpha$ e $u' \leqslant v'$.
3. Sejam $p, k \geq 1$. Se $\sqcap_{j=1}^{k}(u_j \to \tau_j) \leqslant \sqcap_{i=1}^{p}(u'_i \to \tau'_i)$, então $\forall\, 1 \leq i \leq p$, $\exists 1 \leq j \leq k$ tal que $u'_i \leqslant u_j$ e $\tau_j \leqslant \tau'_i$.
4. Se $u \to \tau \leqslant v$, então $v = \sqcap_{i=1}^{p}(u_i \to \tau_i)$ onde $p \geq 1$ e $\forall\, 1 \leq i \leq p$, $u_i \leqslant u$ e $\tau \leqslant \tau_i$.
5. Se $\sqcap_{j=1}^{k}(u_j \to \tau_j) \leqslant v$ onde $k \geq 1$, então $v = \sqcap_{i=1}^{p}(u'_i \to \tau'_i)$ onde $p \geq 1$ e $\forall\, 1 \leq i \leq p$, $\exists 1 \leq j \leq k$ tal que $u'_i \leqslant u_j$ and $\tau_j \leqslant \tau'_i$.

Demonstração. Veja [56] □

Lema 5.2.6:

1. Se $\Gamma \leqslant \Gamma'$ e $u \leqslant u'$, então $u.\Gamma \leqslant u'.\Gamma'$.
2. $\Gamma \leqslant \Gamma'$ sse $|\Gamma| = |\Gamma'| = m$ e, se $m > 0$ então $\forall 1 \leq i \leq m$, $\Gamma_i \leqslant \Gamma'_i$.
3. Se $|\Gamma| = sup(M)$, então $\Gamma \leqslant \omega^M$.
4. Se $\omega^M \leqslant \Gamma$, então $\Gamma = \omega^M$.
5. $\langle \Gamma \vdash u\rangle \leqslant \langle \Gamma' \vdash u'\rangle$ sse $\Gamma' \leqslant \Gamma$ e $u \leqslant u'$.
6. Se $\Gamma \leqslant \Gamma'$ e $\Delta \leqslant \Delta'$, então $\Gamma \sqcap \Delta \leqslant \Gamma' \sqcap \Delta'$.

Demonstração. 1. Por indução na derivação de $\Gamma \leqslant \Gamma'$ tem-se que se $\Gamma \leqslant \Gamma'$, então $v.\Gamma \leqslant v.\Gamma'$. Portanto, com a regra tr obtém-se o resultado.

2. *Somente se*) Por indução na derivação de $\Gamma \leqslant \Gamma'$. *Se*) Por indução em m, usando o item 1 acima.

3. Pelo Lema 5.2.4.2 e o item 2.

4. Pelo item 2, $|\Gamma| = sup(M) = m$. Se $m = 0$, então $\omega^M = \Gamma = nil$. Senão, para cada $1 \leq i \leq m$, $\omega \leqslant \Gamma_i$. Portanto, pelo Lema 5.2.4.3, $\forall 1 \leq i \leq m$, $\Gamma_i = \omega$.

5. *Somente se*) Por indução na derivação de $\langle \Gamma \vdash u \rangle \leqslant \langle \Gamma' \vdash u' \rangle$. *Se*) Pela regra $\leqslant_{\langle\rangle}$.

6. Esse é um corolário do item 2. □

5.2.1 Propriedades

O lema a seguir mostra a relação em uma tipagem $\langle \Gamma \vdash u \rangle$ de M em $\lambda_{dB}^{\sqcap}$, entre o comprimento de Γ e os índices livres de M.

Lema 5.2.7 (Redundância para $\lambda_{dB}^{\sqcap}$):

1. Se $M : \langle \Gamma \vdash u \rangle$, então $|\Gamma| = sup(M)$.

2. Para todo Γ e M tais que $|\Gamma| = sup(M)$, tem-se que $M : \langle \Gamma \vdash \omega \rangle$.

Demonstração. 1. Por indução na derivação de $M : \langle \Gamma \vdash u \rangle$.

2. Pela regra ω, $M : \langle \omega^M \vdash \omega \rangle$. Pelo Lema 5.2.6.3, $\Gamma \leqslant \omega^M$. Portanto, por $\leqslant_{\langle\rangle}$ e $\leqslant$, $M : \langle \Gamma \vdash \omega \rangle$. □

Consequentemente, a redundância permitida no sistema $\lambda_{dB}^{\sqcap}$ é limitada pelo índice ocorrendo livre no termo com o valor máximo.

O lema a seguir mostra que outra versão das regras var e $\sqcap_i$, axioma e introdução da interseção respectivamente, são deriváveis a partir da regras de inferência de tipagem e de relação de subtipos, apresentadas na Definição 5.2.2.

Lema 5.2.8 (Regras admissíveis em $\lambda_{dB}^{\sqcap}$):

1. A regra $\sqcap_i'$ é admissível em $\lambda_{dB}^{\sqcap}$.

$$\frac{M:\langle\Gamma\vdash u_1\rangle \qquad M:\langle\Delta\vdash u_2\rangle}{M:\langle\Gamma\sqcap\Delta\vdash u_1\sqcap u_2\rangle}\ \sqcap_i'$$

2. A regra var$'$ é admissível em $\lambda_{dB}^{\sqcap}$.

$$\frac{u\in\mathcal{U}'}{\underline{1}:\langle u.nil\vdash u\rangle}\ \text{var}'$$

Demonstração. 1. Suponha que $M:\langle\Gamma\vdash u_1\rangle$ e $M:\langle\Delta\vdash u_2\rangle$. Pelo Lema 5.2.7.1, $|\Gamma|=|\Delta|=m$. Assim, $|\Gamma\sqcap\Delta|=m$ e $(\Gamma\sqcap\Delta)_i=\Gamma_i\sqcap\Delta_i$, $\forall 1\leq i\leq m$. Pela regra $\sqcap_e$ e o Lema 5.2.6.2, $\Gamma\sqcap\Delta\leqslant\Gamma$ e $\Gamma\sqcap\Delta\leqslant\Delta$. Logo, pelas regras $\leqslant_{\langle\rangle}$ e $\leqslant$, $M:\langle\Gamma\sqcap\Delta\vdash u_1\rangle$ e $M:\langle\Gamma\sqcap\Delta\vdash u_2\rangle$. Portanto, pela regra $\sqcap_i$, $M:\langle\Gamma\sqcap\Delta\vdash u_1\sqcap u_2\rangle$.

2. Pelo Lema 5.2.4.1 tem-se que $u=\omega$ ou $u=\sqcap_{i=1}^{k}\tau_i$ onde $\forall 1\leq i\leq k$, $\tau_i\in\mathcal{T}'$:

- Se $u=\omega$, então pela regra ω obtém-se o resultado.
- Se $u=\sqcap_{i=1}^{k}\tau_i$ então, pela regra var, $\underline{1}:\langle\tau_i.nil\vdash\tau_i\rangle$ e, por $k-1$ aplicações da regra$\sqcap_i'$, $\underline{1}:\langle u.nil\vdash u\rangle$. □

Lema 5.2.9 (Geração para $\lambda_{dB}^{\sqcap}$):

1. Se $\underline{n}:\langle\Gamma\vdash u\rangle$, então $\Gamma_n=v$ onde $v\leqslant u$.

2. Se $\lambda.M:\langle\Gamma\vdash u\rangle$ e $sup(M)>0$, então $u=\omega$ ou $u=\sqcap_{i=1}^{k}(v_i\rightarrow\tau_i)$ onde $k\geq 1$ e $\forall 1\leq i\leq k$, $M:\langle v_i.\Gamma\vdash\tau_i\rangle$.

3. Se $\lambda.M:\langle\Gamma\vdash u\rangle$ e $sup(M)=0$, então $\Gamma=nil$, $u=\omega$ ou $u=\sqcap_{i=1}^{k}(v_i\rightarrow\tau_i)$ onde $k\geq 1$ e $\forall 1\leq i\leq k$, $M:\langle nil\vdash\tau_i\rangle$.

Demonstração. 1. Por indução na derivação de $\underline{n}:\langle\Gamma\vdash u\rangle$. Pelo Lema 5.2.7.1, $|\Gamma|=n$.

- Se $\underline{1}:\langle\tau.nil\vdash\tau\rangle$, nada há provar.
- Se $\underline{n}:\langle\omega^{\underline{n}}\vdash\omega\rangle$, nada há provar.
- Seja $\dfrac{\underline{n}:\langle\Gamma\vdash u\rangle}{\underline{n+1}:\langle\omega.\Gamma\vdash u\rangle}$. Tem-se que $(\omega.\Gamma)_{n+1}=\Gamma_n$ e, por HI, $\Gamma_n=v$ onde $v\leqslant u$.
- Seja $\dfrac{\underline{n}:\langle\Gamma\vdash u_1\rangle \qquad \underline{n}:\langle\Gamma\vdash u_2\rangle}{\underline{n}:\langle\Gamma\vdash u_1\sqcap u_2\rangle}$. Por HI, $\Gamma_n=v$ onde $v\leqslant u_1$ e $v\leqslant u_2$. Portanto, pela regra $\sqcap$, $v\leqslant u_1\sqcap u_2$.

- Seja $\dfrac{\underline{n}:\langle\Gamma\vdash u\rangle \qquad \langle\Gamma\vdash u\rangle\leqslant\langle\Gamma'\vdash u'\rangle}{\underline{n}:\langle\Gamma'\vdash u'\rangle}$. Por HI, $\Gamma_n = v$ onde $v\leqslant u$. Pelo Lema 5.2.6.5, $\Gamma'\leqslant\Gamma$ e $u\leqslant u'$. Assim, pelo Lema 5.2.6.2, $\Gamma'_n = v'\leqslant v$. Portanto, pela regra tr, $v'\leqslant u'$.

2. Por indução na derivação de $\lambda.M:\langle\Gamma\vdash u\rangle$.

   - Se $\lambda.M:\langle\omega^{\lambda.M}\vdash\omega\rangle$, nada há provar.
   - Se $\dfrac{M:\langle u.\Gamma\vdash\tau\rangle}{\lambda.M:\langle\Gamma\vdash u\to\tau\rangle}$, nada há provar.
   - Seja $\dfrac{\lambda.M:\langle\Gamma\vdash u_1\rangle \qquad \lambda.M:\langle\Gamma\vdash u_2\rangle}{\lambda.M:\langle\Gamma\vdash u_1\sqcap u_2\rangle}$. Por HI, tem-se o casos a seguir:
     - Se $u_1 = u_2 = \omega$, então $u_1\sqcap u_2 = \omega$.
     - Se $u_1 = \omega$, $u_2 = \sqcap_{i=1}^{k}(v_i\to\tau_i)$ onde $k\geq 1$ e $\forall 1\leq i\leq k$, $M:\langle v_i.\Gamma\vdash\tau_i\rangle$, então $u_1\sqcap u_2 = u_2$
     - Se $u_2 = \omega$, $u_1 = \sqcap_{i=1}^{k}(v'_i\to\tau'_i)$ onde $k\geq 1$ e $\forall 1\leq i\leq k$, $M:\langle v'_i.\Gamma\vdash\tau'_i\rangle$, então $u_1\sqcap u_2 = u_1$
     - Se $u_1 = \sqcap_{i=1}^{k}(v_i\to\tau_i)$,$u_2 = \sqcap_{i=k+1}^{k+l}(v_i\to\tau_i)$, onde $k,l\geq 1$ e $\forall 1\leq i\leq k+l$, $M:\langle v_i.\Gamma\vdash\tau_i\rangle$, então $u_1\sqcap u_2 = \sqcap_{i=1}^{k+l}(v_i\to\tau_i)$.
   - Seja $\dfrac{\lambda.M:\langle\Gamma\vdash u\rangle \qquad \langle\Gamma\vdash u\rangle\leqslant\langle\Gamma'\vdash u'\rangle}{\lambda.M:\langle\Gamma'\vdash u'\rangle}$. Pelo Lema 5.2.6.5, $\Gamma'\leqslant\Gamma$ e $u\leqslant u'$. Por HI, tem-se os seguintes casos:
     - Se $u=\omega$ então, pelo Lema 5.2.4.3, $u'=\omega$.
     - Senão, $u = \sqcap_{i=1}^{k}(v_i\to\tau_i)$ onde $k\geq 1$ e $\forall 1\leq i\leq k$, $M:\langle v_i.\Gamma\vdash\tau_i\rangle$. Pelo Lema 5.2.4.1, tem-se que ou $u'=\omega$, logo nada há provar, ou, juntamente com o Lema 5.2.5.5, $u' = \sqcap_{i=1}^{p}(v'_i\to\tau'_i)$ onde $p\geq 1$ e $\forall\, 1\leq i\leq p$, $\exists 1\leq j_i\leq k$ tal que $v'_i\leqslant v_{j_i}$ e $\tau_{j_i}\leqslant\tau'_i$. Para cada $1\leq i\leq p$, pelos Lemas 5.2.6.1 e 5.2.6.5, tem-se que $\langle v_{j_i}.\Gamma\vdash\tau_{j_i}\rangle\leqslant\langle v'_i.\Gamma'\vdash\tau'_i\rangle$ logo $M:\langle v'_i.\Gamma'\vdash\tau'_i\rangle$.

3. Pelo Lema 3.1.10.2 tem-se que $sup(\lambda.M) = 0$ logo, pelo Lema 5.2.7.1, $\Gamma = nil$. Portanto, $\lambda.M:\langle nil\vdash u\rangle$. A demonstração é similar a do item 2 acima, onde $\to'_i$ é usado no passo indutivo ao invés de $\to_i$. $\square$

5.2.2 Redução de sujeito

A propriedade de redução de sujeito é provada de maneira padrão, com o lema de geração já provado (Lema 5.2.9) e um lema de substituição a ser provado (Lema 5.2.11). Um lema auxiliar será demonstrado a seguir, apresentando a releção entre tipagens e o mecanismo de atualização de índices.

Lema 5.2.10 (Atualização para $\lambda^{\sqcap}_{dB}$): Se $M : \langle\Gamma \vdash u\rangle$ e $0 \leq i < sup(M)$, então $M^{+i} : \langle\Gamma_{\leq i}.\omega.\Gamma_{>i} \vdash u\rangle$.

Demonstração. Por indução na derivação de $M : \langle\Gamma \vdash u\rangle$.

- Seja $\underline{1} : \langle\tau.nil \vdash \tau\rangle$. Para $i = 0$ tem-se que $\underline{1}^{+} = \underline{2}$ e, por varn, $\underline{2} : \langle\omega.\tau.nil \vdash \tau\rangle$.
- Se $M : \langle\omega^{M} \vdash \omega\rangle$, nada há provar.
- Seja $\dfrac{\underline{n} : \langle\Gamma \vdash u\rangle}{\underline{n{+}1} : \langle\omega.\Gamma \vdash u\rangle}$. Se $i{=}0$ então, pela regra varn, $\underline{n{+}2} : \langle\omega.\omega.\Gamma \vdash u\rangle$. Senão, note que $\underline{n^{+i} + 1} = \underline{n{+}1}^{+(i+1)} = \underline{n{+}2}$. Por HI tem-se $\underline{n}^{+i} : \langle\Gamma_{\leq i}.\omega.\Gamma_{>i} \vdash u\rangle$. Portanto, pela regra varn, $\underline{n{+}2} : \langle\omega.\Gamma_{\leq i}.\omega.\Gamma_{>i} \vdash u\rangle$.
- Seja $\dfrac{M : \langle u.\Gamma \vdash \tau\rangle}{\lambda.M : \langle\Gamma \vdash u \to \tau\rangle}$. Pelo Lema 3.1.10.2 tem-se que $sup(M) > i{+}1$ logo, por HI, $M^{+(i+1)} : \langle u.\Gamma_{\leq i}.\omega.\Gamma_{>i} \vdash \tau\rangle$. Portanto, pela regra $\to_i$ e a definição de i-elevação, $(\lambda.M)^{+i} : \langle\Gamma_{\leq i}.\omega.\Gamma_{>i} \vdash u \to \tau\rangle$.
- Seja $\dfrac{M_1 : \langle\Gamma \vdash u \to \tau\rangle \qquad M_2 : \langle\Delta \vdash u\rangle}{(M_1\ M_2) : \langle\Gamma \sqcap \Delta \vdash \tau\rangle}$. Pelo Lema 3.1.10.1 tem-se que $sup(M_1) > i$ ou $sup(M_2) > i$. Suponha s.p.d.g. que $i < sup(M_1), sup(M_2)$. Por HI, M_1^{+i} : $\langle\Gamma_{\leq i}.\omega.\Gamma_{>i} \vdash u \to \tau\rangle$ e $M_2^{+i} : \langle\Delta_{\leq i}.\omega.\Delta_{>i} \vdash u\rangle$. Assim, pela regra $\to_e$ e observando que $(\Gamma_{\leq i}.\omega.\Gamma_{>i}) \sqcap (\Delta_{\leq i}.\omega.\Delta_{>i}) = (\Gamma \sqcap \Delta)_{\leq i}.\omega.(\Gamma \sqcap \Delta)_{>i}$, tem-se que $(M_1\ M_2)^{+i}$: $\langle(\Gamma \sqcap \Delta)_{\leq i}.\omega.(\Gamma \sqcap \Delta)_{>i} \vdash \tau\rangle$.
- Seja $\dfrac{M : \langle\Gamma \vdash u_1\rangle \qquad M : \langle\Gamma \vdash u_2\rangle}{M : \langle\Gamma \vdash u_1 \sqcap u_2\rangle}$. Por HI tem-se que $M^{+i} : \langle\Gamma_{\leq i}.\omega.\Gamma_{>i} \vdash u_1\rangle$ e $M^{+i} : \langle\Gamma_{\leq i}.\omega.\Gamma_{>i} \vdash u_2\rangle$. Portanto, pela regra $\sqcap_i$, $M^{+i} : \langle\Gamma_{\leq i}.\omega.\Gamma_{>i} \vdash u_1 \sqcap u_2\rangle$.
- Seja $\dfrac{M : \langle\Gamma \vdash u\rangle \qquad \langle\Gamma \vdash u\rangle \leqslant \langle\Gamma' \vdash u'\rangle}{M : \langle\Gamma' \vdash u'\rangle}$. Por HI, $M^{+i} : \langle\Gamma_{\leq i}.\omega.\Gamma_{>i} \vdash u\rangle$ e, pelo Lema 5.2.6.5, $\Gamma' \leqslant \Gamma$ e $u \leqslant u'$. Assim, pelo Lema 5.2.6.2, $\Gamma'_{\leq i}.\omega.\Gamma'_{>i} \leqslant \Gamma_{\leq i}.\omega.\Gamma_{>i}$. Portanto, pelas regras $\leqslant_{\langle\rangle}$ e $\leqslant$, $M^{+i} : \langle\Gamma'_{\leq i}.\omega.\Gamma'_{>i} \vdash u'\rangle$. $\square$

Lema 5.2.11 (Substituição para $\lambda^{\sqcap}_{dB}$): Sejam $M:\langle\Gamma\vdash u\rangle$ e $i\in\mathbb{N}^*$, para $i\leq sup(M)$, e $N:\langle\Delta\vdash\Gamma_i\rangle$:

1. Se $\underline{i}\notin FI(M)$, então $\{\,\underline{i}\,/N\}M:\langle\Gamma_{<i}.\Gamma_{>i}\vdash u\rangle$.

2. Senão, se $sup(N)\geq i-1$, então $\{\,\underline{i}\,/N\}M:\langle(\Gamma_{<i}.\Gamma_{>i})\sqcap\Delta\vdash u\rangle$.

Demonstração. Por indução na derivação de $M:\langle\Gamma\vdash u\rangle$.

1. Observe que $i<|\Gamma|=sup(M)$:

   - Se $\underline{1}:\langle\tau.nil\vdash\tau\rangle$, nada há provar.
   - Seja $M:\langle\omega^M\vdash\omega\rangle$. Pelo Lema 3.1.14.1, $sup(\{\,\underline{i}\,/N\}M)=sup(M)-1$. Assim, $\omega^{\{\,\underline{i}\,/N\}M}=(\omega^M)_{<i}.(\omega^M)_{>i}$ e o resultado é válido trivialmente pela regra ω.
   - Seja $\dfrac{\underline{n}:\langle\Gamma\vdash u\rangle}{\underline{n+1}:\langle\omega.\Gamma\vdash u\rangle}$. Pelo Lema 5.2.7.1, $|\omega.\Gamma|=n+1$ logo $i<(n+1)$ e $\{\,\underline{i}\,/N\}\,\underline{n+1}=\underline{n}$. Note que $(\omega.\Gamma)_i=\Gamma_{(i-1)}$ logo por HI tem-se que $\{\,\underline{i-1}\,/N\}\,\underline{n}:\langle\Gamma_{<(i-1)}.\Gamma_{>(i-1)}\vdash u\rangle$. Como $(i-1)<n$, $\{\,\underline{i-1}\,/N\}\,\underline{n}=\underline{n-1}$. Portanto, pela regra varn, $\underline{n}:\langle\omega.\Gamma_{<(i-1)}.\Gamma_{>(i-1)}\vdash u\rangle$.
   - Seja $\dfrac{M:\langle u.\Gamma\vdash\tau\rangle}{\lambda.M:\langle\Gamma\vdash u\to\tau\rangle}$. Se $sup(N)=0$ então, pelo Lema 3.1.11.1, $N^+\equiv N$. Senão, pelo Lema 5.2.10, $N^+:\langle\omega.\Delta\vdash\Gamma_i\rangle$. Por HI tem-se que $\{\,\underline{i+1}\,/N^+\}M:\langle u.\Gamma_{<i}.\Gamma_{>i}\vdash\tau\rangle$ logo, por $\to_i$, $\lambda.\{\,\underline{i+1}\,/N^+\}M:\langle\Gamma_{<i}.\Gamma_{>i}\vdash u\to\tau\rangle$.
   - Seja $\dfrac{M_1:\langle\Gamma\vdash u\to\tau\rangle \qquad M_2:\langle\Gamma'\vdash u\rangle}{(M_1\ M_2):\langle\Gamma\sqcap\Gamma'\vdash\tau\rangle}$. Suponha, s.p.d.g., que $i<sup(M_1)$ e $i<sup(M_2)$ logo $(\Gamma\sqcap\Gamma')_i=\Gamma_i\sqcap\Gamma'_i$. Pelas regras $\sqcap_e$, $\leqslant_{\langle\rangle}$ e $\leqslant$ tem-se que $N:\langle\Delta\vdash\Gamma_i\rangle$ e $N:\langle\Delta\vdash\Gamma'_i\rangle$. Assim, por HI, $\{\,\underline{i}\,/N\}M_1:\langle\Gamma_{<i}.\Gamma_{>i}\vdash u\to\tau\rangle$ e $\{\,\underline{i}\,/N\}M_2:\langle\Gamma'_{<i}.\Gamma'_{>i}\vdash u\rangle$. Portanto, pela regra $\to_e$, $(\{\,\underline{i}\,/N\}M_1\ \{\,\underline{i}\,/N\}M_2):\langle(\Gamma_{<i}\sqcap\Gamma'_{<i}).(\Gamma_{>i}\sqcap\Gamma'_{>i})\vdash\tau\rangle$.
   - Seja $\dfrac{M:\langle\Gamma\vdash u_1\rangle \qquad M:\langle\Gamma\vdash u_2\rangle}{M:\langle\Gamma\vdash u_1\sqcap u_2\rangle}$. Por HI tem-se que $\{\,\underline{i}\,/N\}M:\langle\Gamma_{<i}.\Gamma_{>i}\vdash u_1\rangle$ e $\{\,\underline{i}\,/N\}M:\langle\Gamma_{<i}.\Gamma_{>i}\vdash u_2\rangle$. Portanto, pela regra $\sqcap_i$ tem-se que $\{\,\underline{i}\,/N\}M:\langle\Gamma_{<i}.\Gamma_{>i}\vdash u_1\sqcap u_2\rangle$.
   - Seja $\dfrac{M:\langle\Gamma\vdash u\rangle \qquad \langle\Gamma\vdash u\rangle\leqslant\langle\Gamma'\vdash u'\rangle}{M:\langle\Gamma'\vdash u'\rangle}$. Pelo Lema 5.2.6.5, $\Gamma'\leqslant\Gamma$ e $u\leqslant u'$ logo, pelo Lema 5.2.6.2, $\Gamma'_i\leqslant\Gamma_i$ e $\Gamma'_{<i}.\Gamma'_{>i}\leqslant\Gamma_{<i}.\Gamma_{>i}$. Assim, pelas regras $\leqslant_{\langle\rangle}$

e $\leqslant$ tem-se $N:\langle\Delta\vdash\Gamma_i\rangle$ e, por HI, $\{\,\underline{i}\,/N\}M:\langle\Gamma_{<i}.\Gamma_{>i}\vdash u\rangle$. Portanto, pelas regras $\leqslant_{\langle\rangle}$ e $\leqslant$, $\{\,\underline{i}\,/N\}M:\langle\Gamma'_{<i}.\Gamma'_{>i}\vdash u'\rangle$.

2. - Se $\underline{1}:\langle\tau.nil\vdash\tau\rangle$, nada há provar.
   - Seja $M:\langle\omega^M\vdash\omega\rangle$. Tem-se os seguintes casos:
     - Se $FI(M)=\{\,\underline{i}\,\}$, então $|\omega^M|=i$ logo $(\omega^M)_{<i}.(\omega^M)_{>i}=\omega^{M'}$, onde M' é qualquer termo tal que $sup(M')=i-1$. Assim, $\omega^{M'}\sqcap\Delta=\Delta$. Pelos Lemas 3.1.14.3 e 5.2.7.1, $sup(\{\,\underline{i}\,/N\}M)=sup(N)=|\Delta|$. Portanto, pelo Lema 5.2.7.2, $\{\,\underline{i}\,/N\}M:\langle\Delta\vdash\omega\rangle$.
     - Senão, pelo Lema 3.1.14.3 e Lema 5.2.7.1 tem-se que $sup(\{\,\underline{i}\,/N\}M)=max(sup(N),sup(M){-}1)=max(|\Delta|,|\omega^M|-1)=|\Delta\sqcap((\omega^M)_{<i}.(\omega^M)_{>i})|$. Portanto, pelo Lema 5.2.7.2, $\{\,\underline{i}\,/N\}M:\langle\Delta\sqcap((\omega^M)_{<i}.(\omega^M)_{>i})\vdash\omega\rangle$.
   - Seja $\dfrac{\underline{n}:\langle\Gamma\vdash u\rangle}{\underline{n{+}1}:\langle\omega.\Gamma\vdash u\rangle}$. Para $i=n+1$ tem-se que $\{\,\underline{n{+}1}\,/N\}\,\underline{n{+}1}=N$ e, pelo Lema 5.2.7.1, $|\Gamma|=n$. Pelo Lema 5.2.9, $\Gamma_n=v$, onde $v\leqslant u$. Assim, pela regra $\sqcap_e$ e Lema 5.2.6.2, $(\omega.\Gamma_{<n}.nil)\sqcap\Delta\leqslant\Delta$. Portanto, pelas regras $\leqslant_{\langle\rangle}$ e $\leqslant$, $N:\langle(\omega.\Gamma_{<n}.nil)\sqcap\Delta\vdash u\rangle$.
   - Seja $\dfrac{M:\langle u.\Gamma\vdash\tau\rangle}{\lambda.M:\langle\Gamma\vdash u\to\tau\rangle}$. Observe que $(u.\Gamma)_{(i+1)}=\Gamma_i$. Se $sup(N)=0$ então, pelo Lema 3.1.11.1, $N^+\equiv N$, senão, pelo Lema 5.2.10, tem-se que $N^+:\langle\omega.\Delta\vdash\Gamma_i\rangle$. Por HI, $\{\,\underline{i{+}1}\,/N^+\}M:\langle(u.\Gamma_{<i}.\Gamma_{>i})\sqcap\Delta'\vdash\tau\rangle$, onde Δ' é nil ou $\omega.\Delta$. Se $\Delta'\equiv\omega.\Delta$, então $(u.\Gamma_{<i}.\Gamma_{>i})\sqcap\Delta'=u.\big((\Gamma_{<i}.\Gamma_{>i})\sqcap\Delta\big)$. Assim, pela regra $\to_i$, $\lambda.\{\,\underline{i{+}1}\,/N^+\}M:\langle(\Gamma_{<i}.\Gamma_{>i})\sqcap\Delta\vdash u\to\tau\rangle$. O caso $\Delta'\equiv nil$ é trivial.
   - Seja $\dfrac{M_1:\langle\Gamma\vdash u\to\tau\rangle \qquad M_2:\langle\Gamma'\vdash u\rangle}{(M_1\ M_2):\langle\Gamma\sqcap\Gamma'\vdash\tau\rangle}$. Se $\underline{i}\in FI(M_1)$ e $\underline{i}\in FI(M_2)$, então $(\Gamma\sqcap\Gamma')_i=\Gamma_i\sqcap\Gamma'_i$ logo, pelas regras $\sqcap_e$, $\leqslant_{\langle\rangle}$ e $\leqslant$, $N:\langle\Delta\vdash\Gamma_i\rangle$ e $N:\langle\Delta\vdash\Gamma'_i\rangle$. Por HI, $\{\,\underline{i}\,/N\}M_1:\langle(\Gamma_{<i}.\Gamma_{>i})\sqcap\Delta\vdash u\to\tau\rangle$ e $\{\,\underline{i}\,/N\}M_2:\langle(\Gamma'_{<i}.\Gamma'_{>i})\sqcap\Delta\vdash u\rangle$. Note que $(\Gamma_{<i}.\Gamma_{>i})\sqcap\Delta\sqcap(\Gamma'_{<i}.\Gamma'_{>i})\sqcap\Delta=\big((\Gamma\sqcap\Gamma')_{<i}.(\Gamma\sqcap\Gamma')_{>i}\big)\sqcap\Delta$. Portanto, pela regra $\to_e$, $\{\,\underline{i}\,/N\}(M_1\ M_2):\langle\big((\Gamma\sqcap\Gamma')_{<i}.(\Gamma\sqcap\Gamma')_{>i}\big)\sqcap\Delta\vdash\tau\rangle$. Os casos quando $\underline{i}\notin FI(M_1)$ ou $\underline{i}\notin FI(M_2)$ são similares, usando o item 1 no passo indutivo sempre que necessário.
   - Seja $\dfrac{M:\langle\Gamma\vdash u_1\rangle \qquad M:\langle\Gamma\vdash u_2\rangle}{M:\langle\Gamma\vdash u_1\sqcap u_2\rangle}$. Por HI, $\{\,\underline{i}\,/N\}M:\langle(\Gamma_{<i}.\Gamma_{>i})\sqcap\Delta\vdash u_1\rangle$ e $\{\,\underline{i}\,/N\}M:\langle(\Gamma_{<i}.\Gamma_{>i})\sqcap\Delta\vdash u_2\rangle$. Portanto, pela regra $\sqcap_i$ tem-se que $\{\,\underline{i}\,/N\}M:\langle(\Gamma_{<i}.\Gamma_{>i})\sqcap\Delta\vdash u_1\sqcap u_2\rangle$.

- Seja $\dfrac{M:\langle \Gamma \vdash u\rangle \qquad \langle \Gamma \vdash u\rangle \leqslant \langle \Gamma' \vdash u'\rangle}{M:\langle \Gamma' \vdash u'\rangle}$. Pelo Lema 5.2.6.5, $\Gamma' \leqslant \Gamma$ e $u \leqslant u'$ logo, pelo Lema 5.2.6.2, $\Gamma'_i \leqslant \Gamma_i$ e $\Gamma'_{<i}.\Gamma'_{>i} \leqslant \Gamma_{<i}.\Gamma_{>i}$. Assim, pelas regras $\leqslant_{\langle\rangle}$ e $\leqslant$ tem-se que $N:\langle \Delta \vdash \Gamma_i\rangle$ e, por HI, $\{\underline{i}/N\}M:\langle (\Gamma_{<i}.\Gamma_{>i}) \sqcap \Delta \vdash u\rangle$. Pelo Lema 5.2.6.6, $(\Gamma'_{<i}.\Gamma'_{>i}) \sqcap \Delta \leqslant (\Gamma_{<i}.\Gamma_{>i}) \sqcap \Delta$ logo, pelas regras $\leqslant_{\langle\rangle}$ e $\leqslant$, $\{\underline{i}/N\}M:\langle (\Gamma'_{<i}.\Gamma'_{>i}) \sqcap \Delta \vdash u'\rangle$. □

Apesar da possibilidade de uma lei de redundância através da relação de subtipos e a regra de inferência $\leqslant$ correspondente, pelo Lema 5.2.7 observamos que essa redundância tem um limite superior. Portanto, como os índices livres que desaparecem depois de reduções β podem ser maiores que esse limitante, precisamos de uma definição similar ao $\Gamma\downharpoonleft_M$ apresentado para λ_{dB}^{SM}.

Definição 5.2.12 (Restrição para contextos): Sejam M um termo tal que $sup(M) = m$ e Γ um contexto. A restrição de Γ a $sup(M)$, dada por $\Gamma_{\leq m}.nil$, é denotada por $\Gamma\upharpoonright_M$.

A definição acima nos permite tipar o termo resultante de uma redução β com um contexto mais curto, relacionado ao contexto original. Primeiro, provamos algumas propriedades dessa nova restrição sobre contextos.

Lema 5.2.13:

1. Se $sup(N) \leq sup(M)$, então $\omega^M\upharpoonright_N = \omega^N$.
2. Se $|\Gamma| \leq sup(M)$, então $(\Gamma \sqcap \Delta)\upharpoonright_M = \Gamma \sqcap \Delta\upharpoonright_M$.
3. Se $sup(N) > 0$, então $(u.\Gamma)\upharpoonright_N = u.\Gamma\upharpoonright_{(\lambda.N)}$.

Demonstração.

1. Diretamente da Definição 5.2.12 e da definição de ω^M.
2. Seja $sup(M) = m$. Portanto, $(\Gamma \sqcap \Delta)\upharpoonright_M = (\Gamma \sqcap \Delta)_{\leq m}.nil = (\Gamma_{\leq m} \sqcap \Delta_{\leq m}).nil = (\Gamma_{\leq m}.nil) \sqcap (\Delta_{\leq m}.nil) = \Gamma \sqcap (\Delta_{\leq m}.nil) = \Gamma \sqcap \Delta\upharpoonright_M$.
3. Se $sup(N) > 0$ então, pelo Lema 3.1.10.2, $sup(\lambda.N) = sup(N) - 1$. Assim, $(u.\Gamma)\upharpoonright_N = (u.\Gamma)_{\leq sup(N)}.nil = u.\Gamma_{\leq (sup(N)-1)}.nil = u.\Gamma\upharpoonright_{(\lambda.N)}$. □

Por fim, temos a seguir os Teoremas 5.2.14 e 5.2.15 estabelecendo respectivamente a SR para a contração β e, ao contrário do obtido para λ_{dB}^{SM}, a SR para redução β em $\lambda_{dB}^{\sqcap}$.

Teorema 5.2.14 (SR para contração β**):** Se $(\lambda.M\ N) : \langle\Gamma \vdash u\rangle$ então $\{\underline{1}/N\}M : \langle\Gamma\upharpoonright_{\{\underline{1}/N\}M} \vdash u\rangle$.

Demonstração. Por indução na derivação de $(\lambda.M\ N){:}\langle\Gamma \vdash u\rangle$.

- Seja $(\lambda.M\ N){:}\langle\omega^{(\lambda.M\ N)} \vdash \omega\rangle$. Pelo Lema 3.1.15, $sup(\{\underline{1}/N\}M) \leq sup(\lambda.M\ N)$ logo, pelo Lema 5.2.13.1, $\omega^{(\lambda.M\ N)}\upharpoonright_{\{\underline{1}/N\}M} = \omega^{\{\underline{1}/N\}M}$. Pela regra ω obtém-se o resultado trivialmente.
- Seja $\dfrac{\lambda.M{:}\langle\Delta \vdash u\rightarrow\tau\rangle \qquad N{:}\langle\Delta' \vdash u\rangle}{(\lambda.M\ N){:}\langle\Delta\sqcap\Delta' \vdash \tau\rangle}$. Tem-se os seguintes casos:
  - Caso $sup(M) = 0$. Pelo Lema 5.2.9.3 tem-se que $\Delta = nil$ e $M : \langle nil \vdash \tau\rangle$. Pelo Lema 3.1.12.3 tem-se que $\{\underline{1}/N\}M \equiv M$. Portanto, $\Delta\sqcap\Delta' = \Delta'$ e $\Delta'\upharpoonright_{\{\underline{1}/N\}M} = \Delta'\upharpoonright_M = nil$.
  - Caso $sup(M)>0$. Pelo Lema 5.2.9.2, $M{:}\langle u.\Delta \vdash \tau\rangle$:
    - Se $\underline{1}\notin FI(M)$ então, pelo Lema 5.2.11.1, $\{\underline{1}/N\}M{:}\langle\Delta \vdash \tau\rangle$. Pelo Lema 5.2.13.2, $(\Delta\sqcap\Delta')\upharpoonright_{\{\underline{1}/N\}M} = \Delta\sqcap(\Delta'\upharpoonright_{\{\underline{1}/N\}M})$ logo, pela regra $\sqcap_e$ e pelo Lema 5.2.6.2, $(\Delta\sqcap\Delta')\upharpoonright_{\{\underline{1}/N\}M} \leqslant \Delta$. Portanto, pelas regras $\leqslant_{\langle\rangle}$ e $\leqslant$, $\{\underline{1}/N\}M{:}\langle(\Delta\sqcap\Delta')\upharpoonright_{\{\underline{1}/N\}M} \vdash \tau\rangle$.
    - Se $\underline{1}\in FI(M)$ então, pelo Lema 5.2.11.2, $\{\underline{1}/N\}M : \langle\Delta\sqcap\Delta' \vdash \tau\rangle$. Pelo Lema 5.2.7.1, $|\Delta\sqcap\Delta'| = sup(\{\underline{1}/N\}M)$ logo $(\Delta\sqcap\Delta')\upharpoonright_{\{\underline{1}/N\}M} = \Delta\sqcap\Delta'$.
- Seja $\dfrac{(\lambda.M\ N){:}\langle\Gamma \vdash u_1\rangle \qquad (\lambda.M\ N){:}\langle\Gamma \vdash u_2\rangle}{(\lambda.M\ N){:}\langle\Gamma \vdash u_1\sqcap u_2\rangle}$. Por HI, $\{\underline{1}/N\}M{:}\langle\Gamma\upharpoonright_{\{\underline{1}/N\}M} \vdash u_1\rangle$ e $\{\underline{1}/N\}M{:}\langle\Gamma\upharpoonright_{\{\underline{1}/N\}M} \vdash u_2\rangle$. Logo, pela regra $\sqcap_i$, $\{\underline{1}/N\}M{:}\langle\Gamma\upharpoonright_{\{\underline{1}/N\}M} \vdash u_1\sqcap u_2\rangle$.
- Seja $\dfrac{(\lambda.M\ N){:}\langle\Gamma \vdash u\rangle \qquad \langle\Gamma \vdash u\rangle \leqslant \langle\Gamma' \vdash u'\rangle}{(\lambda.M\ N){:}\langle\Gamma' \vdash u'\rangle}$. Por HI, $\{\underline{1}/N\}M{:}\langle\Gamma\upharpoonright_{\{\underline{1}/N\}M} \vdash u\rangle$. Pelo Lema 5.2.6.5, $\Gamma' \leqslant \Gamma$ e $u \leqslant u'$ logo, pelo Lema 5.2.6.2, $\Gamma'\upharpoonright_{\{\underline{1}/N\}M} \leqslant \Gamma\upharpoonright_{\{\underline{1}/N\}M}$. Portanto, pelas regras $\leqslant_{\langle\rangle}$ e $\leqslant$, $\{\underline{i}/N\}M{:}\langle\Gamma'\upharpoonright_{\{\underline{1}/N\}M} \vdash u'\rangle$. □

Teorema 5.2.15 (SR para redução β**):** Se $M{:}\langle\Gamma \vdash u\rangle$ e $M \rightarrow_\beta N$, então $N{:}\langle\Gamma\upharpoonright_N \vdash u\rangle$.

Demonstração. Por indução na derivação de $M{:}\langle\Gamma \vdash u\rangle$.

- Seja $M{:}\langle\omega^M \vdash \omega\rangle$. Tem-se que $FI(N)\subseteq FI(M)$ logo $sup(N)\leq sup(M)$. Pelo Lema 5.2.13.1 tem-se que $\omega^M\upharpoonright_N = \omega^N$. Portanto, pela regra ω, $N{:}\langle\omega^N \vdash \omega\rangle$.

- Seja $\dfrac{M' : \langle v.\Gamma \vdash \tau\rangle}{\lambda.M' : \langle \Gamma \vdash v \to \tau\rangle}$. Por HI, $N' : \langle (v.\Gamma)\lceil_{N'} \vdash \tau\rangle$, onde $M' \to_\beta N'$.

  Se $sup(N') = 0$, então $N' : \langle nil \vdash \tau\rangle$. Pela regra $\to'_i$ tem-se que $\lambda.N' : \langle nil \vdash \omega \to \tau\rangle$ logo, pelas regras $\to$, $\leqslant_{\langle\rangle}$ e $\leqslant$, $\lambda.N' : \langle nil \vdash v \to \tau\rangle$.

  Se $sup(N') > 0$ então, pelo Lema 5.2.13.3, $(v.\Gamma)\lceil_{N'} = v.\Gamma\lceil_{\lambda.N'}$. Portanto, pela regra $\to_i$, $\lambda.N' : \langle \Gamma\lceil_{\lambda.N'} \vdash v \to \tau\rangle$.

- Seja $\dfrac{M' : \langle nil \vdash \tau\rangle}{\lambda.M' : \langle nil \vdash \omega \to \tau\rangle}$. Para $M' \to_\beta N'$ tem-se, pelo Teorema 3.1.16, $sup(N') \leq sup(M') = 0$. Por HI, $N' : \langle nil \vdash \tau\rangle$ logo, pela regra $\to'_i$, $\lambda.N' : \langle nil \vdash \omega \to \tau\rangle$.

- Seja $\dfrac{M_1 : \langle \Delta \vdash u \to \tau\rangle \qquad M_2 : \langle \Delta' \vdash u\rangle}{(M_1\ M_2) : \langle \Delta \sqcap \Delta' \vdash \tau\rangle}$. Seja $N \equiv (N_1\ M_2)$ tal que $M_1 \to_\beta N_1$. Por HI tem-se $N_1 : \langle \Delta\lceil_{N_1} \vdash u \to \tau\rangle$ logo, pela regra $\to_e$, $(N_1\ M_2) : \langle \Delta\lceil_{N_1} \sqcap \Delta' \vdash \tau\rangle$.

  - Se $sup(N_1) \geq sup(M_2)$ então $sup(N) = sup(N_1)$. Portanto, pelo Lema 5.2.13.2, $(\Delta \sqcap \Delta')\lceil_{N_1} = \Delta\lceil_{N_1} \sqcap \Delta'$.

  - Se $sup(M_2) > sup(N_1)$ então $sup(N) = sup(M_2)$. Assim, pelo Lema 5.2.13.2, $(\Delta \sqcap \Delta')\lceil_{M_2} = \Delta\lceil_{M_2} \sqcap \Delta'$. Pela regra $\sqcap_e$ e pelo Lema 5.2.6.2 tem-se que $(\Delta\lceil_{M_2})_{>sup(N_1)} \sqcap \Delta'_{>sup(N_1)} \leqslant \Delta'_{>sup(N_1)}$. Logo, pelo Lema 5.2.6.2 tem-se que $(\Delta \sqcap \Delta')\lceil_{N_1}.\,((\Delta\lceil_{M_2})_{>sup(N_1)} \sqcap \Delta'_{>sup(N_1)}) \leqslant (\Delta \sqcap \Delta')\lceil_{N_1}.\,\Delta'_{>sup(N_1)}$. Observe que, pelo Lema 5.2.3.4 e pela Definição 5.2.12, $(\Delta \sqcap \Delta')\lceil_{N_1}.\,\Delta'_{>sup(N_1)} = \Delta\lceil_{N_1} \sqcap \Delta'$ e $(\Delta \sqcap \Delta')\lceil_{N_1}.\,((\Delta\lceil_{M_2})_{>sup(N_1)} \sqcap \Delta'_{>sup(N_1)}) = \Delta\lceil_{M_2} \sqcap \Delta'$. Portanto, pelas regras $\leqslant_{\langle\rangle}$ e $\leqslant$, $N : \langle \Delta\lceil_{M_2} \sqcap \Delta' \vdash \tau\rangle$.

- Seja $\dfrac{M : \langle \Gamma \vdash u_1\rangle \qquad M : \langle \Gamma \vdash u_2\rangle}{M : \langle \Gamma \vdash u_1 \sqcap u_2\rangle}$. Por HI tem-se $N : \langle \Gamma\lceil_N \vdash u_1\rangle$ e $N : \langle \Gamma\lceil_N \vdash u_2\rangle$. Portanto, pela regra $\sqcap_i$, $N : \langle \Gamma\lceil_N \vdash u_1 \sqcap u_2\rangle$.

- Seja $\dfrac{M : \langle \Gamma' \vdash u'\rangle \qquad \langle \Gamma' \vdash u'\rangle \leqslant \langle \Gamma \vdash u\rangle}{M : \langle \Gamma \vdash u\rangle}$. Por HI, $N : \langle \Gamma'\lceil_N \vdash u'\rangle$. Pelo Lema 5.2.6.5 tem-se que $\Gamma \leqslant \Gamma'$ e $u' \leqslant u$ e pelo Lema 5.2.6.2 tem-se que $\Gamma\lceil_N \leqslant \Gamma'\lceil_N$. Portanto, pelas regras $\leqslant_{\langle\rangle}$ e $\leqslant$, $N : \langle \Gamma\lceil_N \vdash u\rangle$. $\square$

Capítulo 6

Caracterização de PT para formas β-normais no $\lambda_{dB}^{SM_r}$

Neste capítulo apresentamos as tipagens principais no sistema $\lambda_{dB}^{SM_r}$ para formas β-normais do λ_{dB}-calculus. Esse trabalho, publicado em [101], é análogo ao trabalho de caracterização realizado em [37]. A restrição da forma dos tipos na regra ($\mathrm{var_r}$), em relação a regra (var) de λ_{dB}^{SM}, é essencial para que a operação sintática associada à definição de tipagem principal seja apenas a substituição de tipos. A substituição de tipos utilizada é ligeiramente diferente da substituição para tipos simples, apresentada na Definição 2.2.4, pois mapeia variáveis de tipo para elementos em $\mathcal{T}$. A seguir, apresentamos a definição para a extensão para elementos em $\mathcal{U}$ e conceitos utilizados na prova de completitude do algoritmo de inferência, apresentado na Seção 6.1.

Definição 6.0.16 (Substituição para IT restrita):

1. Seja $s:\mathcal{A} \to \mathcal{T}$ uma substituição. A extensão de s para $\mathcal{T}$ é dada por $s(u \to \tau) = s(u) \to s(\tau)$ e para elementos em $\mathcal{U}$ por $s(\omega) = \omega$ e $s(u \wedge v) = s(u) \wedge s(v)$.

   A extensão para contextos é dada por $s(nil) = nil$ e $s(u.\Gamma) = s(u).s(\Gamma)$.

2. O **domínio** de uma substituição s é definido por $Dom(s) = \{\alpha \mid s(\alpha) \neq \alpha\}$.

3. Uma substituição s tal que $Dom(s) = \{\alpha\}$ é denotada por $[\alpha/\sigma]$, onde $s(\alpha) = \sigma$.

4. Para duas substituições s_1 e s_2 com domínio disjuntos, seja a **composição disjunta** $s_1 + s_2$ definida por:

$$(s_1 + s_2)(\alpha) \begin{cases} s_i(\alpha) & \text{se } \alpha \in Dom(s_i), \text{para } i \in \{1, 2\} \\ \alpha & \text{se } \alpha \notin Dom(s_1) \cup Dom(s_2) \end{cases}$$

5. Seja $TV(u)$ o **conjunto de ocorrências de variáveis de tipo** de $u \in \mathcal{U}$. A extensão para contextos é direta.

Portanto, a definição de PT para o sistema $\lambda_{dB}^{SM_r}$ depende apenas da substituição de tipos como definida acima. Essa característica é similar a sistemas com tipos simples, como o sistema $\lambda_{dB}^{\rightarrow}$ apresentado no Capítulo 4.

Porém, como visto no Capítulo 5, essa restrição impede que o sistema tenha a propriedade de SR para a redução β. Apesar de SR ser uma propriedade básica, que deve ser satisfeita por qualquer sistema de atribuição de tipos, o sistema infere tipagens para todas as β-nfs. Além disso, o sistema apresentado é uma restrição de sistemas de IT mais complexos que têm sido amplamente investigados, onde as inferências de PT são precedidas de um processo de normalização do termo (*cf.* [95]). Assim, o sistema $\lambda_{dB}^{SM_r}$ compõe uma abordagem pertinente na caracterização de PT em sistemas de IT para $\lambda_{dB}^{SM_r}$.

6.1 Inferência de tipos e tipagem principal

A seguir, introduzimos a definição de tipagem principal para β-nfs em $\lambda_{dB}^{SM_r}$.

Definição 6.1.1 (Tipagem principal para β-nfs em $\lambda_{dB}^{SM_r}$): Seja N uma β-nf. Uma **tipagem principal** de N em $\lambda_{dB}^{SM_r}$ é uma tipagem $\Theta = \langle \Gamma \vdash \tau \rangle$ tal que:

1. $\lambda_{dB}^{SM_r} \Vdash N : \Theta$

2. Se $\lambda_{dB}^{SM_r} \Vdash N : \Theta'$ para alguma tipagem $\Theta' = \langle \Gamma' \vdash \tau' \rangle$, então existe uma substituição s tal que $s(\Theta) = \Theta'$.

A Definição 6.1.1 acima é similar a Definição 4.1.4, das tipagens principais em $\lambda_{dB}^{\rightarrow}$. Apesar disso, não está claro como provar a correspondência da Definição 6.1.1 e a Definição 2.2.2. O problema é saber se, para $M \in Termos_{\lambda_{dB}^{SM_r}}(\Theta)$, apenas a substituição é suficiente para obtermos todas as tipagens de M em $\lambda_{dB}^{SM_r}$.

A seguir, introduzimos o algortitmo `Infer` para inferência de tipos para β-nfs, similar ao algoritmo de [37].

Definição 6.1.2 (Algoritmo para Inferência de Tipos): Seja N uma β-nf:

$\texttt{Infer}(N) =$

Case $N = \underline{n}$
let α é uma v.t. nova
return $(\omega^{\underline{n-1}}.\alpha.nil, \alpha)$

Case $N = \lambda.N'$
let $(\Gamma', \sigma) = \texttt{Infer}(N')$
if $(\Gamma' = u.\Gamma)$ **then**
return $(\Gamma, u \to \sigma)$
else
return $(nil, \omega \to \sigma)$

Case $N = (\underline{n}\, N_1 \cdots N_m)$
let $(\Gamma^1, \sigma_1) = \texttt{Infer}(N_1)$
$\vdots$
$(\Gamma^m, \sigma_m) = \texttt{Infer}(N_m)$
α é uma v.t. nova
return $((\omega^{\underline{n-1}}.\sigma_1 \to \cdots \to \sigma_m \to \alpha.nil) \wedge \Gamma^1 \wedge \cdots \wedge \Gamma^m, \alpha)$

A noção de variável de tipo nova é utilizada na prova de completitude. Tomar variáveis novas é a forma de garantir que a cada vez que tomamos um elemento do conjunto $\mathcal{A}$, esse elemento é diferente dos tomados anteriormente. Portanto, duas chamadas ao algoritmo que não se sobreponham retornam pares com conjuntos disjuntos de variáveis de tipo.

Teorema 6.1.3 (Correção de Infer): Se N é uma β-nf e $\texttt{Infer}(N) = (\Gamma, \sigma)$, então $N : \langle \Gamma \vdash_{\lambda_{dB}^{SM_r}} \sigma \rangle$.

Demonstração. Por indução na estrutura de N.

- Se $N \equiv \underline{n}$ então $\texttt{Infer}(\underline{n}) = (\omega^{\underline{n-1}}.\alpha.nil, \alpha)$. Pela regra var_r, $\underline{1} : \langle \alpha.nil \vdash \alpha \rangle$ e, pela aplicação de varn $n-1$ vezes, $\underline{n} : \langle \omega^{\underline{n-1}}.\alpha.nil \vdash \alpha \rangle$.

- Seja $N \equiv \lambda.N'$. Se $(\Gamma', \sigma) = \texttt{Infer}(N')$ então, por HI, $N' : \langle \Gamma' \vdash \sigma \rangle$. Assim, se $\Gamma' = u.\Gamma$ então $\texttt{Infer}(\lambda.N') = (\Gamma, u \to \sigma)$ e, por $\to_i$, $\lambda.N' : \langle \Gamma \vdash u \to \sigma \rangle$, senão $\texttt{Infer}(\lambda.N') = (nil, \omega \to \sigma)$ e, por $\to_i'$, $\lambda.N' : \langle nil \vdash \omega \to \sigma \rangle$.

- Seja $N \equiv \underline{n}\, N_1 \cdots N_m$. Se $\forall 1 \leq i \leq m$, $(\Gamma^i, \sigma_i) = \texttt{Infer}(N_i)$ então, por HI, $\forall 1 \leq i \leq m$, $N_i : \langle \Gamma^i \vdash \sigma_i \rangle$. Seja $\Delta = \omega^{\underline{n-1}}.\sigma_1 \to \cdots \to \sigma_m \to \alpha.nil$. Assim, $\texttt{Infer}(N) = (\Delta \wedge \Gamma^1 \wedge \cdots \wedge \Gamma^m, \alpha)$ para alguma v.t. α nova. Por var_r e por varn $n-1$-vezes, $\underline{n} : \langle \Delta \vdash \sigma_1 \to \cdots \to \sigma_m \to \alpha \rangle$ e, por $\to_e$ m-vezes, $N : \langle \Delta \wedge \Gamma^1 \wedge \cdots \wedge \Gamma^m \vdash \alpha \rangle$. $\square$

Observe que, como a escolha das v.t. novas não está formalmente definida, o algoritmo `Infer` está bem definido módulo o nome das variáveis de tipo.

Corolário 6.1.4: Se N é uma β-nf então N é tipável em $\lambda_{dB}^{SM_r}$.

Teorema 6.1.5 (Completitude de `Infer`): Seja N uma β-nf. Se $N : \langle \Gamma \vdash_{\lambda_{dB}^{SM_r}} \sigma \rangle$, então para $(\Gamma', \sigma') = \mathtt{Infer}(N)$ existe uma substituição de tipos s tal que $s(\Gamma') = \Gamma$ e $s(\sigma') = \sigma$.

Demonstração. Por indução na estrutura de N.

- Seja $N \equiv \underline{n}$. Se $\underline{n} : \langle \Gamma \vdash \sigma \rangle$ então, pelo Lemas 5.1.7 e 5.1.8.1, $\Gamma = \underline{\omega^{n-1}}.\sigma.nil$. Tem-se que $\mathtt{Infer}(\,\underline{n}\,) = (\underline{\omega^{n-1}}.\alpha.nil, \alpha)$ logo seja $s = [\alpha/\sigma]$.

- Seja $N \equiv \lambda.N'$ e suponha que $\lambda.N' : \langle \Gamma \vdash \sigma \rangle$.

  Se $\Gamma = nil$, então pelo Lema 5.1.8.3 tem-se que ou $\sigma = \omega \to \tau$ e $N' : \langle nil \vdash \tau \rangle$ ou $\sigma = \wedge_{j=1}^{n} \sigma_j \to \tau$ e $N' : \langle \wedge_{j=1}^{n} \sigma_j.nil \vdash \tau \rangle$. No primeiro caso, por HI tem-se $\mathtt{Infer}(N') = (\Gamma', \tau')$ e existe s tal que $s(\tau') = \tau$ e $s(\Gamma') = nil$ logo $\Gamma' = nil$. Assim, $\mathtt{Infer}(\lambda.N') = (nil, \omega \to \tau')$ e $s(\omega \to \tau') = s(\omega) \to s(\tau') = \sigma$. No segundo caso, por HI tem-se $\mathtt{Infer}(N') = (\Gamma', \tau')$ e existe s tal que $s(\tau') = \tau$ e $s(\Gamma') = \wedge_{j=1}^{n} \sigma_j.nil$. Assim, $\Gamma' = u.nil$ onde $s(u) = \wedge_{j=1}^{n} \sigma_j$. Portanto, $\mathtt{Infer}(\lambda.N') = (nil, u \to \tau')$ e $s(u \to \tau') = s(u) \to s(\tau') = \sigma$.

  Se $|\Gamma| > 0$, então pelo Lema 5.1.8.4 tem-se que $\sigma = u \to \tau$ e $N' : \langle u.\Gamma \vdash \tau \rangle$. A demostração é então análoga a anterior.

- Seja $N \equiv (\,\underline{n}\, N_1 \cdots N_m)$. Se $(\,\underline{n}\, N_1 \cdots N_m) : \langle \Gamma \vdash \sigma \rangle$ então, pelo Lema 5.1.8.5, $\Gamma = (\underline{\omega^{n-1}}.\sigma_1 \to \cdots \to \sigma_m \to \sigma.nil) \wedge \Gamma^1 \wedge \cdots \wedge \Gamma^m$ e $\forall 1 \leq i \leq m$, $N_i : \langle \Gamma^i \vdash \sigma_i \rangle$. Por HI, $\forall 1 \leq i \leq m$, $\mathtt{Infer}(N_i) = (\Gamma^{i'}, \sigma_i')$ e existe uma s_i tal que $s_i(\sigma_i') = \sigma_i$ e $s_i(\Gamma^{i'}) = \Gamma^i$. Tem-se que $\mathtt{Infer}(N) = ((\underline{\omega^{n-1}}.\sigma_1' \to \cdots \to \sigma_m' \to \alpha.nil) \wedge \Gamma^{1'} \wedge \cdots \wedge \Gamma^{m'}, \alpha)$, para alguma v.t. α nova. O domínio de cada s_i é composto pelas v.t.s retornadas por cada chamada ao `Infer` para o N_i correspondente. Consequentemente, os domínios são dois a dois disjuntos. Assim, para $s = [\alpha/\sigma] + s_1 + \cdots + s_m$ tem-se o resultado desejado. $\square$

Logo, o par retornado por `Infer` para uma β-nf N é a tipagem mais geral de N no sistema $\lambda_{dB}^{SM_r}$. Note que essas tipagens são únicas módulo renomeamento para variáveis de tipo.

Teorema 6.1.6: Se N é uma β-nf e $(\Gamma, \sigma) = \texttt{Infer}(N)$, então $\Theta = \langle \Gamma \vdash \sigma \rangle$ é uma tipagem principal de N em $\lambda_{dB}^{SM_r}$.

Demonstração. Seja N uma β-nf e $(\Gamma, \sigma) = \texttt{Infer}(N)$. Pelo Teorema 6.1.3 tem-se que $N : \langle \Gamma \vdash_{\lambda_{dB}^{SM_r}} \sigma \rangle$. Seja $\Theta' = \langle \Gamma' \vdash \sigma' \rangle$ uma tipagem qualquer de N em $\lambda_{dB}^{SM_r}$. Pelo Teorema 6.1.5 tem-se que existe uma substituição s tal que $s(\Theta') = \Theta$. Logo, pela Definição 6.1.1 tem-se que Θ é uma tipagem principal de N em $\lambda_{dB}^{SM_r}$. $\square$

A seguir, apresentamos um lema estabelecendo a relação entre o multiconjunto de índices livres de uma β-nf e os tipos designados pelo contexto em uma PT deste termo.

Lema 6.1.7: Se $\texttt{Infer}(N) = (\Gamma, \sigma)$ para N uma β-nf, então $\Gamma_i = \wedge_{j=1}^{m} \sigma_j$ se, e somente se, $\underline{i} \in^m \overline{FI}(N)$.

Demonstração. Por indução na estrutura de N. Note que, pelo Lema 3.1.9.3, $\underline{n} \notin FI(M)$ sse $\underline{n} \notin \overline{FI}(M)$.

- Se $N \equiv \underline{n}$, nada há provar.
- Seja $N \equiv \lambda.N'$. Pelo Lema 3.1.9.2 tem-se que $\underline{i} \in^m \overline{FI}(\lambda.N')$ sse $\underline{i+1} \in^m \overline{FI}(N')$. Por HI, se $\texttt{Infer}(N') = (\Gamma', \sigma')$ então $\Gamma'_{i+1} = \wedge_{j=1}^{m} \sigma'_j$ sse $\underline{i+1} \in^m \overline{FI}(N')$. Se $\Gamma' = nil$, nada há provar. Senão, para $\Gamma' = u.\Gamma$, $\texttt{Infer}(\lambda.N') = (\Gamma, u \to \sigma')$. Assim, $\Gamma_i = \Gamma'_{i+1} = \wedge_{j=1}^{m} \sigma'_j$ sse $\underline{i+1} \in^m \overline{FI}(N')$ sse $\underline{i} \in^m \overline{FI}(\lambda.N')$.
- Seja $N \equiv \underline{n}\, N_1 \cdots N_m$. Se $\forall 1 \leq k \leq m$, $\texttt{Infer}(N_k) = (\Gamma^k, \sigma^k)$ então, por HI, $\Gamma^k_i = \wedge_{j=1}^{l_k} \sigma^k_j$ sse $\underline{i} \in^{l_k} \overline{FI}(N_k)$. Tem-se que $\texttt{Infer}(\underline{n}\, N_1 \cdots N_m) = (\Delta, \alpha)$, para $\Delta = (\omega^{\underline{n-1}}.\sigma_1 \to \cdots \to \sigma_m \to \alpha.nil) \wedge \Gamma^1 \wedge \cdots \wedge \Gamma^m$ onde α é uma v.t. nova. Note que, para todo $\underline{i} \in FI(N)$, $\underline{i} \in^l FI(N)$ onde $l = 1 + \sum_{k=1}^{m} l_k$ se $i = n$ e $l = \sum_{k=1}^{m} l_k$ caso contrário, e $\underline{i} \in^{l_k} \overline{FI}(N_k)$. Assim, $\Delta_i = \wedge_{j=1}^{l} \sigma_j$. $\square$

No Lema 6.1.7 acima, as hipóteses de N ser uma β-nf e da tipagem estar em $\lambda_{dB}^{SM_r}$ têm um papel crucial. Seguem contraexemplos quando uma destas não é satisfeita.

Exemplo 6.1.1: (a) Seja $M \equiv \big(\lambda.\big((\underline{2}\ \underline{1})\ (\underline{3}\ \underline{1})\big)\ \underline{4}\big)$. Tem-se que $M : \langle \Gamma \vdash_{\lambda_{dB}^{SM_r}} \alpha_4 \rangle$, para $\Gamma = \alpha_1 \to \alpha_2 \,.\, \alpha_3 \to \alpha_2 \to \alpha_4 \,.\, (\alpha_1 \wedge \alpha_3).nil$.

(b) Seja $N \equiv (\underline{1}\ \underline{3})$. Tem-se que $(\underline{1}\ \underline{3}) : \langle \omega^{\underline{2}}.(\alpha \wedge \beta).nil \vdash_{\lambda_{dB}^{SM}} \tau \rangle$, pois pela regra (var), $\underline{1} : \langle (\alpha \wedge \beta) \to \tau.nil \vdash (\alpha \wedge \beta) \to \tau \rangle$.

6.2 Caracterização das tipagens principais

Apresentamos nesta seção, uma caracterização das tipagens principais das β-nfs em $\lambda_{dB}^{SM_r}$, análoga à caracterização apresentada em [37]. Como ponto de partida, introduzimos subconjuntos próprios de $\mathcal{T}$ e $\mathcal{U}$ contendo os pares retornados por `Infer`.

Definição 6.2.1: 1. Sejam $\mathcal{T}_C$, $\mathcal{T}_{NF}$ e $\mathcal{U}_C$ definidos por:

$$\rho \in \mathcal{T}_C ::= \mathcal{A} \,|\, \mathcal{T}_{NF} \to \mathcal{T}_C \qquad\qquad \varphi \in \mathcal{T}_{NF} ::= \mathcal{A} \,|\, \mathcal{U}_C \to \mathcal{T}_{NF}$$

$$v \in \mathcal{U}_C ::= \omega \,|\, \mathcal{U}_C \wedge \mathcal{U}_C \,|\, \mathcal{T}_C$$

2. Seja $\mathcal{C}$ o conjunto de contextos $\Gamma ::= nil \,|\, v.\Gamma$ tais que $v \in \mathcal{U}_C$. Observe que $\mathcal{C}$ é fechado para $\wedge$.

Os conjuntos $\mathcal{T}_C$, $\mathcal{T}_{NF}$ e $\mathcal{C}$ são similares aos conjuntos $\mathcal{T}_E$, $\mathcal{T}_{NF}$ e $\mathcal{E}$, respectivamente, descritos na Seção 2.5.1.

Lema 6.2.2: Seja N uma β-nf. Se $\texttt{Infer}(N) = (\Gamma, \sigma)$, então $(\Gamma, \sigma) \in \mathcal{C} \times \mathcal{T}_{NF}$.

Demonstração. Por indução na estrutura de N.

- Seja $N \equiv \underline{n}$. Tem-se que $\texttt{Infer}(\underline{n}) = (\omega^{\underline{n-1}}.\alpha.nil, \alpha)$. Note que $\alpha \in \mathcal{T}_{NF}$ e $\omega, \alpha \in \mathcal{U}_C$ logo $\omega^{\underline{n-1}}.\alpha.nil \in \mathcal{C}$.

- Seja $N \equiv \lambda.N'$. Se $(\Gamma', \sigma) = \texttt{Infer}(N')$ então, por HI, $\sigma \in \mathcal{T}_{NF}$ e $\Gamma' \in \mathcal{C}$. Se $\Gamma' = v.\Gamma$ então $\texttt{Infer}(\lambda.N') = (\Gamma, v \to \sigma)$ logo $\Gamma \in \mathcal{C}$ e, como $v \in \mathcal{U}_C$, $v \to \sigma \in \mathcal{T}_{NF}$. Senão, $\texttt{Infer}(\lambda.N') = (nil, \omega \to \sigma)$ e a suposição é verdadeira.

- Seja $N \equiv \underline{n}\, N_1 \cdots N_m$. Se $\forall 1 \le i \le m$, $(\Gamma^i, \sigma_i) = \texttt{Infer}(N_i)$ então, por HI, $\Gamma^i \in \mathcal{C}$ e $\sigma_i \in \mathcal{T}_{NF}$. Seja $\Delta = \omega^{\underline{n-1}}.\sigma_1 \to \cdots \to \sigma_m \to \alpha.nil$, para alguma v.t. α nova, assim

$\texttt{Infer}(N) = (\Delta \wedge \Gamma^1 \wedge \cdots \wedge \Gamma^m, \alpha)$. Note que $\sigma_1 \to \cdots \to \sigma_m \to \alpha \in \mathcal{T}_C \subset \mathcal{U}_C$. Assim, como $\Delta, \Gamma^1, \ldots, \Gamma^m \in \mathcal{C}$, tem-se que $\Delta \wedge \Gamma^1 \wedge \cdots \wedge \Gamma^m \in \mathcal{C}$ e $\alpha \in \mathcal{T}_{NF}$. □

Definição 6.2.3: Seja $Im(\texttt{Infer})$ o conjunto formado pelos pares (Γ, σ) tais que $(\Gamma, \sigma) = \texttt{Infer}(N)$ para alguma β-nf N.

Corolário 6.2.4: $Im(\texttt{Infer}) \subseteq \mathcal{C} \times \mathcal{T}_{NF}$.

Usaremos definições para ocorrências finais e polaridade, *i.e.* positiva ou negativa, de ocorrência de variáveis de tipo similares as definições de J-L. Krivine em [68].

Definição 6.2.5 (Polaridade): Seja $\tau \in \mathcal{T}$. As **ocorrências positivas** e **negativas** de uma variável de tipo α são definidas por indução na estrutura de τ:

(i) Se $\tau \in \mathcal{A}$ então a possível ocorrência de α é positiva.

(ii) Se $\tau = u \to \sigma$ então as ocorrências positivas (resp. negativas) de α em τ são as ocorrências positivas (resp. negativas) de α em σ e as ocorrências negativas (resp. positivas) de α em u

(iii) Se $u = \omega$ então a possível ocorrência de α é positiva. Senão, para $u = \wedge_{i=1}^m \sigma_i$, as ocorrências positivas (resp. negativas) de α em u são as ocorrências positivas (resp. negativas) de α em σ_i, para qualquer $1 \leq i \leq m$.

A Definição 6.2.5 acima coincide com a definição dada em [39], quando restrita aos tipos em $\mathcal{T}_C$ e em $\mathcal{T}_{NF}$.

Definição 6.2.6 (Polaridade para contextos): Sejam $\alpha \in \mathcal{A}$ e Γ um contexto. As ocorrências positivas (resp. negativas) de α em Γ são as ocorrências positivas (resp. negativas) de α em Γ_i, $\forall 1 \leq i \leq |\Gamma|$.

Definição 6.2.7 (Ocorrência final): Sejam $u \in \mathcal{U}$ e $\alpha \in \mathcal{A}$, então α é uma **ocorrência final** de u se é um dos casos a seguir:

(i) Se $u \in \mathcal{A}$ ou $u = \omega$, então a possível ocorrência de α em u é final.

(ii) Se $u = u' \to \sigma$ e α é uma ocorrência final de σ.

(iii) Se $u = u_1 \wedge u_2$ e α é uma ocorrência final de u_1 ou u_2.

Definição 6.2.8 ($\mathcal{C}$-tipos): Sejam $\Gamma \in \mathcal{C}$ e $\varphi \in \mathcal{T}_{NF}$. Os **$\mathcal{C}$-tipos** T são definidos por:

$$T ::= \Gamma \Rightarrow \varphi \mid \Delta \Rightarrow, \qquad \text{onde } |\Delta| > 0$$

Observe que, para toda β-nf N, $\texttt{Infer}(N)$ tem associado um único $\mathcal{C}$-tipo T^N. Os A-*types* em [37] são definidos tomando o conjunto de multiconjuntos associado a um contexto de tipos e tranformandos-os em um único multiconjunto usado a esquerda de $\Rightarrow$. É importante lembrar que contexto de tipos para o λ-calculus com nomes é um conjunto de designação de tipos para variáveis de termos. Assim, para um contexto A e um tipo τ, $\overline{A} \Rightarrow \tau$ é o A-tipo correspondente, onde $\overline{A}$ é o multiconjunto obtido de A. Na Definição 6.2.8 acima a estrutura sequencial dos contextos é preservada.

Definição 6.2.9: Seja $T = \Gamma \Rightarrow \varphi$ um $\mathcal{C}$-tipo. Um $\mathcal{C}$-tipo T' está **contido** em T, denotado por $T' \sqsubseteq T$, se $T' = \Gamma' \Rightarrow$ ou $T' = \Gamma' \Rightarrow \varphi$ para $\Gamma' \sqsubseteq \Gamma$. Se $\Gamma' \sqsubset \Gamma$ então T' está **estritamente contido** em T, denotado por $T' \sqsubset T$.

Observe que na Definição 6.2.9 acima temos que Γ' pode ser *nil* no caso $T' = \Gamma' \Rightarrow \varphi$.

Definição 6.2.10 (Subtermos a esquerda): O **conjunto** $L(T)$ **dos subtermos à esquerda** de um $\mathcal{C}$-tipo T é definido por indução estrutural:

(i) $L(\Gamma \Rightarrow) = L(\Gamma)$.

(ii) $L(\Gamma \Rightarrow \varphi) = L(\Gamma) \cup L(\varphi)$.

(iii) $L(v.\Gamma) = \{v\} \cup L(\Gamma)$ se $v \neq \omega$ e $L(\Gamma)$ caso contrário.

(iv) $L(nil) = \emptyset$.

(v) $L(v \rightarrow \varphi) = \{v\} \cup L(\varphi)$ se $v \neq \omega$ e $L(\varphi)$ caso contrário.

(vi) $L(\alpha) = \emptyset$.

A extensão de alguns conceitos já definidos para tipo e contexto é direta e apresentada a seguir.

Definição 6.2.11: 1. $TV(\Gamma \Rightarrow \varphi) = TV(\Gamma) \cup TV(\varphi)$.

2. As ocorrências positivas (resp. negativas) de $\alpha \in \mathcal{A}$ em $T = \Gamma \Rightarrow \varphi$ são as ocorrências positivas (resp. negativas) de α em φ e as ocorrências negativas (resp. positivas) de α em Γ.

Definição 6.2.12 ($\mathcal{C}$-tipo fechado): Um $\mathcal{C}$-tipo T é **fechado** se cada variável de tipo $\alpha \in TV(T)$ tem exatamente uma ocorrência positiva e uma ocorrência negativa em T.

Lema 6.2.13: 1. $v.\Gamma \Rightarrow \varphi$ é fechado sse $\Gamma \Rightarrow v \rightarrow \varphi$ é fechado.

2. $nil \Rightarrow \varphi$ é fechado sse $nil \Rightarrow \omega \rightarrow \varphi$ é fechado.

3. Se $\forall 1 \leq i \leq m$, $T_i = \Gamma^i \Rightarrow \varphi_i$ é fechado e $TV(T_i)$ são dois a dois disjuntos, então $(\underline{\omega^{n-1}}.\varphi_1 \rightarrow \cdots \rightarrow \varphi_m \rightarrow \alpha.nil) \wedge \Gamma^1 \wedge \cdots \wedge \Gamma^m \Rightarrow \alpha$ é fechado, para qualquer v.t. α nova.

Demonstração. 1. Sejam $T = v.\Gamma \Rightarrow \varphi$ e $T' = \Gamma \Rightarrow v \rightarrow \varphi$. Note que $TV(T) = TV(T')$ e que a polaridade para as ocorrências das v.t.s em v são exatamente as mesmas em T e em T'.

2. Análoga à demonstração anterior.

3. Seja $T = (\underline{\omega^{n-1}}.\varphi_1 \rightarrow \cdots \rightarrow \varphi_m \rightarrow \alpha.nil) \wedge \Gamma^1 \wedge \cdots \wedge \Gamma^m \Rightarrow \alpha$ o $\mathcal{C}$-tipo como suposto. Como $TV(T_i)$ são dois a dois disjuntos, $TV(T) = \cup_{i=1}^{m} TV(T_i) \cup \{\alpha\}$ e T tem exatamente duas ocorrências de cada v.t. Note que $\forall 1 \leq i \leq m$, as ocorrências de v.t.s em Γ^i e em φ_i têm exatamente a mesma polaridade em T_i e em T e que α tem uma ocorrência positiva e uma negativa em T. Portanto, T é fechado. $\square$

Definição 6.2.14 ($\mathcal{C}$-tipo fechado ao final): Um $\mathcal{C}$-tipo $T = \Gamma \Rightarrow \varphi$ é **fechado ao final**, ou f.f., se a ocorrência final de φ é também uma ocorrência final de um elemento em $L(T)$.

Lema 6.2.15: 1. $v.\Gamma \Rightarrow \varphi$ é fechado ao final sse $\Gamma \Rightarrow v \rightarrow \varphi$ é fechado ao final.

2. $nil \Rightarrow \varphi$ é fechado ao final sse $nil \Rightarrow \omega \rightarrow \varphi$ é fechado ao final.

Demonstração. 1. Sejam $T = v.\Gamma \Rightarrow \varphi$ e $T' = \Gamma \Rightarrow v \rightarrow \varphi$. Note que a ocorrência final de $v \rightarrow \varphi$ é a mesma de φ. Se $v \neq \omega$, então pela Definição 6.2.10 tem-se que $L(T) = L(v.\Gamma) \cup L(\varphi) = \{v\} \cup L(\Gamma) \cup L(\varphi) = L(\Gamma) \cup L(v \to \varphi) = L(T')$. Senão, $L(T) = L(\omega.\Gamma) \cup L(\varphi) = L(\Gamma) \cup L(\varphi) = L(\Gamma) \cup L(\omega \to \varphi) = L(T')$. Portanto, T é f.f. sse T' é f.f.

2. Análoga a demonstração anterior. □

Definição 6.2.16 ($\mathcal{C}$-tipo fechado minimal): Um $\mathcal{C}$-tipo T é **fechado minimal**, ou f.m., se T é fechado e não existe T' fechado que esteja estritamente contido em T.

Lema 6.2.17:

1. Seja $v \neq \omega$. Se $v.\Gamma \Rightarrow \varphi$ é fechado minimal, então $\Gamma \Rightarrow v \to \varphi$ é fechado minimal.
2. $\omega.\Gamma \Rightarrow \varphi$ é fechado minimal sse $\Gamma \Rightarrow \omega \to \varphi$ é fechado minimal.
3. $nil \Rightarrow \varphi$ é fechado minimal sse $nil \Rightarrow \omega \to \varphi$ é fechado minimal.
4. Se $\forall 1 \leq i \leq m$, $T_i = \Gamma^i \Rightarrow \varphi_i$ é f.m. e $TV(T_i)$ são dois a dois disjuntos, então $T = (\omega^{\underline{n-1}}.\varphi_1 \to \cdots \to \varphi_m \to \alpha.nil) \wedge \Gamma^1 \wedge \cdots \wedge \Gamma^m \Rightarrow \alpha$ é fechado minimal para qualquer v.t. α nova.

Demonstração.

1. Sejam $T = v.\Gamma \Rightarrow \varphi$ f.m. e $T' = \Gamma \Rightarrow v \to \varphi$ onde $v \neq \omega$. Pelo Lema 6.2.131 tem-se que T' é fechado. Suponha que $T'' \sqsubset T'$. Se $T'' = \Gamma' \Rightarrow v \to \varphi$ então $T''' = v.\Gamma' \Rightarrow \varphi \sqsubset T$. Pelo Lema 6.2.13.1, T'' é fechado sse T''' é fechado. Assim, por T ser f.m., T'' não pode ser fechado. Se $T'' = \Gamma' \Rightarrow$ tem-se similarmente que T'' não pode ser fechado. Portanto, T' é f.m.
2. Pelo Lema 6.2.13.1 tem-se que $\omega.\Gamma \Rightarrow \varphi$ é fechado sse $\Gamma \Rightarrow \omega \to \varphi$ é fechado. Seja T tal que $T \sqsubset \omega.\Gamma \Rightarrow \varphi$. Tem-se que $T = \omega.\Gamma' \Rightarrow \varphi$ sse $\Gamma' \sqsubset \Gamma$ sse $T' = \Gamma' \Rightarrow \omega \to \varphi \sqsubset \Gamma \Rightarrow \omega \to \varphi$. Existe um T' correspondente quando $T = nil \Rightarrow \varphi$ e quando $T = \omega.\Gamma' \Rightarrow$. Portanto, pelo Lema 6.2.13.1, existe um T fechado e estritamente contido em $\omega.\Gamma \Rightarrow \varphi$ sse existe um T' fechado e estritamente contido em $\Gamma \Rightarrow \omega \to \varphi$.
3. Análoga a prova anterior.
4. Seja T definido como o suposto no enunciado e seja T' tal que $T' \sqsubseteq T$. Pelo Lema 6.2.13.3 tem-se que T é fechado e suponha que T' seja fechado. Se $T' = \Gamma' \Rightarrow$ então, como $|\Gamma'| > 0$, existe um $\Delta^i \neq nil$ para algum i tal que $\Delta^i \sqsubseteq \Gamma^i$ e $\Delta^i \sqsubseteq \Gamma'$. Note que

$TV(\Gamma^i)$ são dois a dois disjuntos, logo se $\Delta^i \neq \Gamma^i$ então $\Delta^i \Rightarrow$ é fechado e $\Delta^i \Rightarrow\ \sqsubset T^i$. Portanto, $\Delta^i = \Gamma^i$ e, analogamente, tem-se que $(\omega^{\underline{n-1}}.\varphi_1 \to \cdots \to \varphi_m \to \alpha.nil) \sqsubseteq \Gamma'$, obtendo um $\mathcal{C}$-tipo T' que não é fechado. Se $T' = \Gamma' \Rightarrow \alpha$ então com um argumento similar tem-se que $\Gamma' = (\omega^{\underline{n-1}}.\varphi_1 \to \cdots \to \varphi_m \to \alpha.nil) \wedge \Gamma^1 \wedge \cdots \wedge \Gamma^m$. Assim, T' é fechado sse T é fechado e $T' = T$. Portanto, T é f.m.. $\square$

Definição 6.2.18 ($\mathcal{C}$-tipo completo): Um $\mathcal{C}$-tipo T é **completo** se T é fechado ao final e fechado minimal.

Lema 6.2.19:

1. Seja $v \neq \omega$. Se $v.\Gamma \Rightarrow \varphi$ é completo, então $\Gamma \Rightarrow v \to \varphi$ é completo.
2. $\omega.\Gamma \Rightarrow \varphi$ é completo sse $\Gamma \Rightarrow \omega \to \varphi$ é completo.
3. $nil \Rightarrow \varphi$ é completo sse $nil \Rightarrow \omega \to \varphi$ é completo.
4. Se $\forall 1 \leq i \leq m$, $T_i = \Gamma^i \Rightarrow \varphi_i$ é completo e $TV(T_i)$ são dois a dois disjuntos, então $T = (\omega^{\underline{n-1}}.\varphi_1 \to \cdots \to \varphi_m \to \alpha.nil) \wedge \Gamma^1 \wedge \cdots \wedge \Gamma^m \Rightarrow \alpha$ é completo para qualquer v.t. α nova.

Demonstração.

1. Pelos Lemas 6.2.15.1 e 6.2.17.1.
2. Pelos Lemas 6.2.15.1 e 6.2.17.2.
3. Pelos Lemas 6.2.15.2 e 6.2.17.3.
4. Pelo Lema 6.2.17.4 tem-se que o T como suposto no enuciado é f.m.. Note que $(\varphi_1 \to \cdots \to \varphi_m \to \alpha) \wedge (\Gamma^1 \wedge \cdots \wedge \Gamma^m)_n \in L(T)$ logo T é f.f. $\square$

Lema 6.2.20: Se N é uma β-nf então T^N é completo.

Demonstração. Por indução na estrutura de N.

- Seja $N \equiv \underline{n}$. Tem-se que $\texttt{Infer}(N) = (\omega^{\underline{n-1}}.\alpha.nil, \alpha)$, assim $T^N = \omega^{\underline{n-1}}.\alpha.nil \Rightarrow \alpha$. Note que $L(T^N) = \{\alpha\}$. Logo, T^N é fechado e fechado ao final. Os dois únicos $\mathcal{C}$-tipos estritamente contidos em T^N são $\omega^{\underline{n-1}}.\alpha.nil \Rightarrow$ e $nil \Rightarrow \alpha$, que não são fechados. Portanto, T^N é fechado minimal.

- Seja $N \equiv \lambda.N'$. Se $(\Gamma', \varphi) = \texttt{Infer}(N')$ então, por HI, $T^{N'} = \Gamma' \Rightarrow \varphi$ é completo.

  Se $\Gamma' = v.\Gamma$ então $\texttt{Infer}(\lambda.N') = (\Gamma, v \to \varphi)$ e $T^N = \Gamma \Rightarrow v \to \varphi$. Se $v \neq \omega$, então pelo Lema 6.2.19.1 T^N é completo. Senão, pelo Lema 6.2.19.2, T^N é completo.

  Se $\Gamma' = nil$ então $\texttt{Infer}(\lambda.N') = (nil, \omega \to \varphi)$ e, pelo Lema 6.2.19.3, T^N é completo.

- Seja $N \equiv \underline{n}\, N_1 \cdots N_m$. Se $\forall 1 \leq i \leq m$, $(\Gamma^i, \varphi_i) = \texttt{Infer}(N_i)$ então, por HI, T^{N_i} é completo. Observe que $TV(T^{N_i})$ são dois a dois disjuntos porque são retornados por chamadas de `Infer` que não se sobrepõem. Tem-se que $\texttt{Infer}(N) = ((\omega^{\underline{n-1}}.\varphi_1 \to \cdots \to \varphi_m \to \alpha.nil) \wedge \Gamma^1 \wedge \cdots \wedge \Gamma^m, \alpha)$, para alguma v.t. α nova. Portanto, pelo Lema 6.2.19.4, T^N é completo. $\square$

Observe que nos itens 1 e 4 do Lema 6.2.19 acima nós apenas temos as provas de *suficiência*. Apresentamos a seguir, contraexemplos para cada condição de *necessidade*.

Exemplo 6.2.1: Seja $T = \Gamma \Rightarrow \varphi$ um $\mathcal{C}$-tipo completo. Então, para qualquer $\alpha \in \mathcal{A}$ nova, seja $T' = \Gamma \Rightarrow (\alpha \to \alpha) \to \varphi$. Assim, T' é completo mas $\alpha \to \alpha.\Gamma \Rightarrow \varphi$ não é f.m..

Exemplo 6.2.2: Seja $T = \beta_1 \to (\beta_2 \to \beta_3) \to \beta_4.(\beta_1 \to \beta_4) \to (\beta_3 \to \beta_2) \to \alpha.nil \Rightarrow \alpha$. Note que T é completo mas não existe uma partição de $\mathcal{C}$-tipos completos, tal que T possa ser obtido como descrito no Lema 6.2.19.4.

Portanto, para obtermos $\mathcal{C}$-tipos completos tais que essas condições de *necessidade* sejam satisfeitas, introduzimos a noção de $\mathcal{C}$-tipos principais, como feito em [37].

Definição 6.2.21 ($\mathcal{C}$-tipo principal): Seja T um $\mathcal{C}$-tipo completo. T é chamado de **principal** se:

(i) $T = \omega^{\underline{n-1}}.\alpha.nil \Rightarrow \alpha$.

(ii) $T = nil \Rightarrow \omega \to \varphi$ e $nil \Rightarrow \varphi$ é principal.

(iii) $T = \Gamma \Rightarrow v \to \varphi$, onde $\Gamma \neq nil$ ou $v \neq \omega$, e $v.\Gamma \Rightarrow \varphi$ é principal.

(iv) $T = \Gamma \Rightarrow \alpha$ e $\Gamma = (\omega^{\underline{n-1}}.\varphi_1 \to \cdots \to \varphi_m \to \alpha.nil) \wedge \Gamma^1 \wedge \cdots \wedge \Gamma^m$ tal que $\forall 1 \leq i \leq m$, $\Gamma^i \Rightarrow \varphi_i$ é principal.

Observe que na Definição 6.2.21 acima requisitamos explicitamente a existência da partição correspondente no caso $T = \Gamma \Rightarrow \alpha$ para $\Gamma \not\equiv \omega^{\underline{n-1}}.\alpha.nil$ e que $v.\Gamma \Rightarrow \varphi$ seja principal, logo completo, para $T = \Gamma \Rightarrow v \rightarrow \varphi$ onde $\Gamma \neq nil$ ou $v \neq \omega$. Apesar de termos pelo Lema 6.2.19.2 que $T = nil \Rightarrow \omega \rightarrow \varphi$ é completo se, e somente se, $T' = nil \Rightarrow \varphi$ é completo, esse caso deve ser definido de maneira similar aos anteriores. Caso contrário, se na Definição 6.2.21 temos apenas: "$T = nil \Rightarrow \omega \rightarrow \varphi$" então garantimos apenas que T' seja completo, permitindo que um contraexemplo como o do Exemplo 6.2.1 seja apresentado.

Lema 6.2.22: Se N é uma β-nf então T^N is principal.

Demonstração. Por indução na estrutura de N. Pelo Lema 6.2.20, T^N é completo:

- Se $N \equiv \underline{n}$ então $T^N = \omega \underline{^{n-1}}.\alpha.nil \Rightarrow \alpha$.
- Sejam $N \equiv \lambda.N'$ e $T^{N'} = \Gamma' \Rightarrow \varphi$. Por HI tem-se que $T^{N'}$ é principal.

  Se $\Gamma' = v.\Gamma$ então $T^{\lambda.N'} = \Gamma \Rightarrow v \rightarrow \varphi$. Observe que se $\Gamma = nil$ então, pelo Lema 5.1.7, $v \neq \omega$. Portanto, $T^{\lambda.N'}$ é principal.

  Senão, $\Gamma' = nil$ e $T^{\lambda.N'} = nil \Rightarrow \omega \rightarrow \varphi$. Portanto, $T^{\lambda.N'}$ é principal.
- Sejam $N \equiv \underline{n}\, N_1 \cdots N_m$ e $\forall 1 \leq i \leq m$, $T^{N_i} = \Gamma^i \Rightarrow \varphi_i$ que por HI é principal. Tem-se que $T^N = (\omega \underline{^{n-1}}.\varphi_1 \rightarrow \cdots \rightarrow \varphi_m \rightarrow \alpha.nil) \wedge \Gamma^1 \wedge \cdots \wedge \Gamma^m \Rightarrow \alpha$ para alguma v.t. α nova. Portanto, T^N é principal. $\square$

Portanto, a caracterização sintática de $\mathcal{C}$-tipos principais contém as tipagens principais das β-nfs, retornadas por `Infer`.

Definição 6.2.23 (Pares principais): Seja $\mathcal{P} = \{(\Gamma, \varphi) \in \mathcal{C} \times \mathcal{T}_{NF} \mid \Gamma \Rightarrow \varphi \text{ é principal}\}$.

Em outras palavras, pelo Lema 6.2.22 e análogo ao apresentado em [37]:

$$Im(\texttt{Infer}) \subseteq \mathcal{P}$$

Definição 6.2.24: Seja $FO(\alpha, \Gamma)$ o conjunto definido por:

$$FO(\alpha, \Gamma) = \{(i, \Gamma_i) \mid \alpha \text{ é uma ocorrência final de } \Gamma_i, \forall 1 \leq i \leq |\Gamma|\}$$

O conjunto $FO(\alpha, \Gamma)$ para um $T = \Gamma \Rightarrow \alpha$ principal, especificamente fechado e fechado ao final, tem propriedades que são usadas na definição do algoritmo de reconstrução.

Lema 6.2.25: Seja $T = \Gamma \Rightarrow \alpha$ um $\mathcal{C}$-tipo. Se T é fechado ao final então $FO(\alpha, \Gamma) \neq \emptyset$. Se além disso T é fechado, então $FO(\alpha, \Gamma)$ tem exatamente um elemento (i, v), tal que $v = (\varphi_1 \rightarrow \cdots \rightarrow \varphi_m \rightarrow \alpha) \wedge v'$ para $m \geq 0$ e $v' \in \mathcal{U}_C$ onde $\alpha \notin TV(v')$.

Demonstração. Seja $T = \Gamma \Rightarrow \alpha$. Pela Definição 6.2.10, $L(T) = \{\Gamma_i \neq \omega, \forall 1 \leq i \leq |\Gamma|\}$ logo se T é fechado ao final então pelo menos um elemento de Γ tem α como sua ocorrência final. Seja $(i, v) \in FO(\alpha, \Gamma)$. Se T é também fechado, então Γ tem exatamente uma ocorrência positiva de α, logo α ocorre unicamente em $v = \Gamma_i$. Note que $v \in \mathcal{U}_C$. Se $v \in \mathcal{T}_C$ então por indução estrutural tem-se que $v = \varphi_1 \rightarrow \cdots \rightarrow \varphi_m \rightarrow \alpha$ para $m \geq 0$ ($v = \alpha$ se $m = 0$). Senão, $v = v_1 \wedge v_2$ e α ocorre positivamente em v_1 ou em v_2. Assim, por indução na estrutura dos elementos em $\mathcal{U}_C$, comutatividade e associatividade de $\wedge$, obtém-se o resultado desejado. $\square$

Introduzimos o algoritmo `Recon`, para reconstruir uma β-nf N a partir de $(\Gamma, \varphi) \in \mathcal{P}$ tal que `Infer`$(N) = (\Gamma, \varphi)$, similar ao algoritmo introduzido em [37].

Definição 6.2.26 (Algoritmo de reconstrução):

`Recon`$(\Gamma, \tau) =$

Case (nil, α)
fail

Case (Γ, α)
let $\{(i^1, u_1), \ldots, (i^m, u_m)\} = FO(\alpha, \Gamma)$
if $m = 1$ & $u_1 = (\tau_1 \rightarrow \cdots \rightarrow \tau_n \rightarrow \alpha) \wedge u'$ t.q. $\alpha \notin TV(u')$
  then if $\forall 1 \leq i \leq n, \exists\, \Gamma^i \sqsubseteq \Gamma$ t.q. $\Gamma^i \Rightarrow \tau_i$ é principal
    then let $(N_1, \Delta^1) =$ `Recon`(Γ^1, τ_1)
      $\vdots$
      $(N_n, \Delta^n) =$ `Recon`(Γ^n, τ_n)
      $\Delta' = \omega^{\underline{i^1 - 1}}.\tau_1 \rightarrow \cdots \rightarrow \tau_n \rightarrow \alpha.nil$
      $\Gamma' = \Delta' \wedge \Gamma^1 \wedge \cdots \wedge \Gamma^n$
      $\Delta = \Gamma / \Gamma'$
    return $(\underline{i}^1 N_1 \cdots N_n, \Delta \wedge \Delta^1 \wedge \cdots \wedge \Delta^n)$
    else fail
  else fail

Case $(\Gamma, u \rightarrow \tau)$
if $\Gamma = nil$ & $u = \omega$
  then let $(N, \Delta) =$ `Recon`(nil, τ)
  else let $(N, \Delta) =$ `Recon`$(u.\Gamma, \tau)$
if $\Delta = nil$
  then return $(\lambda.N, \Delta)$
  else fail

Lema 6.2.27: Seja $(\Gamma, \varphi) \in \mathcal{P}$. Então $\texttt{Recon}(\Gamma, \varphi) = (N, nil)$, onde N é uma β-nf tal que $\texttt{Infer}(N) = (\Gamma, \varphi)$.

Demonstração. Por recorrência no número de chamadas ao $\texttt{Recon}$.

- Caso (Γ, α). Seja $T = \Gamma \Rightarrow \alpha$.

  Por hipótese tem-se que $(\Gamma, \alpha) \in \mathcal{P}$, assim T é principal logo fechado e fechado ao final. Pelo Lema 6.2.25, $FO(\alpha, \Gamma) = \{(i, (\varphi_1 \to \cdots \to \varphi_m \to \alpha) \wedge v')\}$ onde $\alpha \notin TV(v')$. Como Γ_i possui a única ocorrência de α em Γ, $\alpha \notin TV(\Delta'')$ para $\Delta'' = \Gamma / (\omega \underline{i-1} . \varphi_1 \to \cdots \to \varphi_m \to \alpha.nil)$.

  Se $m = 0$, então em $\texttt{Recon}$ tem-se que $\Gamma' = \Delta' = \omega \underline{i-1} . \alpha.nil$ logo $T = \Gamma' \wedge \Delta'' \Rightarrow \alpha$. T é fechado minimal, assim $\Delta'' = nil$ e $\Gamma = \Gamma'$. Portanto, $\texttt{Recon}(\Gamma, \alpha) = (\underline{i}, nil)$ e $\texttt{Infer}(\underline{i}) = (\omega \underline{i-1} . \alpha.nil, \alpha)$.

  Senão, por T ser principal, $\Gamma = (\omega \underline{n-1} . \varphi_1 \to \cdots \to \varphi_m \to \alpha.nil) \wedge \Gamma^1 \wedge \cdots \wedge \Gamma^m$ tal que $\forall 1 \leq j \leq m$, $\Gamma^j \Rightarrow \varphi_j$ é principal. Assim, $n = i$ e por HI tem-se que $\forall 1 \leq j \leq m$, $\texttt{Recon}(\Gamma^j, \varphi_j) = (N_j, nil)$ onde N_j é uma β-nf tal que $\texttt{Infer}(N_j) = (\Gamma^j, \varphi_j)$. Assim, tem-se em $\texttt{Recon}$ que $\Gamma = \Gamma'$ logo $\Delta = nil$. Então, $\texttt{Recon}(\Gamma, \alpha) = (\underline{i}\, N_1 \cdots N_m, nil)$ e $\texttt{Infer}(\underline{i}\, N_1 \cdots N_m) = ((\omega \underline{i-1} . \varphi_1 \to \cdots \to \varphi_m \to \alpha.nil) \wedge \Gamma^1 \wedge \cdots \wedge \Gamma^m, \alpha)$.

- Caso $(\Gamma, v \to \varphi)$. Seja $T = \Gamma \Rightarrow v \to \varphi$.

  Por hipótese tem-se que $(\Gamma, v \to \varphi) \in \mathcal{P}$ logo T é principal.

  Se $\Gamma = nil$ e $v = \omega$ então $T' = nil \Rightarrow \varphi$ é principal logo por HI tem-se que $\texttt{Recon}(nil, \varphi) = (N, nil)$ onde N é uma β-nf tal que $\texttt{Infer}(N) = (nil, \varphi)$. Portanto, $\texttt{Recon}(nil, \omega \to \varphi) = (\lambda.N, nil)$ e $\texttt{Infer}(\lambda.N) = (nil, \omega \to \varphi)$.

  Senão, $T' = v.\Gamma \Rightarrow \varphi$ é principal. Por HI tem-se que $\texttt{Recon}(v.\Gamma, \varphi) = (N, nil)$ onde N é uma β-nf tal que $\texttt{Infer}(N) = (v.\Gamma, \varphi)$. Portanto, $\texttt{Recon}(\Gamma, v \to \varphi) = (\lambda.N, nil)$ e $\texttt{Infer}(\lambda.N) = (\Gamma, v \to \varphi)$. □

Observe que, pelo Lema 6.2.27, temos que:

$$\mathcal{P} \subseteq Im(\texttt{Infer})$$

Assim, $\mathcal{P}$ é o conjunto de todas, e apenas, as tipagens principais das β-nfs em $\lambda_{dB}^{SM_r}$. Portanto,

$$\mathcal{P} = Im(\texttt{Infer})$$

Capítulo 7

Tipos com interseção para o λs_e-calculus

No presente capítulo, introduzimos um sistema de tipos com interseção para o cálculo de substituição explícita λs_e, baseado no sistema λ_{dB}^{SM}. Esse sistema é estendido tal que o sistema obtido possa inferir tipagens para os novos termos presentes no cálculo.

Na Seção 7.1 apresentamos um sistema de tipos com interseção para o λs. Pelo fato desse sistema ser mais próximo ao originalmente proposto na definição formal da substituição β para o λ_{dB}, as propriedades para o sistema de tipos proposto correspondem às propriedades de λ_{dB}^{SM}. As regras de inferência introduzidas para a tipagem de termos com operadores de atualização e substituição têm por referência os Lemas 5.1.9 e 5.1.10, sobre os operadores implícitos do λ_{dB} de atualização e substituição, respectivamente. Assim, o sistema λs^{SM} representa um passo em direção à adição de tipos com interseção para o λs_e, introduzido na Seção 7.2. Após algumas considerações sobre a adição das regras para composição de operadores ao λs, e as consequências nas tipagens de λs^{SM}, apresentamos o sistema $\lambda {s_e}^{\wedge}$.

As considerações e propostas de mudanças nas regras de inferência de tipagem em cada um dos sistemas propostos, têm por finalidade a obtenção de sistemas de tipos com interseção que sejam o mais restrito possível tal que satisfaçam a propriedade de SR. B. Guillaume apresenta em [47] um contraexemplo mostrando que o λs_e não é PSN, onde um termo que é tipável em $\lambda_{dB}^{\rightarrow}$, logo tipável em $\lambda^{\rightarrow}$, tem uma estratégia de redução infinita em λs_e. Assim, qualquer sistema de tipos que seja uma extensão do sistema de tipos simples $\lambda {s_e}^{\rightarrow}$ têm a mesma característica. O intuito de introduzir esses sistemas restritos é o de manter a relação desses sistemas com a propriedade de relevância do λ_{dB}^{SM}, provada no Lema 5.1.7. Assim, averiguamos se em tais sistemas existe a possibilidade de uma caracterização de termos SN.

A estrutura das duas seções segue um padrão onde o sistema é apresentado após uma breve análise de alternativas para as novas regras de inferência. Após a introdução das regras, lemas de geração são apresentados e finalmente a propriedade de SR é investigada. As provas de SR são feitas através da análise de cada regra do cálculo correspondente.

O conjunto de tipos $\mathcal{T}$ utilizado nesses sistemas é o mesmo utilizado no sistema λ_{dB}^{SM}, introduzido na Definição 5.1.1.1. Assim, as propriedades sobre os tipos descrita no Lema 5.1.2 e sobre contextos no Lema 5.1.3 são válidas nos sistemas introduzidos neste capítulo.

7.1 Tipos com interseção para o λs-calculus

O primeiro passo em direção a um sistema de tipos com interseção para o λs_e-calculus será o estudo de um sistema de tipos para o λs-calculus baseado em λ_{dB}^{SM}. O λs é a extensão natural do λ_{dB}, que torna a substituição explícita incluíndo operadores de substituição e atualização no cálculo. Portanto, para um sistema de tipos com interseção para o λs nós estendemos o sistema λ_{dB}^{SM}, com regras de inferência para esses operadores.

Para uma regra de inferência para o operador de atualização φ podemos observar o Lema 5.1.9 como referência, que relaciona tipagens em λ_{dB}^{SM} e i-elevações. Assim, como φ_k^i representa $i-1$ aplicações da k-elevação em λs, temos as seguintes regras:

$$(\omega\text{-}\varphi)\ \frac{M : \langle \Gamma \vdash \tau\rangle}{\varphi_k^i M : \langle \Gamma_{\leq k}.\omega^{\underline{i-1}}.\Gamma_{>k} \vdash \tau\rangle},\ \text{onde } |\Gamma| > k$$

$$(nil\text{-}\varphi)\ \frac{M : \langle \Gamma \vdash \tau\rangle}{\varphi_k^i M : \langle \Gamma \vdash \tau\rangle},\ \text{onde } |\Gamma| \leq k$$

A regra $(nil\text{-}\varphi)$ evita que o nil do contexto original seja trocado por um contexto omega.

Para o operador de substituição σ, devemos observar o Lema 5.1.10. O termo $M\sigma^i N$ está relacionado com o termo $\{\underline{i}/N\}M$. A partir dos Lemas 5.1.10.3 e 5.1.10.4 temos as seguintes regras:

$$(\wedge\text{-}nil\text{-}\sigma)\ \frac{N : \langle nil \vdash \sigma_1\rangle \ldots N : \langle nil \vdash \sigma_m\rangle \qquad M : \langle \omega^{\underline{i-1}}.\wedge_{j=1}^m \sigma_j.nil \vdash \tau\rangle}{M\sigma^i N : \langle nil \vdash \tau\rangle}$$

$$(\wedge\text{-}\omega\text{-}\sigma)\ \frac{N\!:\!\langle nil \vdash \sigma_1\rangle \ldots N\!:\!\langle nil \vdash \sigma_m\rangle \qquad M\!:\!\langle\Gamma \vdash \tau\rangle}{M\sigma^i N\!:\!\langle \Gamma_{<(i-k)}.nil \vdash \tau\rangle},\ \text{onde } \Gamma_i = \wedge_{j=1}^m \sigma_j$$

$$(\wedge\text{-}\sigma)\ \frac{N\!:\!\langle \Delta^1 \vdash \sigma_1\rangle \ldots N\!:\!\langle \Delta^m \vdash \sigma_m\rangle \qquad M\!:\!\langle\Gamma \vdash \tau\rangle}{M\sigma^i N\!:\!\langle (\Gamma_{<i}.\Gamma_{>i}) \wedge \omega^{\underline{i-1}}.(\Delta^1 \wedge \cdots \wedge \Delta^m) \vdash \tau\rangle},\ \text{onde } \Gamma_i = \wedge_{j=1}^m \sigma_j$$

Onde, na regra $(\wedge\text{-}\omega\text{-}\sigma)$ tem-se que $\Gamma = \Gamma_{<(i-k)}.\omega^{\underline{k}}.\wedge_{j=1}^m \sigma_j.nil$ e $\Gamma_{(i-k-1)} \neq \omega$ e na regra $(\wedge\text{-}\sigma)$ tem-se que $\Delta^k \neq nil$, para algum $1 \leq k \leq m$, ou $\Gamma_{>i} \neq \omega^{\underline{n}}$.

Note que as regras $(\wedge\text{-}nil\text{-}\sigma)$ e $(\wedge\text{-}\omega\text{-}\sigma)$ estão relacionadas ao caso no Lema 5.1.10.3 em que o índice a ser substituído é o $sup(M)$. Se a regra $(\wedge\text{-}\sigma)$ fosse aplicada para os casos tratados por essas regras, teríamos tipagens de termos em λs com contextos omega.

Exemplo 7.1.1: Seja $\underline{3} : \langle \omega^{\underline{2}}.\alpha \to \alpha \vdash \alpha \to \alpha\rangle$ e $\lambda.\underline{1} : \langle nil \vdash \alpha \to \alpha\rangle$ em λ_{dB}^{SM}. Aplicando a regra $(\wedge\text{-}\sigma)$, e ignorando a condição de aplicação descrita acima, tem-se que $\underline{3}\sigma^3(\lambda.\underline{1})$: $\langle \omega^{\underline{2}} \vdash \alpha \to \alpha\rangle$.

Portanto, com as regras definidas com condições laterais de aplicação como acima, podemos apresentar mais adiante um conceito apropriado de relevância para o sistema.

Para os casos de substituição vazia tem-se as seguintes regras:

$$(nil\text{-}\sigma)\ \frac{N\!:\!\langle \Delta \vdash \rho\rangle \qquad M\!:\!\langle\Gamma \vdash \tau\rangle}{M\sigma^i N\!:\!\langle \Gamma \wedge (\omega^{\underline{i-1}}.\Delta) \vdash \tau\rangle},\ \text{onde } |\Gamma| < i$$

$$(\omega\text{-}\sigma)\ \frac{N\!:\!\langle \Delta \vdash \rho\rangle \qquad M\!:\!\langle\Gamma \vdash \tau\rangle}{M\sigma^i N\!:\!\langle (\Gamma_{<i}.\Gamma_{>i}) \wedge (\omega^{\underline{i-1}}.\Delta) \vdash \tau\rangle},\ \text{onde } \Gamma_i = \omega$$

Observe que, apesar das regras $(nil\text{-}\sigma)$ e $(\omega\text{-}\sigma)$ estarem relacionadas aos Lemas 5.1.10.1 e 5.1.10.2, respectivamente, mantemos a princípio a informação da tipagem de N. Sabe-se que esta informação deve desaparecer na tipagem da s-nf correspondente, pois o termo N desaparece. A questão é em que momento essa informação deve ser descartada. No exemplo a seguir temos que ao mantermos essa informação, como nas regras acima, perdemos a propriedade de SR.

Exemplo 7.1.2: Tomando o sistema de tipos formado por λ_{dB}^{SM} estendido com as regras propostas acima tem-se que:

$$\cfrac{\cfrac{\cfrac{\underline{2}\!:\!\langle \omega.\alpha_1 \to \alpha_2.nil \vdash \alpha_1 \to \alpha_2\rangle \qquad \underline{3}\!:\!\langle \omega^{\underline{2}}.\alpha_1.nil \vdash \alpha_1\rangle}{(\underline{2}\ \underline{3})\!:\!\langle \omega.\alpha_1 \to \alpha_2.\alpha_1.nil \vdash \alpha_2\rangle}}{\lambda.(\underline{2}\ \underline{3})\!:\!\langle \alpha_1 \to \alpha_2.\alpha_1.nil \vdash \omega \to \alpha_2\rangle} \qquad \underline{3}\!:\!\langle \omega^{\underline{2}}.\beta.nil \vdash \beta\rangle}{(\lambda.(\underline{2}\ \underline{3})\ \underline{3})\!:\!\langle \alpha_1 \to \alpha_2.\alpha_1.\beta.nil \vdash \alpha_2\rangle}$$

Sejam $M \equiv (\lambda.(\underline{2}\ \underline{3})\ \underline{3})$ e $M' \equiv ((\underline{2}\,\sigma^1\underline{3})\ (\underline{3}\,\sigma^1\underline{3}))$. Tem-se que $M \longrightarrow^*_{\lambda s_e} M'$ e que $M' : \langle \alpha_1 \rightarrow \alpha_2.\alpha_1.(\beta \wedge \beta).nil \vdash \alpha_2 \rangle$. Sejam Γ e Γ' os contextos das tipagens de M e de M', respectivamente. Note que $\Gamma' = \Gamma \wedge (\omega^{\,2}.\beta.nil)$.

Portanto, apesar da informação de tipo desaparecer na tipagem da s-nf correspondente, esta é duplicada na tipagem do termo resultante após cada redução com a regra (σ-app-transition). Assim, a informação do contexto de um termo aplicado a ω deve ser "esquecida" assim que o termo integre o corpo de uma substituição. As regras (ω-σ) e (nil-σ) são então modificadas como segue:

$$(nil\text{-}\sigma)\ \frac{N:\langle \Delta \vdash \rho \rangle \qquad M:\langle \Gamma \vdash \tau \rangle}{M\sigma^i N:\langle \Gamma \vdash \tau \rangle},\ \text{onde } |\Gamma| < i$$

$$(\omega\text{-}\sigma)\ \frac{N:\langle \Delta \vdash \rho \rangle \qquad M:\langle \Gamma \vdash \tau \rangle}{M\sigma^i N:\langle \Gamma_{<i}.\Gamma_{>i} \vdash \tau \rangle},\text{onde } \Gamma_i = \omega$$

Observe que as duas regras podem ser substituídas por uma única regra onde a condição para Γ seria a disjunção das condições acima. Mantemos as duas regras neste sistema para ter uma correspondência direta com os Lemas 5.1.10.1 e 5.1.10.2. Assim, com o sistema λ^{SM}_{dB} estendido com as regras descritas acima introduzimos o sistema λs^{SM}.

7.1.1 O sistema λs^{SM} e propriedades

Definição 7.1.1 (O sistema λs^{SM}): As regras de inferência de tipagem do sistema λs^{SM} são compostas pelas regras de λ^{SM}_{dB}, apresentadas na Definição 5.1.4, e as regras introduzidas na Figura 7.1.1.

Para uma noção apropriada de relevância para sistemas de tipos para o λs-calculus, usamos um conceito diferente do representado pelos índice livre. A noção de índice disponível apresentada na Definição 3.2.7 nos permite apresentar o lema de relevância para λs^{SM} a seguir.

Lema 7.1.2 (Relevância para λs^{SM}): Se $M:\langle \Gamma \vdash_{\lambda s^{SM}} \tau \rangle$ então:

i) $|\Gamma| = sav(M)$.

ii) $\forall 1 \leq i \leq |\Gamma|,\ \Gamma_i \neq \omega$ sse $i \in AI(M)$.

$$(nil\text{-}\sigma)\ \frac{N:\langle\Delta\vdash\rho\rangle \qquad M:\langle\Gamma\vdash\tau\rangle}{M\sigma^i N:\langle\Gamma\vdash\tau\rangle},\ |\Gamma|<i$$

$$(\omega\text{-}\sigma)\ \frac{N:\langle\Delta\vdash\rho\rangle \qquad M:\langle\Gamma\vdash\tau\rangle}{M\sigma^i N:\langle\Gamma_{<i}.\Gamma_{>i}\vdash\tau\rangle},\ \Gamma_i=\omega$$

$$(\wedge\text{-}nil\text{-}\sigma)\ \frac{N:\langle nil\vdash\sigma_1\rangle\ldots N:\langle nil\vdash\sigma_m\rangle \qquad M:\langle\underline{\omega^{i-1}}.\wedge_{j=1}^m\sigma_j.nil\vdash\tau\rangle}{M\sigma^i N:\langle nil\vdash\tau\rangle}$$

$$(\wedge\text{-}\omega\text{-}\sigma)\ \frac{N:\langle nil\vdash\sigma_1\rangle\ldots N:\langle nil\vdash\sigma_m\rangle \qquad M:\langle\Gamma\vdash\tau\rangle}{M\sigma^i N:\langle\Gamma_{<(i-k)}.nil\vdash\tau\rangle},\ \Gamma_i=\wedge_{j=1}^m\sigma_j\ (*)$$

$$(\wedge\text{-}\sigma)\ \frac{N:\langle\Delta^1\vdash\sigma_1\rangle\ldots N:\langle\Delta^m\vdash\sigma_m\rangle \qquad M:\langle\Gamma\vdash\tau\rangle}{M\sigma^i N:\langle(\Gamma_{<i}.\Gamma_{>i})\wedge\underline{\omega^{i-1}}.(\Delta^1\wedge\cdots\wedge\Delta^m)\vdash\tau\rangle},\ \Gamma_i=\wedge_{j=1}^m\sigma_j\ (**)$$

$$(\omega\text{-}\varphi)\ \frac{M:\langle\Gamma\vdash\tau\rangle}{\varphi_k^i M:\langle\Gamma_{\le k}.\underline{\omega^{i-1}}.\Gamma_{>k}\vdash\tau\rangle},\ |\Gamma|>k \quad (nil\text{-}\varphi)\ \frac{M:\langle\Gamma\vdash\tau\rangle}{\varphi_k^i M:\langle\Gamma\vdash\tau\rangle},\ |\Gamma|\le k$$

(*) $\Gamma=\Gamma_{<(i-k)}.\underline{\omega^k}.\wedge_{j=1}^m\sigma_j.nil$ e $\Gamma_{(i-k-1)}\neq\omega$
(**) $\Delta^k\neq nil$, para algum $1\le k\le m$, ou $\Gamma_{>i}\neq nil$

Figura 7.1.1: Regras de inferência para o sistema λs^{SM}

Demonstração. Por indução na derivação de $M:\langle\Gamma\vdash_{\lambda s^{SM}}\tau\rangle$. As provas para as regras var, varn, $\to_i$, $\to_i'$, $\to_e$ e $\to_e'$ são análogas a prova do Lema 5.1.7, fazendo uso da definição de AI, sav, e os Lemas 3.2.9, 3.2.10.1 e 3.2.10.2. A seguir, apresentamos a demonstração para as regras de inferência restantes.

- Seja $\dfrac{M:\langle\Gamma\vdash\tau\rangle}{\varphi_k^i M:\langle\Gamma\vdash\tau\rangle}$, onde $|\Gamma|\le k$. Por HI tem-se que $|\Gamma|=sav(M)$ e $\forall 1\le j\le|\Gamma|$, $\Gamma_j\neq\omega$ sse $\underline{j}\in AI(M)$. Assim, pelo Lema 3.2.10.3 tem-se que $sav(\varphi_k^i M)=sav(M)$. Note que $AI(\varphi_k^i M)=AI(M)_{\le k}\cup(AI(M)_{>k}+(i-1))=AI(M)$.

- Seja $\dfrac{M:\langle\Gamma\vdash\tau\rangle}{\varphi_k^i M:\langle\Gamma_{\le k}.\underline{\omega^{i-1}}.\Gamma_{>k}\vdash\tau\rangle}$, onde $|\Gamma|>k$. Por HI tem-se que $|\Gamma|=sav(M)$ e $\forall 1\le j\le|\Gamma|$, $\Gamma_j\neq\omega$ sse $\underline{j}\in AI(M)$. Pelo Lema 3.2.10.3 tem-se que $sav(\varphi_k^i M)=sav(M)+(i{-}1)$. Logo, para $\Gamma'=\Gamma_{\le k}.\underline{\omega^{i-1}}.\Gamma_{>k}$ tem-se que $|\Gamma'|=|\Gamma|+(i{-}1)=sav(M){+}(i{-}1)=sav(\varphi_k^i M)$. Tem-se que $AI(\varphi_k^i M)=AI(M)_{\le k}\cup(AI(M)_{>k}{+}(i{-}1))$. Portanto, tem-se que $\forall 1\le j\le k$, $\Gamma_j'=\Gamma_j\neq\omega$ sse $\underline{j}\in AI(M)_{\le k}$ e $\forall k<j<k{+}i$, $\Gamma_j'=\omega$ e $\forall k{+}i\le j\le|\Gamma|{+}(i{-}1)=|\Gamma'|$, $\Gamma_j'=\Gamma_{j-i+1}\neq\omega$ sse $\underline{j{-}i{+}1}\in AI(M)_{\le k}$ sse $\underline{j}\in(AI(M)_{>k}+(i-1))$.

- Seja $\dfrac{N:\langle\Delta\vdash\rho\rangle \qquad M:\langle\Gamma\vdash\tau\rangle}{M\sigma^i N:\langle\Gamma\vdash\tau\rangle}$, onde $|\Gamma|<i$. Por HI tem-se que $|\Gamma|=sav(M)$ e

$i \notin AI(M)$ logo, pelos Lemas 3.2.10.5 e 3.2.10.4, $sav(M\sigma^i N) = sav(M^{-i}) = sav(M)$. Note que $AI(M\sigma^i N) = AI(M^{-i}) = AI(M)$.

- Seja $\dfrac{N:\langle \Delta \vdash \rho\rangle \qquad M:\langle \Gamma \vdash \tau\rangle}{M\sigma^i N:\langle \Gamma_{<i}.\Gamma_{>i} \vdash \tau\rangle}$, onde $\Gamma_i = \omega$. Por HI tem-se que $sav(M) = |\Gamma|$ e $\underline{i} \notin AI(M)$. Seja $\Gamma' = \Gamma_{<i}.\Gamma_{>i}$. Assim, pelos Lemas 3.2.10.5 e 3.2.10.4, $sav(M\sigma^i N) = sav(M^{-i}) = sav(M) - 1 = |\Gamma| - 1 = |\Gamma'|$. Note que $AI(M\sigma^i N) = AI(M^{-i})$ onde $(AI(M)_{>i})\backslash 1 \neq \emptyset$. Portanto, $\forall 1 \leq j < i$, $\Gamma'_j = \Gamma_j \neq \omega$ sse $\underline{j} \in AI(M)_{<i}$ e $\forall i \leq j < sav(M)$, $\Gamma'_j = \Gamma_{j+1} \neq \omega$ sse $\underline{j+1} \in AI(M)_{>i}$ sse $\underline{j} \in (AI(M)_{>i})\backslash 1$.

- Seja $\dfrac{N:\langle nil \vdash \sigma_1\rangle \ldots N:\langle nil \vdash \sigma_m\rangle \qquad M:\langle \omega^{\underline{i-1}}.\wedge_{j=1}^m \sigma_j.nil \vdash \tau\rangle}{M\sigma^i N:\langle nil \vdash \tau\rangle}$. Por HI tem-se que $sav(N) = 0$ e $AI(M) = \{\,\underline{i}\,\}$. Portanto, $AI(M\sigma^i N) = AI(M^{-i}) \cup AI(\varphi_0^i N) = \emptyset$.

- Seja $\dfrac{N:\langle nil \vdash \sigma_1\rangle \ldots N:\langle nil \vdash \sigma_m\rangle \qquad M:\langle \Gamma \vdash \tau\rangle}{M\sigma^i N:\langle \Gamma_{<(i-k)}.nil \vdash \tau\rangle}$, onde $\Gamma = \Gamma_{<(i-k)}.\omega^{\underline{k}}.\wedge_{j=1}^m \sigma_j.nil$ e $\Gamma_{(i-k-1)} \neq \omega$. Por HI tem-se que $sav(N) = 0$, $sav(M) = i$, $\underline{i-k-1} \in AI(M)$ e $\forall (i-k) \leq j \leq i$, $\underline{j} \notin AI(M)$. Pelo Lema 3.2.10.3 tem-se que $sav(\varphi_0^i N) = sav(N)$. Note que $AI(M^{-i}) = AI(M)_{<i} = AI(M)_{<(i-k)}$ e que $sav(M^{-i}) = i-k-1$. Assim, pelo Lema 3.2.10.5 tem-se que $sav(M\sigma^i N) = max(sav(M^{-i}), sav(\varphi_0^i N)) = i-k-1 = |\Gamma_{<(i-k)}.nil|$. Note que $AI(M\sigma^i N) = AI(M^{-i}) \cup AI(\varphi_0^i N) = AI(M)_{<(i-k)}$ e seja $\Gamma' = \Gamma_{<(i-k)}.nil$. Portanto, $\forall 1 \leq j \leq i-k-1$, $\Gamma'_j = \Gamma_j \neq \omega$ sse $\underline{j} \in AI(M)_{<(i-k)} = AI(M\sigma^i N)$.

- Seja $\dfrac{N:\langle \Delta^1 \vdash \sigma_1\rangle \ldots N:\langle \Delta^m \vdash \sigma_m\rangle \qquad M:\langle \Gamma \vdash \tau\rangle}{M\sigma^i N:\langle (\Gamma_{<i}.\Gamma_{>i}) \wedge \omega^{\underline{i-1}}.(\Delta^1 \wedge \cdots \wedge \Delta^m) \vdash \tau\rangle}$, onde $\Gamma_i = \wedge_{j=1}^m \sigma_j$ e $\Delta^k \neq nil$ ou $\Gamma_{>i} \neq nil$. Suponha que $sav(N) \neq 0$. Por HI tem-se que $\forall 1 \leq k \leq m$, $|\Delta^k| = sav(N)$ e $\forall 1 \leq j \leq |\Delta^k|$, $\Delta^k_j \neq \omega$ sse $\underline{j} \in AI(N)$. Seja $\Delta' = \omega^{\underline{i-1}}.(\Delta^1 \wedge \cdots \wedge \Delta^m)$. Pelo Lema 3.2.10.3 tem-se que $sav(\varphi_0^i N) = sav(N) + (i-1)$. Note que $AI(\varphi_0^i N) = AI(N) + (i-1)$ e $\forall 1 \leq j \leq i-1$ tem-se $\Delta'_j = \omega$ e $\forall i \leq j \leq |\Delta'| = |\Delta^k| + (i-1)$ tem-se $\Delta'_j = (\Delta^1 \wedge \cdots \wedge \Delta^m)_{j-i+1} \neq \omega$ sse $\underline{j-i+1} \in AI(N)$ sse $\underline{j} \in AI(N) + (i-1)$. Também por HI tem-se que $|\Gamma| = sav(M)$ e $\forall 1 \leq j \leq |\Gamma|$, $\Gamma_j \neq \omega$ sse $\underline{j} \in AI(M)$. Suponha que $\Gamma_{>i} \neq nil$ e seja $\Gamma' = (\Gamma_{<i}.\Gamma_{>i})$. Portanto, $sav(M^{-i}) = sav(M) - 1 = |\Gamma'|$ e $\forall 1 \leq j \leq |\Gamma'|$, $\Gamma'_j \neq \omega$ sse $\underline{j} \in AI(M^{-i})$. Assim, $\forall 1 \leq j \leq |\Gamma' \wedge \Delta'|$, $(\Gamma' \wedge \Delta')_j \neq \omega$ sse $\Gamma'_j \neq \omega$ ou $\Delta'_j \neq \omega$ sse $\underline{j} \in AI(M^{-i})$ ou $\underline{j} \in AI(\varphi_0^i N)$ sse $\underline{j} \in AI(M^{-i}) \cup AI(\varphi_0^i N) = AI(M\sigma^i N)$, pois $\Gamma_i \neq \omega$ logo $\underline{i} \in AI(M)$. Além disso, $|\Gamma' \wedge \Delta'| = max(|\Gamma'|, |\Delta'|) = max(sav(M^{-i}), sav(\varphi_0^i N)) = sav(M\sigma^i N)$. O caso para $sav(N) = 0$, e consequentemente por HI $\Delta^k = nil$, e para

$\Gamma_{>i} = nil$ são similares. □

A seguir apresentamos um lema de geração que representa as propriedades que λs^{SM} e λ_{dB}^{SM} têm em comum.

Lema 7.1.3 (Geração para λs^{SM}):

1. Se $\underline{n} : \langle \Gamma \vdash_{\lambda s^{SM}} \tau \rangle$, então $\Gamma_n = \tau$.

2. Se $\lambda.M : \langle nil \vdash_{\lambda s^{SM}} \tau \rangle$, então $\tau = \omega \to \sigma$ e $M : \langle nil \vdash_{\lambda s^{SM}} \sigma \rangle$ ou $\tau = \wedge_{i=1}^{n} \sigma_i \to \sigma$, $n > 0$, e $M : \langle \wedge_{i=1}^{n} \sigma_i.nil \vdash_{\lambda s^{SM}} \sigma \rangle$ para $\sigma, \sigma_1, \ldots, \sigma_n \in \mathcal{T}$.

3. Se $\lambda.M : \langle \Gamma \vdash_{\lambda s^{SM}} \tau \rangle$ e $|\Gamma| > 0$, então $\tau = u \to \sigma$ para algum $u \in \mathcal{U}$ e $\sigma \in \mathcal{T}$, onde $M : \langle u.\Gamma \vdash_{\lambda s^{SM}} \sigma \rangle$.

4. Se $(M\ N) : \langle \Gamma \vdash_{\lambda s^{SM}} \tau \rangle$ então $\Gamma = \Gamma^1 \wedge \Gamma^2$ tal que $M : \langle \Gamma^1 \vdash_{\lambda s^{SM}} \omega \to \tau \rangle$ e $N : \langle \Gamma^2 \vdash_{\lambda s^{SM}} \rho \rangle$ ou $M : \langle \Gamma^1 \vdash_{\lambda s^{SM}} (\wedge_{i=1}^{m} \sigma_i) \to \tau \rangle$ onde $\Gamma^2 = \Delta^1 \wedge \cdots \wedge \Delta^m$ e $\forall 1 \leq i \leq m$, $N : \langle \Delta^i \vdash_{\lambda s^{SM}} \sigma_i \rangle$.

Demonstração. Os itens 1, 2 e 3 são similares as provas dos itens 1, 3 e 4 do Lema 5.1.8, respectivamente.

4. Por análise de casos na derivação de $(M\ N) : \langle \Gamma \vdash \tau \rangle$. □

A seguir apresentamos um lema de geração para tipagens relacionadas aos operadores de substituição e atualização do λs.

Lema 7.1.4 (Geração para operadores em λs^{SM}):

1. Seja $\varphi_k^i N : \langle \Gamma \vdash_{\lambda s^{SM}} \tau \rangle$. Se $|\Gamma| \leq k$, então $N : \langle \Gamma \vdash_{\lambda s^{SM}} \tau \rangle$. Se $|\Gamma| > k$, então $N : \langle \Gamma_{\leq k}.\Gamma_{\geq k+i} \vdash_{\lambda s^{SM}} \tau \rangle$, onde $\Gamma = \Gamma_{\leq k}.\omega^{\underline{i-1}}.\Gamma_{\geq k+i}$.

2. Se $M \sigma^i N : \langle nil \vdash_{\lambda s^{SM}} \tau \rangle$, então $M : \langle nil \vdash_{\lambda s^{SM}} \tau \rangle$ e $N : \langle \Delta \vdash_{\lambda s^{SM}} \rho \rangle$ ou $M : \langle \omega^{\underline{i-1}}. \wedge_{j=1}^{m} \sigma_j.nil \vdash_{\lambda s^{SM}} \tau \rangle$ onde $\forall 1 \leq j \leq m$, $N : \langle nil \vdash_{\lambda s^{SM}} \sigma_j \rangle$.

3. Se $M \sigma^i N : \langle \Gamma \vdash_{\lambda s^{SM}} \tau \rangle$ e $0 < |\Gamma| < i$, então $M : \langle \Gamma \vdash_{\lambda s^{SM}} \tau \rangle$ e $N : \langle \Delta \vdash_{\lambda s^{SM}} \rho \rangle$ ou $M : \langle \Gamma.\omega^{\underline{n}}. \wedge_{j=1}^{m} \sigma_j.nil \vdash_{\lambda s^{SM}} \tau \rangle$ onde $n \geq 0$, $|\Gamma.\omega^{\underline{n}}. \wedge_{j=1}^{m} \sigma_j.nil| = i$ e $\forall 1 \leq j \leq m$, $N : \langle nil \vdash_{\lambda s^{SM}} \sigma_j \rangle$.

4. Se $M\sigma^i N : \langle \Gamma \vdash_{\lambda s^{SM}} \tau \rangle$ e $|\Gamma| \geq i$ então $M : \langle \Gamma_{<i}.\omega.\Gamma_{\geq i} \vdash_{\lambda s^{SM}} \tau \rangle$ e $N : \langle \Delta \vdash_{\lambda s^{SM}} \rho \rangle$ ou $M : \langle \Gamma_{<i}. \wedge_{j=1}^{m} \sigma_j.\Gamma' \vdash_{\lambda s^{SM}} \tau \rangle$ onde $\Gamma_{\geq i} = \Gamma' \wedge \Delta^1 \wedge \cdots \wedge \Delta^m$, para $|\Gamma_{\geq i}| > 0$, e $\forall 1 \leq j \leq m$, $N : \langle \Delta^j \vdash_{\lambda s^{SM}} \sigma_j \rangle$.

Demonstração. 1. Por análise de casos na derivação de $\varphi_k^i N : \langle \Gamma \vdash \tau \rangle$.

2. Por análise de casos na derivação de $M\sigma^i N : \langle nil \vdash \tau \rangle$.

3. Por análise de casos na derivação de $M\sigma^i N : \langle \Gamma \vdash \tau \rangle$, para $0 < |\Gamma| < i$.

4. Por análise de casos na derivação de $M\sigma^i N : \langle \Gamma \vdash \tau \rangle$, para $|\Gamma| \geq i$. □

Observe que as possibilidades apresentadas nos itens do Lema 7.1.4 acima são diferenciadas por características sintáticas dos respectivos termos. Seja $M\sigma^i N : \langle \Gamma \vdash \tau \rangle$. A relação entre i e o comprimento $|\Gamma|$ determina qual dos itens acima é aplicável. Suponha que $0 < |\Gamma| < i$. Assim, pelo item 3 acima tem-se duas possibilidades. No caso do sistema λs^{SM}, para saber qual delas se aplica basta saber o valor de $sav(M)$. Portanto, se $sav(M) < i$, então $M : \langle \Gamma \vdash \tau \rangle$ e $N : \langle \Delta \vdash \rho \rangle$. Se $sav(M) > i$ então a segunda alternativa referente ao item 3 é a que se aplica.

Porém, mantemos essa informação fora dos lemas de geração para λs^{SM} pois no caso do sistema proposto para o λs_e, não temos essa relação direta entre tipagens e propriedades puramente sintáticas dos termos.

7.1.2 Redução de sujeito

As s-nf são exatamente os termos de λ_{dB}, onde a teoria de λs^{SM} coincide com λ_{dB}^{SM}. Assim, a propriedade de redução de sujeito será valida apenas para a β-contração e com a noção de restrição para contextos. Com essa finalidade, provamos SR para o s-calculus.

Teorema 7.1.5 (SR para s em λs^{SM}): Seja $M : \langle \Gamma \vdash_{\lambda s^{SM}} \tau \rangle$. Se $M \to_s M'$, então $M' : \langle \Gamma \vdash_{\lambda s^{SM}} \tau \rangle$.

Demonstração. A prova é feita através da análise da propriedade em cada regra de s.

- (σ-λ-transition): Seja $(\lambda.M)\sigma^i N : \langle \Gamma \vdash \tau \rangle$.

Se $\Gamma = nil$, então pelo Lema 7.1.4.2 tem-se dois casos.

Para o primeiro tem-se $\lambda.M : \langle nil \vdash \tau \rangle$ e $N : \langle \Delta \vdash \rho \rangle$ logo, pelo Lema 7.1.3.2, $M : \langle nil \vdash \tau' \rangle$ onde $\tau = \omega \to \tau'$ ou $M : \langle \wedge_{j=1}^m \sigma_j.nil \vdash \tau' \rangle$ onde $\tau = \wedge_{j=1}^m \sigma_j \to \tau'$. Note que $i \geq 1$ logo $i{+}1 \geq 2$. Assim, por (nil-σ) tem-se $M\sigma^{i+1}N : \langle nil \vdash \tau' \rangle$ ou $M\sigma^{i+1}N : \langle \wedge_{j=1}^m \sigma_j.nil \vdash \tau' \rangle$ logo $\lambda.(M\sigma^{i+1}N) : \langle nil \vdash \omega \to \tau' \rangle$ por $\to_i'$ ou $\lambda.(M\sigma^{i+1}N) : \langle nil \vdash \wedge_{j=1}^m \sigma_j \to \tau' \rangle$ por $\to_i$.

Para o segundo caso tem-se $\lambda.M : \langle \underline{\omega^{i-1}}.\wedge_{j=1}^m \sigma_j.nil \vdash_{\lambda s^{SM}} \tau \rangle$ onde $\forall 1 \leq j \leq m$, $N : \langle nil \vdash \sigma_j \rangle$. Assim, pelo Lema 7.1.3.3 tem-se $M : \langle u.\underline{\omega^{i-1}}.\wedge_{j=1}^m \sigma_j.nil \vdash \tau' \rangle$ onde $\tau = u \to \tau'$. Portanto, se $u \neq \omega$ então por ($\wedge$-ω-σ) tem-se que $M\sigma^{i+1}N : \langle u.nil \vdash \tau' \rangle$ e se $u = \omega$ então por ($\wedge$-nil-σ) tem-se que $M\sigma^{i+1}N : \langle nil \vdash \tau' \rangle$. Logo, $\lambda.(M\sigma^{i+1}N) : \langle nil \vdash u \to \tau' \rangle$ por $\to_i$ ou $\lambda.(M\sigma^{i+1}N) : \langle nil \vdash \omega \to \tau' \rangle$ por $\to_i'$.

Se $0 < |\Gamma| < i$, então pelo Lema 7.1.4.3 tem-se dois casos.

No primeiro caso tem-se que $\lambda.M : \langle \Gamma \vdash \tau \rangle$ e $N : \langle \Delta \vdash \rho \rangle$ logo, pelo Lema 7.1.3.3, tem-se $M : \langle u.\Gamma \vdash \tau' \rangle$ onde $\tau = u \to \tau'$. Note que $|\Gamma|{+}1 < i{+}1$. Assim, por (nil-σ) tem-se $M\sigma^{i+1}N : \langle u.\Gamma \vdash \tau' \rangle$ portanto, por $\to_i$, $\lambda.(M\sigma^{i+1}N) : \langle \Gamma \vdash u \to \tau' \rangle$.

No segundo tem-se que $\lambda.M : \langle \Gamma.\underline{\omega^{n}}.\wedge_{j=1}^m \sigma_j.nil \vdash \tau \rangle$ onde $n \geq 0$, $|\Gamma.\underline{\omega^{n}}.\wedge_{j=1}^m \sigma_j.nil| = i$ e $\forall 1 \leq j \leq m$, $N : \langle nil \vdash \sigma_j \rangle$. Pelo Lema 7.1.3.3 tem-se $M : \langle u.\Gamma.\underline{\omega^{n}}.\wedge_{j=1}^m \sigma_j.nil \vdash \tau' \rangle$ onde $\tau = u \to \tau'$. Assim, por ($\wedge$-ω-σ) tem-se que $M\sigma^{i+1}N : \langle u.\Gamma \vdash \tau' \rangle$ e, por $\to_i$, $\lambda.(M\sigma^{i+1}N) : \langle \Gamma \vdash u \to \tau' \rangle$.

Se $|\Gamma| \geq i$, então pelo Lema 7.1.4.4 tem-se dois casos.

No primeiro caso tem-se que $\lambda.M : \langle \Gamma_{<i}.\omega.\Gamma_{\geq i} \vdash \tau \rangle$ e $N : \langle \Delta \vdash \rho \rangle$ logo, pelo Lema 7.1.3.3, tem-se $M : \langle u.\Gamma_{<i}.\omega.\Gamma_{\geq i} \vdash \tau' \rangle$ onde $\tau = u \to \tau'$. Assim, por (ω-σ), $M\sigma^{i+1}N : \langle u.\Gamma_{<i}.\Gamma_{\geq i} \vdash \tau' \rangle$ e, por $\to_i$, $\lambda.(M\sigma^{i+1}N) : \langle \Gamma_{<i}.\Gamma_{\geq i} \vdash u \to \tau' \rangle$.

No segundo caso tem-se $\lambda.M : \langle \Gamma_{<i}.\wedge_{j=1}^m \sigma_j.\Gamma' \vdash \tau \rangle$ onde $\Gamma_{\geq i} = \Gamma' \wedge \Delta^1 \wedge \cdots \wedge \Delta^m$, para $|\Gamma_{\geq i}| > 0$, e $\forall 1 \leq j \leq m$, $N : \langle \Delta^j \vdash \sigma_j \rangle$. Assim, pelo Lema 7.1.3.3, tem-se $M : \langle u.\Gamma_{<i}.\wedge_{j=1}^m \sigma_j.\Gamma' \vdash \tau' \rangle$ onde $\tau = u \to \tau'$. Logo, por ($\wedge$-σ), $M\sigma^{i+1}N : \langle (u.\Gamma_{<i}.\Gamma') \wedge \underline{\omega^{i}}.(\Delta^1 \wedge \cdots \wedge \Delta^m) \vdash \tau' \rangle$. Observe que $(u.\Gamma_{<i}.\Gamma') \wedge \underline{\omega^{i}}.(\Delta^1 \wedge \cdots \wedge \Delta^m) = u.\Gamma$ portanto, por $\to_i$, $\lambda.(M\sigma^{i+1}N) : \langle \Gamma \vdash u \to \tau' \rangle$.

- (σ-app-transition): Seja $(M_1\ M_2)\sigma^i N : \langle \Gamma \vdash \tau \rangle$.

Se $\Gamma = nil$, então pelo Lema 7.1.4.2 tem-se dois casos.

Para o primeiro tem-se $(M_1\ M_2) : \langle nil \vdash \tau\rangle$ e $N : \langle \Delta \vdash \rho\rangle$. Pelo Lema 7.1.3.4 tem-se que $M_1 : \langle nil \vdash \omega \rightarrow \tau\rangle$ e $M_2 : \langle nil \vdash \rho\rangle$ ou $M_1 : \langle nil \vdash \wedge_{j=1}^{m}\sigma_j \rightarrow \tau\rangle$ e $\forall 1 \leq j \leq m$, $M_2 : \langle nil \vdash \sigma_j\rangle$. Para a primeira alternativa tem-se, pela regra (nil-σ), que $M_1\sigma^i N : \langle nil \vdash \omega \rightarrow \tau\rangle$ e $M_2\sigma^i N : \langle nil \vdash \rho\rangle$ logo, por $\rightarrow'_e$, $((M_1\sigma^i N)\ (M_2\sigma^i N)) : \langle nil \vdash \tau\rangle$. A demonstração para a segunda alternativa é análoga.

Para o segundo caso tem-se $(M_1\ M_2) : \langle \omega \underline{i-1}. \wedge_{k=1}^{l} \tau_k.nil \vdash \tau\rangle$ onde $\forall 1 \leq k \leq l$, $N : \langle nil \vdash \tau_k\rangle$. Pelo Lema 7.1.3.4 tem-se que $\omega \underline{i-1}. \wedge_{k=1}^{l} \tau_k.nil = \Gamma^1 \wedge \Gamma^2$ tal que $M_1 : \langle \Gamma^1 \vdash \omega \rightarrow \tau\rangle$ e $M_2 : \langle \Gamma^2 \vdash \rho\rangle$ ou $M_1 : \langle \Gamma^1 \vdash \wedge_{j=1}^{m}\sigma_j \rightarrow \tau\rangle$ onde $\Gamma^2 = \Delta^1 \wedge \cdots \wedge \Delta^m$ e $\forall 1 \leq j \leq m$, $M_2 : \langle \Delta^j \vdash \sigma_j\rangle$. Note que Γ^1 e Γ^2 representam uma partição com tamanho igual ao contexto original ou um deles é nil. Suponha s.p.d.g. que $\Gamma_2 = nil$ logo $\Gamma^1 = \omega \underline{i-1}. \wedge_{k=1}^{l} \tau_k.nil$. Assim, para a primeira alternativa referente ao Lema 7.1.3.4 tem-se por ($\wedge$-nil-σ) que $M_1\sigma^i N : \langle nil \vdash \omega \rightarrow \tau\rangle$ e por (nil-σ) que $M_2\sigma^i N : \langle nil \vdash \rho\rangle$. Portanto, por $\rightarrow'_e$, $((M_1\sigma^i N)\ (M_2\sigma^i N)) : \langle nil \vdash \tau\rangle$. Para a segunda alternativa a demonstração é análoga.

Se $0 \leq |\Gamma| < i$, então pelo Lema 7.1.4.3 tem-se dois casos.

No primeiro caso tem-se que $(M_1\ M_2) : \langle \Gamma \vdash \tau\rangle$ e $N : \langle \Delta \vdash \rho\rangle$. Pelo Lema 7.1.3.4 tem-se que $\Gamma = \Gamma^1 \wedge \Gamma^2$ tal que $M_1 : \langle \Gamma^1 \vdash \omega \rightarrow \tau\rangle$ e $M_2 : \langle \Gamma^2 \vdash \rho\rangle$ ou $M_1 : \langle \Gamma^1 \vdash \wedge_{j=1}^{m}\sigma_j \rightarrow \tau\rangle$ onde $\Gamma^2 = \Delta^1 \wedge \cdots \wedge \Delta^m$ e $\forall 1 \leq j \leq m$, $M_2 : \langle \Delta^j \vdash \sigma_j\rangle$. Observe que $max(|\Gamma^1|, |\Gamma^2|) = |\Gamma| < i$. Assim, para a primeira alternativa referente ao Lema 7.1.3.4 tem-se por (nil-σ) que $M_1\sigma^i N : \langle \Gamma^1 \vdash \omega \rightarrow \tau\rangle$ e $M_2\sigma^i N : \langle \Gamma^2 \vdash \rho\rangle$ logo, por $\rightarrow'_e$, $((M_1\sigma^i N)\ (M_2\sigma^i N)) : \langle \Gamma^1 \wedge \Gamma^2 \vdash \tau\rangle$. Para a segunda alternativa a demonstração é análoga.

Para o segundo caso tem-se que $(M_1\ M_2) : \langle \Gamma.\omega \underline{n}. \wedge_{k=1}^{l} \tau_k.nil \vdash \tau\rangle$ onde $n \geq 0$, $|\Gamma.\omega \underline{n}. \wedge_{k=1}^{l} \tau_k.nil| = i$ e $\forall 1 \leq k \leq l$, $N : \langle nil \vdash \tau_k\rangle$. Pelo Lema 7.1.3.4 tem-se que $\Gamma.\omega \underline{n}. \wedge_{k=1}^{l} \tau_k.nil = \Gamma^1 \wedge \Gamma^2$ tal que $M_1 : \langle \Gamma^1 \vdash \omega \rightarrow \tau\rangle$ e $M_2 : \langle \Gamma^2 \vdash \rho\rangle$ ou $M_1 : \langle \Gamma^1 \vdash \wedge_{j=1}^{m}\sigma_j \rightarrow \tau\rangle$ onde $\Gamma^2 = \Delta^1 \wedge \cdots \wedge \Delta^m$ e $\forall 1 \leq j \leq m$, $M_2 : \langle \Delta^j \vdash \sigma_j\rangle$. Suponha s.p.d.g. que $\Gamma^1 = \Gamma$ e que $\Gamma^2 = \omega \underline{i-1}. \wedge_{k=1}^{l} \tau_k.nil$ logo $\forall 1 \leq j \leq m$, $\Delta^j = \omega \underline{i-1}.u_j.nil$, onde $u_j \sqsubseteq \wedge_{k=1}^{l}\tau_k$ e $u_1 \wedge \cdots \wedge u_m = \wedge_{k=1}^{l}\tau_k$. Assim, para a segunda alternativa referente ao Lema 7.1.3.4 tem-se por (nil-σ) que $M_1\sigma^i N : \langle \Gamma \vdash \wedge_{j=1}^{m}\sigma_j \rightarrow \tau\rangle$ e por ($\wedge$-nil-σ) que

$\forall 1 \leq j \leq m$, $M_2\sigma^i N : \langle nil \vdash \sigma_j \rangle$. Portanto, por $\rightarrow_e$, $((M_1\sigma^i N)\ (M_2\,\sigma^i N)) : \langle \Gamma \vdash \tau \rangle$. Para a primeira alternativa a demonstração é análoga.

Se $|\Gamma| \geq i$, então pelo Lema 7.1.4.4 tem-se dois casos.

No primeiro caso tem-se que $(M_1\ M_2) : \langle \Gamma_{<i}.\omega.\Gamma_{\geq i} \vdash \tau \rangle$ e $N : \langle \Delta \vdash \rho \rangle$. Pelo Lema 7.1.3.4 tem-se que $\Gamma_{<i}.\omega.\Gamma_{\geq i} = \Gamma^1 \wedge \Gamma^2$ tal que $M_1 : \langle \Gamma^1 \vdash \omega \rightarrow \tau \rangle$ e $M_2 : \langle \Gamma^2 \vdash \rho \rangle$ ou $M_1 : \langle \Gamma^1 \vdash \wedge_{j=1}^m \sigma_j \rightarrow \tau \rangle$ onde $\Gamma^2 = \Delta^1 \wedge \cdots \wedge \Delta^m$ e $\forall 1 \leq j \leq m$, $M_2 : \langle \Delta^j \vdash \sigma_j \rangle$. Suponha s.p.d.g. que $|\Gamma^2| < i$ logo $\Gamma^1 = \Gamma'.\omega.\Gamma_{\geq i}$ para $|\Gamma'| = i-1$ tal que $\Gamma^1 \wedge \Gamma^2 = (\Gamma' \wedge \Gamma^2).\omega.\Gamma_{\geq i}$. Assim, para a primeira alternativa referente ao Lema 7.1.3.4 tem-se por (ω-σ) que $M_1\sigma^i N : \langle \Gamma'.\Gamma_{\geq i} \vdash \omega \rightarrow \tau \rangle$ e por (nil-σ) que $M_2\sigma^i N : \langle \Gamma^2 \vdash \rho \rangle$. Portanto, por $\rightarrow'_e$ tem-se que $((M_1\sigma^i N)\ (M_2\sigma^i N)) : \langle (\Gamma' \wedge \Gamma^2).\Gamma_{\geq i} \vdash \tau \rangle$, onde $\Gamma' \wedge \Gamma^2 = \Gamma_{<i}$. Para a segunda alternativa a demonstração é análoga.

No segundo tem-se $(M_1\ M_2) : \langle \Gamma_{<i}. \wedge_{j=1}^m \sigma_j.\Gamma' \vdash \tau \rangle$ onde $\Gamma_{\geq i} = \Gamma' \wedge \Delta^1 \wedge \cdots \wedge \Delta^m$, para $|\Gamma_{\geq i}| > 0$, e $\forall 1 \leq j \leq m$, $N : \langle \Delta^j \vdash \sigma_j \rangle$. Pelo Lema 7.1.3.4 tem-se que $\Gamma^1 \wedge \Gamma^2 = \Gamma_{<i}. \wedge_{j=1}^m \sigma_j.\Gamma'$ tal que $M_1 : \langle \Gamma^1 \vdash \omega \rightarrow \tau \rangle$ e $M_2 : \langle \Gamma^2 \vdash \rho \rangle$ ou $M_1 : \langle \Gamma^1 \vdash \wedge_{k=1}^l \sigma'_k \rightarrow \tau \rangle$ onde $\Gamma^2 = (\Delta')^1 \wedge \cdots \wedge (\Delta')^l$ e $\forall 1 \leq k \leq l$, $M_2 : \langle (\Delta')^k \vdash \sigma'_k \rangle$. Suponha s.p.d.g. que $|\Gamma^2| < i$ logo $\Gamma^1 = \Gamma^1_{<i}. \wedge_{j=1}^m \sigma_j.\Gamma'$ tal que $\Gamma^1_{<i} \wedge \Gamma^2 = \Gamma_{<i}$. Assim, para a primeira alternativa referente ao Lema 7.1.3.4 tem-se por ($\wedge$-σ) que $M_1\sigma^i N : \langle (\Gamma^1_{<i}.\Gamma') \wedge \underline{\omega^{i-1}}.(\Delta^1 \wedge \cdots \wedge \Delta^m) \vdash \omega \rightarrow \tau \rangle$ e por (nil-σ) que $M_2\sigma^i N : \langle \Gamma^2 \vdash \rho \rangle$. Note que $(\Gamma^1_{<i}.\Gamma') \wedge \underline{\omega^{i-1}}.(\Delta^1 \wedge \cdots \wedge \Delta^m) = \Gamma^1_{<i}.(\Gamma' \wedge \Delta^1 \wedge \cdots \wedge \Delta^m) = \Gamma^1_{<i}.\Gamma_{\geq i}$. Portanto, por $\rightarrow'_e$, $((M_1\sigma^i N)\ (M_2\sigma^i N)) : \langle (\Gamma^1_{<i}.\Gamma_{\geq i}) \wedge \Gamma^2 \vdash \tau \rangle$, onde $(\Gamma^1_{<i}.\Gamma_{\geq i}) \wedge \Gamma^2 = (\Gamma^1_{<i} \wedge \Gamma^2).\Gamma_{\geq i} = \Gamma$. Para a segunda alternativa a demonstração é análoga.

- (σ-destruction): Seja $\underline{n}\,\sigma^i N : \langle \Gamma \vdash \tau \rangle$.

  Se $n < i$, então $\underline{n}\,\sigma^i N \rightarrow \underline{n}$ e $AI(\underline{n}\,\sigma^i N) = \{\,\underline{n}\,\}$ logo, pelo Lema 7.1.2, $|\Gamma| = n$. Observe que para qualquer tipagem $\langle \Gamma' \vdash \tau' \rangle$ de $\underline{n}$, $|\Gamma'| = n$. Assim, pelo Lema 7.1.4.3 tem-se que $\underline{n} : \langle \Gamma \vdash \tau \rangle$ e $N : \langle \Delta \vdash \rho \rangle$.

  Se $n = i$, então $\underline{n}\,\sigma^i N \rightarrow \varphi^i_0 N$ e $AI(\underline{n}\,\sigma^i N) = AI(\varphi^i_0 N)$ logo, pelo Lema 7.1.2, $|\Gamma| = sav(\varphi^i_0 N)$. Pelos Lemas 7.1.2 e 7.1.3.1 tem-se que $\underline{i} : \langle \underline{\omega^{i-1}}.\tau.nil \vdash \tau \rangle$. Se $\Gamma = nil$, então pelo Lema 7.1.4.2 tem-se que $N : \langle nil \vdash \tau \rangle$ logo, por (nil-φ), $\varphi^i_0 N : \langle nil \vdash \tau \rangle$. Se $|\Gamma| > 0$, então pelo Lema 3.2.10.3 tem-se que $sav(\varphi^i_0 N) = sav(N) + (i-1)$, onde $sav(N) > 0$, logo $|\Gamma| = sav(\varphi^i_0 N) \geq i$. Assim, pelo Lema 7.1.4.4 tem-se que

$N:\langle\Gamma_{\geq i}\vdash\tau\rangle$, onde $|\Gamma_{\geq i}|>0$. Portanto, por (ω-φ), $\varphi^i_0 N:\langle\omega^{\underline{i-1}}.\Gamma_{\geq i}\vdash\tau\rangle$. Observe que $\Gamma_{<i}=\omega^{\underline{i-1}}$ logo $\omega^{\underline{i-1}}.\Gamma_{\geq i}=\Gamma$.

Se $n>i$, então $\underline{n}\,\sigma^i N\to\underline{n-1}$ e $AI(\underline{n}\,\sigma^i N)=\{\underline{n-1}\}$ logo, pelo Lema 7.1.2, $|\Gamma|=n-1\geq i$. Tem-se que $\Gamma'_i=\omega$, para qualquer contexto Γ' de uma tipagem de $\underline{n}$, onde $n>i$. Portanto, pelo Lema 7.1.4.4, $\underline{n}:\langle\Gamma_{<i}.\omega.\Gamma_{\geq i}\vdash\tau\rangle$ e $N:\langle\Delta\vdash\rho\rangle$. Pelos Lemas 7.1.2 e 7.1.3.1 tem-se que $\Gamma_{<i}.\omega.\Gamma_{\geq i}=\omega^{\underline{n-1}}.\tau.nil$, logo $\Gamma_{<i}.\Gamma_{\geq i}=\omega^{\underline{n-2}}.\tau.nil$. Portanto, por var e varn tem-se que $\underline{n-1}:\langle\Gamma\vdash\tau\rangle$.

- (φ-λ-transition): Seja $\varphi^i_k(\lambda.M):\langle\Gamma\vdash\tau\rangle$.

  Se $|\Gamma|\leq k$ então pelo Lema 7.1.4.1 tem-se $\lambda.M:\langle\Gamma\vdash\tau\rangle$. Se $\Gamma=nil$, então pelo Lema 7.1.3.2 tem-se $M:\langle nil\vdash\tau'\rangle$ onde $\tau=\omega\to\tau'$ ou $M:\langle\wedge^m_{j=1}\sigma_j.nil\vdash\tau'\rangle$ onde $\tau=\wedge^m_{j=1}\sigma_j\to\tau'$. Observe que $1\leq k+1$. Assim, por (nil-φ), $\varphi^i_{k+1}M:\langle nil\vdash\tau'\rangle$ ou $\varphi^i_{k+1}M:\langle\wedge^m_{j=1}\sigma_j.nil\vdash\tau'\rangle$. Portanto, tem-se $\lambda.(\varphi^i_{k+1}M):\langle nil\vdash\omega\to\tau'\rangle$ por $\to'_i$ ou $\lambda.(\varphi^i_{k+1}M):\langle nil\vdash\wedge^m_{j=1}\sigma_j\to\tau\rangle$ por $\to_i$. A demonstração para $|\Gamma|>0$ é análoga.

  Se $|\Gamma|>k$ então pelo Lema 7.1.4.1 tem-se que $\lambda.M:\langle\Gamma_{\leq k}.\Gamma_{\geq k+i}\vdash\tau\rangle$ onde $\Gamma=\Gamma_{\leq k}.\omega^{\underline{i-1}}.\Gamma_{\geq k+i}$. Assim, pelo Lema 7.1.3.3 tem-se $M:\langle u.\Gamma_{\leq k}.\Gamma_{\geq k+i}\vdash\tau'\rangle$ onde $u\to\tau'$ e, por (ω-φ), $\varphi^i_{k+1}M:\langle u.\Gamma_{\leq k}.\omega^{\underline{i-1}}.\Gamma_{\geq k+i}\vdash\tau\rangle$. Portanto, por $\to_i$ tem-se que $\lambda.(\varphi^i_{k+1}M):\langle\Gamma_{\leq k}.\omega^{\underline{i-1}}.\Gamma_{\geq k+i}\vdash u\to\tau'\rangle$.

- (φ-app-transition): Seja $\varphi^i_k(M_1\ M_2):\langle\Gamma\vdash\tau\rangle$.

  Se $|\Gamma|\leq k$ então pelo Lema 7.1.4.1 tem-se $(M_1\ M_2):\langle\Gamma\vdash\tau\rangle$. Pelo Lema 7.1.3.4 tem-se $\Gamma=\Gamma^1\wedge\Gamma^2$ tal que $M_1:\langle\Gamma^1\vdash\omega\to\tau\rangle$ e $M_2:\langle\Gamma^2\vdash\rho\rangle$ ou $M_1:\langle\Gamma^1\vdash\wedge^m_{j=1}\sigma_j\to\tau\rangle$ onde $\Gamma^2=\Delta^1\wedge\cdots\wedge\Delta^m$ e $\forall 1\leq j\leq m$, $M_2:\langle\Delta^j\vdash\sigma_j\rangle$. Tem-se que $max(|\Gamma^1|,|\Gamma^2|)=|\Gamma|\leq k$. Assim, para a primeira alternativa relativa ao Lema 7.1.3.4 tem-se, por (nil-φ), que $\varphi^i_k M_1:\langle\Gamma^1\vdash\omega\to\tau\rangle$ e $\varphi^i_k M_2:\langle\Gamma^2\vdash\rho\rangle$. Portanto, por $\to'_e$, $((\varphi^i_k M_1)\ (\varphi^i_k M_2)):\langle\Gamma^1\wedge\Gamma^2\vdash\tau\rangle$. A demonstração para a segunda alternativa é similar.

  Se $|\Gamma|>k$ então pelo Lema 7.1.4.1 tem-se que $(M_1\ M_2):\langle\Gamma_{\leq k}.\Gamma_{\geq k+i}\vdash\tau\rangle$ onde $\Gamma=\Gamma_{\leq k}.\omega^{\underline{i-1}}.\Gamma_{\geq k+i}$. Pelo Lema 7.1.3.4 tem-se $\Gamma_{\leq k}.\Gamma_{\geq k+i}=\Gamma^1\wedge\Gamma^2$ tal que $M_1:\langle\Gamma^1\vdash\omega\to\tau\rangle$ e $M_2:\langle\Gamma^2\vdash\rho\rangle$ ou $M_1:\langle\Gamma^1\vdash\wedge^m_{j=1}\sigma_j\to\tau\rangle$ onde $\Gamma^2=\Delta^1\wedge\cdots\wedge\Delta^m$ e $\forall 1\leq j\leq m$, $M_2:\langle\Delta^j\vdash\sigma_j\rangle$. Suponha s.p.d.g. que $|\Gamma^2|\leq k$, logo $\Gamma^1=\Gamma^1_{\leq k}.\Gamma_{\geq k+i}$ onde $\Gamma^1_{\leq k}\wedge\Gamma^2=\Gamma_{\leq k}$. Assim, para a primeira alternativa relativa ao Lema 7.1.3.4 tem-se por (ω-φ) que $\varphi^i_k M_1:\langle\Gamma^1_{\leq k}.\omega^{\underline{i-1}}.\Gamma_{\geq k+i}\vdash\omega\to\tau\rangle$ e por (nil-φ) que $\varphi^i_k M_2:\langle\Gamma^2\vdash\rho\rangle$. Portanto, por $\to'_e$, $(\varphi^i_k M_1\ \varphi^i_k M_2) : \langle(\Gamma^1_{\leq k}.\underline{\omega^{i-1}}.\Gamma_{\geq k+i}) \wedge \Gamma^2 \vdash \tau\rangle$, onde $(\Gamma^1_{\leq k}.\underline{\omega^{i-1}}.\Gamma_{\geq k+i}) \wedge \Gamma^2 = (\Gamma^1_{\leq k} \wedge \Gamma^2).\underline{\omega^{i-1}}.\Gamma_{\geq k+i}$. Para a segunda alternativa a demonstração é análoga.

- (φ-destruction): Seja $\varphi^i_k\,\underline{n} : \langle\Gamma \vdash \tau\rangle$.

  Se $n \leq k$, então $\varphi^i_k\,\underline{n} \to \underline{n}$ e $AI(\varphi^i_k\,\underline{n}) = \{\,\underline{n}\,\}$. Logo, $|\Gamma| = sav(\varphi^i_k\,\underline{n}) = sav(\,\underline{n}) \leq k$. Portanto, pelo Lema 7.1.4.1 tem-se que $\underline{n} : \langle\Gamma \vdash \tau\rangle$.

  Se $n > k$, então $\varphi^i_k\,\underline{n} \to \underline{n{+}i{-}1}$ e $AI(\varphi^i_k\,\underline{n}) = \{\,\underline{n{+}i{-}1}\,\}$. Logo, $|\Gamma| = sav(\varphi^i_k\,\underline{n}) = n{+}(i{-}1) > k$. Portanto, pelo Lema 7.1.4.1 tem-se que $\underline{n} : \langle\Gamma_{\leq k}.\Gamma_{\geq k+i} \vdash \tau\rangle$, onde $\Gamma = \Gamma_{\leq k}.\underline{\omega^{i-1}}.\Gamma_{\geq k+i}$. Pelos Lemmas 7.1.2 e 7.1.3.1 tem-se que $\Gamma_{\leq k}.\Gamma_{\geq k+i} = \underline{\omega^{n-1}}.\tau.nil$. Portanto, $\Gamma = \underline{\omega^{n+i-2}}.\tau.nil$ logo, por var e varn, tem-se que $\underline{n{+}i{-}1} : \langle\Gamma \vdash \tau\rangle$. $\square$

Corolário 7.1.6: Se $M : \langle\Gamma \vdash \tau\rangle$ e $M \to_s M'$ então $AI(M) = AI(M')$.

Demonstração. Pelo Teorema 7.1.5 tem-se que se $M : \langle\Gamma \vdash \tau\rangle$ e $M \to_s M'$ então $M' : \langle\Gamma \vdash \tau\rangle$. Portanto, pelo Lema 7.1.2 tem-se que $\forall 1 \leq i \leq |\Gamma|$, $\underline{i} \in AI(M)$ sse $\Gamma_i \neq \omega$ sse $\underline{i} \in AI(M')$. $\square$

Com a propriedade de SR para o cálculo de substituição s e o fato de λs simular a redução β, podemos demonstrar SR para a simulação da contração β apenas analisando a regra que dá início ao processo. Assim como para o sistema λ^{SM}_{dB}, a noção de SR para λs^{SM} depende de um conceito de restrição para contextos. A seguir, apresentamos a noção de restrição relativa a AI, análoga à Definição 5.1.11.

Definição 7.1.7 (Restrição para contextos): Seja $\Gamma\lfloor_M$ um contexto $\Gamma' \sqsubseteq \Gamma$ tal que $|\Gamma'| = sav(M)$ e que $\forall 1 \leq i \leq |\Gamma'|$, $\Gamma'_i \neq \omega$ sse $\underline{i} \in AI(M)$.

As propriedade da restrição acima são análogas às apresentadas no Lema 5.1.12.

Teorema 7.1.8 (SR para simulação da contração β em λs^{SM}): Seja $(\lambda.M\ M') \in \Lambda_{dB}$. Se $(\lambda.M\ M') : \langle\Gamma \vdash_{\lambda s^{SM}} \tau\rangle$, então $\{\,\underline{1}/M'\}M : \langle\Gamma\lfloor_{\{\,\underline{1}/M'\}M} \vdash_{\lambda s^{SM}} \tau\rangle$.

Demonstração. Seja $(\lambda.M\ M') : \langle\Gamma \vdash \tau\rangle$. Pelo Lema 7.1.3.4 tem-se dois casos.

No primeiro caso tem-se $\lambda.M : \langle\Gamma^1 \vdash \omega \to \tau\rangle$ e $M' : \langle\Gamma^2 \vdash \rho\rangle$, onde $\Gamma = \Gamma^1 \wedge \Gamma^2$.

Se $\Gamma^1 = nil$, então pelo Lema 7.1.3.2 tem-se que $M : \langle nil \vdash \tau\rangle$ logo, por (nil-σ), $M\sigma^1 M' : \langle nil \vdash \tau\rangle$. Pelo Teorema 7.1.5 tem-se que $N : \langle nil \vdash \tau\rangle$ para qualquer N tal que

$M\sigma^1 M' \to_s N$. Logo, por indução no número de reduções em s tem-se que $s(M\sigma^1 M')$: $\langle nil \vdash \tau\rangle$, onde $s(M\sigma^1 M') \equiv \{\underline{1}/M'\}M$. Note que, pelo Lema 7.1.2, $AI(M) = \emptyset$ logo $AI(\{\underline{1}/M'\}M) = AI(M\sigma^1 M') = AI(M^{-1}) = \emptyset$. Assim, $(\Gamma^1 \wedge \Gamma^2)\lfloor_{\{\underline{1}/M'\}M} = nil$.

Se $|\Gamma^1| > 0$, então pelo Lema 7.1.3.3 tem-se que $M : \langle \omega.\Gamma^1 \vdash \tau\rangle$ logo, por $(\omega\text{-}\sigma)$, $M\sigma^1 M' : \langle \Gamma^1 \vdash \tau\rangle$. Logo, pelo Teorema 7.1.5, $s(M\sigma^1 M') : \langle \Gamma^1 \vdash \tau\rangle$. Note que, pelo Lema 7.1.2, $\underline{1} \notin AI(M)$ logo $AI(\{\underline{1}/M'\}M) = AI(M\sigma^1 M') = AI(M^{-1}) = AI(\lambda.M)$. Logo, $(\Gamma^1 \wedge \Gamma^2)\lfloor_{\{\underline{1}/M'\}M} = (\Gamma^1 \wedge \Gamma^2)\lfloor_{\lambda.M} = \Gamma_1$.

No segundo caso tem-se $\lambda.M : \langle \Gamma^1 \vdash \wedge_{j=1}^m \sigma_j \to \tau\rangle$ e $\forall 1 \leq j \leq m$, $M' : \langle \Delta^j \vdash \sigma_j\rangle$, onde $\Gamma = \Gamma^1 \wedge \Gamma^2$ para $\Gamma^2 = \Delta^1 \wedge \cdots \wedge \Delta^m$.

Se $\Gamma^1 = nil$, então pelo Lema 7.1.3.2 tem-se que $M_1 : \langle \wedge_{j=1}^m \sigma_j.nil \vdash \tau\rangle$. Se $sav(M') = 0$ então pelo Lema 7.1.2 tem-se que $\forall 1 \leq j \leq m$, $\Delta^j = nil$. Assim, por $(\wedge\text{-}nil\text{-}\sigma)$ tem-se que $M\sigma^1 M' : \langle nil \vdash \tau\rangle$ logo, pelo Teorema 7.1.5, $s(M\sigma^1 M') : \langle nil \vdash \tau\rangle$. Note que $\Gamma^1 \wedge \Gamma^2 = nil$. Se $sav(M') > 0$, então por $(\wedge\text{-}\sigma)$ tem-se que $M\sigma^1 M' : \langle \Delta^1 \wedge \cdots \wedge \Delta^m \vdash \tau\rangle$. Assim, pelo Teorema 7.1.5, $s(M\sigma^1 M') : \langle \Delta^1 \wedge \cdots \wedge \Delta^m \vdash \tau\rangle$. Note que $AI(\{\underline{1}/M'\}M) = AI(M\sigma^1 M') = AI(\varphi_0^1 M') = AI(M')$. Logo, $(\Gamma^1 \wedge \Gamma^2)\lfloor_{\{\underline{1}/M'\}M} = (\Gamma^1 \wedge \Gamma^2)\lfloor_{M'} = \Gamma^2$.

Se $|\Gamma^1| > 0$, então pelo Lema 7.1.3.3 tem-se que $M : \langle \wedge_{j=1}^m \sigma_j.\Gamma^1 \vdash \tau\rangle$ logo, por $(\wedge\text{-}\sigma)$, $M\sigma^1 M' : \langle \Gamma^1 \wedge (\Delta^1 \wedge \cdots \wedge \Delta^m) \vdash \tau\rangle$. Assim, pelo Teorema 7.1.5 tem-se que $s(M\sigma^1 M')$: $\langle \Gamma^1 \wedge \Gamma^2 \vdash \tau\rangle$. Note que $AI(\{\underline{1}/M'\}M) = AI(M\sigma^1 M') = AI(M^{-1}) \cup AI(\varphi_0^1 M') = AI(\lambda.M) \cup AI(M') = AI(\lambda.M\ M')$. □

7.2 Tipos com interseção para o λs_e-calculus

Ao analisarmos o sistema λs^{SM} para o λs_e-calculus, temos que a composição de substituições impossibilita uma caracterização de termos SN em λs_e. A seguir, apresentamos um exemplo.

Exemplo 7.2.1: Sejam $A \equiv (\underline{1}\ \underline{1})$ a autoaplicação, $M \equiv \big(\lambda.(\lambda.\underline{3}\ A)\ \lambda.A\big)$ e $M' \equiv \big(\lambda.(\underline{3}\,\sigma^1 A)\ \lambda.A\big)$. Tem-se por ($\sigma$-generation) que $M \to M'$. Observe que o problema de tipabilidade para M se reduz a tipabilidade do autorreprodutor $R \equiv \big(\lambda.(\underline{1}\ \underline{1})\ \lambda.(\underline{1}\ \underline{1})\big)$, logo M não é tipável em λs^{SM}. Seja

$$\mathcal{D}[\alpha,\beta] = \frac{\underline{1} : \langle \alpha \to \beta.nil \vdash \alpha \to \beta\rangle \qquad \underline{1} : \langle \alpha.nil \vdash \alpha\rangle}{(\underline{1}\ \underline{1}) : \langle (\alpha \to \beta) \wedge \alpha.nil \vdash \beta\rangle}$$

Assim, para M' tem-se a seguinte derivação em λs^{SM}:

$$\dfrac{\dfrac{\dfrac{\mathcal{D}[\alpha_2,\alpha_3] \qquad \underline{3}:\langle\omega^2.\alpha_1.nil \vdash \alpha_1\rangle}{\underline{3}\,\sigma^1 A:\langle\omega.\alpha_1.nil \vdash \alpha_1\rangle}}{\lambda.(\underline{3}\,\sigma^1 A):\langle\alpha_1.nil \vdash \omega \to \alpha_1\rangle} \qquad \dfrac{\mathcal{D}[\alpha_4,\alpha_5]}{\lambda.A:\langle nil \vdash (\alpha_4 \to \alpha_5) \wedge \alpha_4 \to \alpha_5\rangle}}{(\lambda.(\underline{3}\,\sigma^1 A)\ \lambda.A):\langle\alpha_1.nil \vdash \alpha_1\rangle}$$

O termo M é apenas WN em ambos os cálculos mas M' é SN em λs e somente WN em λs_e. Sejam $M'' \equiv (\underline{3}\,\sigma^1 A)\,\sigma^1\lambda.A$ e $M''' \equiv (\underline{3}\,\sigma^2\lambda.A)\,\sigma^1(A\,\sigma^1\lambda.A)$. Note que $M' \to_{(\sigma\text{-gen.})} M'' \to_{(\sigma\text{-}\sigma\text{-trans.})} M'''$ em λs_e e que M'' é tipável com a mesma tipagem de M' enquanto M''' não é tipável em λs^{SM}.

Assim, para uma sistema de tipos com interseção para λs_e alteramos as regras associadas às substituições vazias da seguinte forma:

$$(nil\text{-}\sigma)\ \frac{M:\langle\Gamma \vdash \tau\rangle}{M\sigma^i N:\langle\Gamma \vdash \tau\rangle},\ \text{onde } |\Gamma| < i$$

$$(\omega\text{-}\sigma)\ \frac{M:\langle\Gamma \vdash \tau\rangle}{M\sigma^i N:\langle\Gamma_{<i}.\Gamma_{>i} \vdash \tau\rangle},\text{onde } \Gamma_i = \omega$$

Portanto, retiramos das respectivas premissas a hipótese de N ser tipável e consequentemente o termo M''', apenas WN em λs_e, é tipável no sistema com estas novas regras. Porém, perdemos a possibilidade de uma caracterização de SN através da tipabilidade de um termo no sistema.

Observe que as regras (nil-σ) e (ω-σ) correspondem a aplicações ímplicitas de um argumento com tipo ω. Assim, substituímos a regra $\to'_e$ pela regra $\to^\omega_e$ como segue:

$$\frac{M_1:\langle\Gamma \vdash \omega \to \tau\rangle}{(M_1\ M_2):\langle\Gamma \vdash \tau\rangle}\ \to^\omega_e$$

É importante ressaltar que regras de inferência desta forma são interessantes em sistemas de tipos computacionais, pois permitem a inseção de programas que não são terminantes. Contudo, em relação a propriedade de terminação, os termos associados desta forma ao ω representam computações inúteis.

Obtemos assim um sistema de IT para o λs_e-calculus, apresentado a seguir.

7.2.1 O sistema $\lambda {s_e}^{\wedge}$ e propriedades

O sistema obtido a partir das alterações propostas ao sistema λs^{SM} com o intuito de reobtermos a propriedade de SR é chamado $\lambda {s_e}^{\wedge}$. A seguir, introduzimos o sistema descrito.

Definição 7.2.1 (O sistema $\lambda {s_e}^{\wedge}$): As regras de tipagem do sistema $\lambda {s_e}^{\wedge}$ são dadas a seguir:

$$\frac{}{\underline{1}:\langle\tau.nil \vdash \tau\rangle}\ \text{var} \qquad \frac{M:\langle u.\Gamma \vdash \tau\rangle}{\lambda.M:\langle\Gamma \vdash u\rightarrow\tau\rangle}\ \rightarrow_i$$

$$\frac{\underline{n}:\langle\Gamma \vdash \tau\rangle}{\underline{n+1}:\langle\omega.\Gamma \vdash \tau\rangle}\ \text{varn} \qquad \frac{M:\langle nil \vdash \tau\rangle}{\lambda.M:\langle nil \vdash \omega\rightarrow\tau\rangle}\ \rightarrow_i'$$

$$\frac{M_1:\langle\Gamma \vdash \omega\rightarrow\tau\rangle}{(M_1\ M_2):\langle\Gamma \vdash \tau\rangle}\ \rightarrow_e^{\omega}$$

$$\frac{M_1:\langle\Gamma \vdash \wedge_{i=1}^{n}\sigma_i\rightarrow\tau\rangle \qquad M_2:\langle\Delta^1 \vdash \sigma_1\rangle\ \ldots\ M_2:\langle\Delta^n \vdash \sigma_n\rangle}{(M_1\ M_2):\langle\Gamma\wedge\Delta^1\wedge\cdots\wedge\Delta^n \vdash \tau\rangle}\ \rightarrow_e$$

$$(nil\text{-}\sigma)\ \frac{M:\langle\Gamma \vdash \tau\rangle}{M\sigma^i N:\langle\Gamma \vdash \tau\rangle},\ |\Gamma|<i \qquad (\omega\text{-}\sigma)\ \frac{M:\langle\Gamma \vdash \tau\rangle}{M\sigma^i N:\langle\Gamma_{<i}.\Gamma_{>i} \vdash \tau\rangle},\ \Gamma_i=\omega$$

$$(\wedge\text{-}nil\text{-}\sigma)\ \frac{N:\langle nil \vdash \sigma_1\rangle\ldots N:\langle nil \vdash \sigma_m\rangle \qquad M:\langle\underline{\omega^{i-1}}.\wedge_{j=1}^{m}\sigma_j.nil \vdash \tau\rangle}{M\sigma^i N:\langle nil \vdash \tau\rangle}$$

$$(\wedge\text{-}\omega\text{-}\sigma)\ \frac{N:\langle nil \vdash \sigma_1\rangle\ldots N:\langle nil \vdash \sigma_m\rangle \qquad M:\langle\Gamma \vdash \tau\rangle}{M\sigma^i N:\langle\Gamma_{<(i-k)}.nil \vdash \tau\rangle},\ \Gamma_i=\wedge_{j=1}^{m}\sigma_j\ (*)$$

$$(\wedge\text{-}\sigma)\ \frac{N:\langle\Delta^1 \vdash \sigma_1\rangle\ldots N:\langle\Delta^m \vdash \sigma_m\rangle \qquad M:\langle\Gamma \vdash \tau\rangle}{M\sigma^i N:\langle(\Gamma_{<i}.\Gamma_{>i})\wedge\underline{\omega^{i-1}}.(\Delta^1\wedge\cdots\wedge\Delta^m) \vdash \tau\rangle},\ \Gamma_i=\wedge_{j=1}^{m}\sigma_j\ (**)$$

$$(\omega\text{-}\varphi)\ \frac{M:\langle\Gamma \vdash \tau\rangle}{\varphi_k^i M:\langle\Gamma_{\leq k}.\underline{\omega^{i-1}}.\Gamma_{>k} \vdash \tau\rangle},\ |\Gamma|>k \qquad (nil\text{-}\varphi)\ \frac{M:\langle\Gamma \vdash \tau\rangle}{\varphi_k^i M:\langle\Gamma \vdash \tau\rangle},\ |\Gamma|\leq k$$

(*) $\Gamma=\Gamma_{<(i-k)}.\underline{\omega^k}.\wedge_{j=1}^{m}\sigma_j.nil$ e $\Gamma_{(i-k-1)}\neq\omega$
(**) $\Delta^k\neq nil$, para algum $1\leq k\leq m$, ou $\Gamma_{>i}\neq nil$

Observe que com o sistema de tipos apresentado na Definição 7.2.1 acima, perde-se qualquer relação direta entre o contexto de uma tipagem de um termo e suas características sintáticas, como o FI em λ_{dB}^{SM} e AI em λs^{SM}. Assim não temos um conceito apropriado para o que seria relevância nesse sistema. Apesar disso, o lema a seguir estabelece uma propriedade de contextos de tipagem semelhante aos sistemas anteriores.

Lema 7.2.2: Se $M:\langle \Gamma \vdash_{\lambda s_e^{\wedge}} \tau\rangle$ para $m=|\Gamma|>0$, então $\Gamma_m \neq \omega$.

Demonstração. Por indução na derivação de $M:\langle \Gamma \vdash \tau\rangle$, para $\Gamma \neq nil$.

- Se $\underline{1}:\langle \tau.nil \vdash \tau\rangle$, nada há provar.
- Seja $\dfrac{\underline{n}:\langle \Gamma \vdash \tau\rangle}{\underline{n+1}:\langle \omega.\Gamma \vdash \tau\rangle}$. Por HI tem-se que se $m=|\Gamma|>0$ então $\Gamma_m \neq \omega$. Portanto, $|\omega.\Gamma|=m+1$ e $(\omega.\Gamma)_{m+1}=\Gamma_m \neq \omega$.
- Seja $\dfrac{M:\langle u.\Gamma \vdash \sigma\rangle}{\lambda.M:\langle \Gamma \vdash u \to \sigma\rangle}$. Se $\Gamma = nil$ então por HI $u \neq \omega$ e nada há provar. Se $|\Gamma|=m>0$, então por HI tem-se que $|u.\Gamma|=m+1$ e $(u.\Gamma)_{m+1}=\Gamma_m \neq \omega$.
- Seja $\dfrac{M_1:\langle \Gamma \vdash \omega \to \tau\rangle}{(M_1\ M_2):\langle \Gamma \vdash \tau\rangle}$. Por HI tem-se que se $m=|\Gamma|>0$ então $\Gamma_m \neq \omega$.
- Seja $\dfrac{M_1:\langle \Gamma \vdash \wedge_{k=1}^{n}\sigma_k \to \tau\rangle \qquad M_2:\langle \Delta^1 \vdash \sigma_1\rangle \ \ldots\ M_2:\langle \Delta^n \vdash \sigma_n\rangle}{(M_1\ M_2):\langle \Gamma \wedge \Delta^1 \wedge \cdots \wedge \Delta^n \vdash \tau\rangle}$. Suponha que $m=|\Gamma \wedge \Delta^1 \wedge \cdots \wedge \Delta^n| = max(|\Gamma|, |\Delta^1|, \ldots, |\Delta^n|) > 0$. Suponha s.p.d.g. que $|\Gamma| < m$ e $|\Delta^{j_1}|, \ldots, |\Delta^{j_l}| = m$, onde $l \leq n$. Por HI tem-se que $\forall 1 \leq i \leq l$, $\Delta^{j_i}_m \neq \omega$. Portanto, $(\Gamma \wedge \Delta^1 \wedge \cdots \wedge \Delta^n)_m = \Delta^{j_1}_m \wedge \cdots \wedge \Delta^{j_l}_m \neq \omega$.
- Seja $\dfrac{M:\langle \Gamma \vdash \tau\rangle}{\varphi^i_k M:\langle \Gamma \vdash \tau\rangle}$, onde $|\Gamma| \leq k$. Por HI tem-se que se $m=|\Gamma|>0$ então $\Gamma_m \neq \omega$.
- Seja $\dfrac{M:\langle \Gamma \vdash \tau\rangle}{\varphi^i_k M:\langle \Gamma_{\leq k}.\omega^{\underline{i-1}}.\Gamma_{>k} \vdash \tau\rangle}$, onde $|\Gamma| > k$. Por HI tem-se que se $m=|\Gamma|>0$ então $\Gamma_m \neq \omega$. Portanto, $|\Gamma_{\leq k}.\omega^{\underline{i-1}}.\Gamma_{>k}|=m+(i-1)$ e $(\Gamma_{\leq k}.\omega^{\underline{i-1}}.\Gamma_{>k})_{m+(i-1)}=\Gamma_m \neq \omega$.
- Seja $\dfrac{M:\langle \Gamma \vdash \tau\rangle}{M\sigma^i N:\langle \Gamma \vdash \tau\rangle}$, onde $|\Gamma| < i$. Por HI tem-se que se $m=|\Gamma|>0$ então $\Gamma_m \neq \omega$.
- Seja $\dfrac{M:\langle \Gamma \vdash \tau\rangle}{M\sigma^i N:\langle \Gamma_{<i}.\Gamma_{>i} \vdash \tau\rangle}$, para $\Gamma_i = \omega$. Por HI tem-se que $|\Gamma| = m > i$ e $\Gamma_m \neq \omega$. Portanto, $|\Gamma_{<i}.\Gamma_{>i}|=m-1$ e $(\Gamma_{<i}.\Gamma_{>i})_{m-1}=\Gamma_m \neq \omega$.
- Se $\dfrac{N:\langle nil \vdash \sigma_1\rangle \ldots N:\langle nil \vdash \sigma_m\rangle \qquad M:\langle \Gamma \vdash \tau\rangle}{M\sigma^i N:\langle \Gamma_{<(i-k).nil} \vdash \tau\rangle}$, onde $\Gamma = \Gamma_{<(i-k)}.\omega^{\underline{k}}.\wedge_{j=1}^{m} \sigma_j.nil$ e $\Gamma_{(i-k-1)} \neq \omega$, então nada há provar.
- Seja $\dfrac{N:\langle \Delta^1 \vdash \sigma_1\rangle \ldots N:\langle \Delta^m \vdash \sigma_m\rangle \qquad M:\langle \Gamma \vdash \tau\rangle}{M\sigma^i N:\langle (\Gamma_{<i}.\Gamma_{>i}) \wedge \omega^{\underline{i-1}}.(\Delta^1 \wedge \cdots \wedge \Delta^m) \vdash \tau\rangle}$, onde $\Gamma_i = \wedge_{j=1}^{m}\sigma_j$ e $|\Gamma_{>i} \wedge \Delta^1 \wedge \cdots \wedge \Delta^m| > 0$. Suponha s.p.d.g. que $\Delta^1 \wedge \cdots \wedge \Delta^m = nil$ logo $|\Gamma_{>i}| > 0$. Seja $m=|\Gamma|$, assim $(\Gamma_{<i}.\Gamma_{>i}) \wedge \omega^{\underline{i-1}}.(\Delta^1 \wedge \cdots \wedge \Delta^m)=\Gamma_{<i}.\Gamma_{>i}$ e $|\Gamma_{<i}.\Gamma_{>i}|=m-1$. Por HI tem-se que $\Gamma_m \neq \omega$. Portanto, $(\Gamma_{<i}.\Gamma_{>i})_{m-1}=\Gamma_m \neq \omega$. $\square$

Portanto, o Lema 7.2.2 acima mostra que o maior elemento de um contexto não vazio de uma tipagem é diferente de ω. A seguir, apresentamos um lema de geração para $\lambda {s_e}^{\wedge}$. Assim como para o λs^{SM}, os lemas para termos que tenham um operador em sua raiz são apresentados separadamente.

Lema 7.2.3 (Geração para $\lambda {s_e}^{\wedge}$):

1. Se $\underline{n}:\langle\Gamma \vdash_{\lambda {s_e}^{\wedge}} \tau\rangle$, então $\Gamma = \omega^{\underline{n-1}}.\tau.nil$.

2. Se $\lambda.M:\langle nil \vdash_{\lambda {s_e}^{\wedge}} \tau\rangle$, então $\tau = \omega \rightarrow \sigma$ e $M:\langle nil \vdash_{\lambda {s_e}^{\wedge}} \sigma\rangle$ ou $\tau = \wedge_{i=1}^{n}\sigma_i \rightarrow \sigma$, $n > 0$, e $M:\langle \wedge_{i=1}^{n}\sigma_i.nil \vdash_{\lambda {s_e}^{\wedge}} \sigma\rangle$ para $\sigma, \sigma_1, \ldots, \sigma_n \in \mathcal{T}$.

3. Se $\lambda.M : \langle\Gamma \vdash_{\lambda {s_e}^{\wedge}} \tau\rangle$ e $|\Gamma| > 0$, então $\tau = u \rightarrow \sigma$ para algum $u \in \mathcal{U}$ e $\sigma \in \mathcal{T}$, onde $M:\langle u.\Gamma \vdash_{\lambda {s_e}^{\wedge}} \sigma\rangle$.

4. Se $(M\ N):\langle\Gamma \vdash_{\lambda {s_e}^{\wedge}} \tau\rangle$ então $M:\langle\Gamma \vdash_{\lambda {s_e}^{\wedge}} \omega \rightarrow \tau\rangle$ ou $\Gamma = \Gamma' \wedge \Delta^1 \wedge \cdots \wedge \Delta^m$ tal que $M:\langle\Gamma' \vdash_{\lambda {s_e}^{\wedge}} \wedge_{i=1}^{m}\sigma_i \rightarrow \tau\rangle$ e $\forall 1 \leq i \leq m$, $N:\langle\Delta^i \vdash_{\lambda {s_e}^{\wedge}} \sigma_i\rangle$.

Demonstração. 1. Por indução em n. Para $n = 1$, nada há provar. Seja $\underline{n+1}:\langle\Gamma \vdash \tau\rangle$. Por varn tem-se que $\Gamma = |\omega.\Gamma'|$, onde $\underline{n} : \langle\Gamma' \vdash \tau\rangle$. Assim, por HI tem-se que $\Gamma' = \omega^{\underline{n-1}}.\tau.nil$ logo $\Gamma = \omega^{\underline{n}}.\tau.nil$.

2. Por análise de casos na derivação de $\lambda.M:\langle nil \vdash \tau\rangle$.

3. Por análise de casos na derivação de $\lambda.M:\langle\Gamma \vdash \tau\rangle$, para $|\Gamma| > 0$.

4. Por análise de casos na derivação de $(M\ N):\langle\Gamma \vdash \tau\rangle$. $\square$

Observe que o item 1 do Lema 7.2.3 acima representa um resultado equivalente ao Lema 7.1.3.1 em conjunto com o Lema 7.1.2 no sistema λs^{SM}. Apesar dos itens 2 e 3 serem semelhantes aos Lemas 7.1.3.2 e 7.1.3.3 para λs^{SM}, nesses últimos a alternativa apropriada está ligada aos conjuntos $AI(\lambda.M)$ e $AI(M)$, enquanto nos itens acima não temos essa propriedade. A perda dessa relação é consequência da regra $\rightarrow_e^{\omega}$, exprimida na propriedade apresentada no item 4 acima.

Lema 7.2.4 (Geração para operadores em $\lambda {s_e}^{\wedge}$):

1. Seja $\varphi_k^i N : \langle \Gamma \vdash_{\lambda s_e{}^\wedge} \tau \rangle$. Se $|\Gamma| \leq k$, então $N : \langle \Gamma \vdash_{\lambda s_e{}^\wedge} \tau \rangle$. Se $|\Gamma| > k$, então $N : \langle \Gamma_{\leq k}.\Gamma_{\geq k+i} \vdash_{\lambda s_e{}^\wedge} \tau \rangle$, onde $\Gamma = \Gamma_{\leq k}.\omega \underline{^{i-1}}.\Gamma_{\geq k+i}$.

2. Se $M\sigma^i N : \langle nil \vdash_{\lambda s_e{}^\wedge} \tau \rangle$, então $M : \langle nil \vdash_{\lambda s_e{}^\wedge} \tau \rangle$ ou $M : \langle \omega \underline{^{i-1}}. \wedge_{j=1}^m \sigma_j.nil \vdash_{\lambda s_e{}^\wedge} \tau \rangle$ onde $\forall 1 \leq j \leq m$, $N : \langle nil \vdash_{\lambda s_e{}^\wedge} \sigma_j \rangle$.

3. Se $M\sigma^i N : \langle \Gamma \vdash_{\lambda s_e{}^\wedge} \tau \rangle$ e $0 < |\Gamma| < i$, então $M : \langle \Gamma \vdash_{\lambda s_e{}^\wedge} \tau \rangle$ ou $M : \langle \Gamma' \vdash_{\lambda s_e{}^\wedge} \tau \rangle$ onde $\Gamma' = \Gamma.\omega \underline{^n}. \wedge_{j=1}^m \sigma_j.nil$ para $n \geq 0$ tal que $|\Gamma'| = i$ e $\forall 1 \leq j \leq m$, $N : \langle nil \vdash_{\lambda s_e{}^\wedge} \sigma_j \rangle$.

4. Se $M\sigma^i N : \langle \Gamma \vdash_{\lambda s_e{}^\wedge} \tau \rangle$ e $|\Gamma| \geq i$ então $M : \langle \Gamma_{<i}.\omega.\Gamma_{\geq i} \vdash_{\lambda s_e{}^\wedge} \tau \rangle$ ou $\Gamma_{\geq i} = \Gamma' \wedge \Delta^1 \wedge \cdots \wedge \Delta^m$ para $|\Gamma_{\geq i}| > 0$ tal que $M : \langle \Gamma_{<i}. \wedge_{j=1}^m \sigma_j.\Gamma' \vdash_{\lambda s_e{}^\wedge} \tau \rangle$ e $\forall 1 \leq j \leq m$, $N : \langle \Delta^j \vdash_{\lambda s_e{}^\wedge} \sigma_j \rangle$.

Demonstração. 1. Por análise de casos na derivação de $\varphi_k^i N : \langle \Gamma \vdash \tau \rangle$.

2. Por análise de casos na derivação de $M\sigma^i N : \langle nil \vdash \tau \rangle$.

3. Por análise de casos na derivação de $M\sigma^i N : \langle \Gamma \vdash \tau \rangle$, para $0 < |\Gamma| < i$.

4. Por análise de casos na derivação de $M\sigma^i N : \langle \Gamma \vdash \tau \rangle$, para $|\Gamma| \geq i$. □

7.2.2 Redução de sujeito

Como consequência da não adição de contextos para termos aplicados a ω, o contexto da tipagem de um β-redex permanece sempre inalterado após a aplicação da regra (σ-generation). Consequentemente, ao contrário do feito para o λs^{SM} onde provamos SR sem restrição apenas para o cálculo de substituição associado, podemos provar SR em $\lambda s_e{}^\wedge$ para todo o cálculo λs_e.

Teorema 7.2.5 (SR para $\lambda s_e{}^\wedge$): Se $M : \langle \Gamma \vdash_{\lambda s_e{}^\wedge} \tau \rangle$ e $M \to_{\lambda s_e} M'$, então $M' : \langle \Gamma \vdash_{\lambda s_e{}^\wedge} \tau \rangle$.

Demonstração. A prova é feita através da análise da propriedade em cada regra de reescrita do cálculo.

- (σ-generation): Seja $(\lambda.M\ N) : \langle \Gamma \vdash \tau \rangle$. Pelo Lema 7.2.3.4 tem-se dois casos.

  No primeiro caso, $\lambda.M : \langle \Gamma \vdash \omega \to \tau \rangle$.

Se $\Gamma = nil$, então pelo Lema 7.2.3.2 tem-se que $M : \langle nil \vdash \tau \rangle$. Portanto, por ($nil$-$\sigma$) tem-se que $M\sigma^1 N : \langle nil \vdash \tau \rangle$.

Se $|\Gamma| > 0$, então pelo Lema 7.2.3.3 tem-se que $M : \langle \omega.\Gamma \vdash \tau \rangle$. Portanto, por ($\omega$-$\sigma$) tem-se que $M\sigma^1 N : \langle \Gamma \vdash \tau \rangle$.

No segundo caso, $\Gamma = \Gamma' \wedge \Delta^1 \wedge \cdots \wedge \Delta^m$ tal que $\lambda.M : \langle \Gamma' \vdash \wedge_{j=1}^m \sigma_j \to \tau \rangle$ e $\forall 1 \leq j \leq m$, $N : \langle \Delta^j \vdash \sigma_j \rangle$.

Se $\Gamma' = nil$, então pelo Lema 7.2.3.2 tem-se $M : \langle \wedge_{j=1}^m \sigma_j.nil \vdash \tau \rangle$. Se $\forall 1 \leq j \leq m$, $\Delta^j = nil$, então por ($\wedge$-nil-σ) tem-se que $M\sigma^1 N : \langle nil \vdash \tau \rangle$. Senão, por ($\wedge$-$\sigma$) tem-se que $M\sigma^1 N : \langle \Delta^1 \wedge \cdots \wedge \Delta^m \vdash \tau \rangle$, onde $\Gamma = \Delta^1 \wedge \cdots \wedge \Delta^m$.

Se $|\Gamma'| > 0$, então pelo Lema 7.2.3.3 tem-se que $M : \langle \wedge_{j=1}^m \sigma_j.\Gamma' \vdash \tau \rangle$. Portanto, por ($\wedge$-$\sigma$) tem-se que $M\sigma^1 N : \langle \Gamma' \wedge \Delta^1 \wedge \cdots \wedge \Delta^m \vdash \tau \rangle$.

- (σ-λ-transition): demonstração análoga a demonstração de SR para (σ-λ-transition) em λs^{SM}, apresentada no Teorema 7.1.5.

- (σ-app-transition): Seja $(M_1\ M_2)\,\sigma^i N : \langle \Gamma \vdash \tau \rangle$.

  Se $\Gamma = nil$, então pelo Lema 7.2.4.2 tem-se dois casos.

  Para o primeiro caso tem-se que $(M_1\ M_2) : \langle nil \vdash \tau \rangle$. Assim, pelo Lema 7.2.3.4 tem-se que $M_1 : \langle nil \vdash \omega \to \tau \rangle$ ou $M_1 : \langle nil \vdash \wedge_{j=1}^m \sigma_j \to \tau \rangle$ e $\forall 1 \leq j \leq m$, $M_2 : \langle nil \vdash \sigma_j \rangle$. Para a segunda alternativa tem-se, pela regra (nil-σ), que $M_1\sigma^i N : \langle nil \vdash \wedge_{j=1}^m \sigma_j \to \tau \rangle$ e $\forall 1 \leq j \leq m$, $M_2\sigma^i N : \langle nil \vdash \sigma_j \rangle$ logo, por $\to_e$, $((M_1\sigma^i N)\ (M_2\sigma^i N)) : \langle nil \vdash \tau \rangle$. A demonstração para a primeira alternativa é análoga.

  Para o segundo caso tem-se $(M_1\ M_2) : \langle \underline{\omega^{i-1}}. \wedge_{k=1}^l \tau_k.nil \vdash \tau \rangle$ onde $\forall 1 \leq k \leq l$, $N : \langle nil \vdash \tau_k \rangle$. Pelo Lema 7.2.3.4 tem-se que $M_1 : \langle \underline{\omega^{i-1}}. \wedge_{k=1}^l \tau_k.nil \vdash \omega \to \tau \rangle$ ou $\underline{\omega^{i-1}}. \wedge_{k=1}^l \tau_k.nil = \Gamma' \wedge \Delta^1 \wedge \cdots \wedge \Delta^m$ tal que $M_1 : \langle \Gamma' \vdash \wedge_{j=1}^m \sigma_j \to \tau \rangle$ e $\forall 1 \leq j \leq m$, $M_2 : \langle \Delta^j \vdash \sigma_j \rangle$. Note que $\Gamma', \Delta^1, \ldots, \Delta^m$ representam uma partição com comprimento igual ao contexto original ou nil. Suponha s.p.d.g. que $\Gamma' = nil$. Assim, para a segunda alternativa referente ao Lema 7.2.3.4 tem-se por (nil-σ) que $M_1\sigma^i N : \langle nil \vdash \wedge_{j=1}^m \sigma_j \to \tau \rangle$ e $\forall 1 \leq j \leq m$, se $\Delta^j = nil$ então $M_2\sigma^i N : \langle nil \vdash \sigma_j \rangle$ por (nil-σ) e se $\Delta^j \neq nil$ então $M_2\sigma^i N : \langle nil \vdash \sigma_j \rangle$ por ($\wedge$-nil-σ). Portanto, por

$\rightarrow_e$, $((M_1\sigma^i N)\ (M_2\sigma^i N)) : \langle nil \vdash \tau\rangle$. Para a primeira alternativa a demonstração é análoga.

Se $0 \leq |\Gamma| < i$, então pelo Lema 7.2.4.3 tem-se dois casos.

No primeiro caso tem-se que $(M_1\ M_2) : \langle \Gamma \vdash \tau\rangle$. Pelo Lema 7.2.3.4 tem-se que $M_1 : \langle \Gamma \vdash \omega \rightarrow \tau\rangle$ ou $\Gamma = \Gamma' \wedge \Delta^1 \wedge \cdots \wedge \Delta^m$ tal que $M_1 : \langle \Gamma' \vdash \wedge_{j=1}^m \sigma_j \rightarrow \tau\rangle$ e $\forall 1 \leq j \leq m$, $M_2 : \langle \Delta^j \vdash \sigma_j\rangle$. Observe que $max(|\Gamma'|, |\Delta^1|, \ldots, |\Delta^m|) = |\Gamma| < i$. Assim, para a segunda alternativa referente ao Lema 7.2.3.4 tem-se por (nil-σ) que $M_1\sigma^i N : \langle \Gamma' \vdash \wedge_{j=1}^m \sigma_j \rightarrow \tau\rangle$ e $\forall 1 \leq j \leq m$, $M_2\sigma^i N : \langle \Delta^j \vdash \sigma_j\rangle$ logo, por $\rightarrow_e$, $((M_1\sigma^i N)\ (M_2\sigma^i N)) : \langle \Gamma' \wedge \Delta^1 \wedge \cdots \wedge \Delta^m \vdash \tau\rangle$. Para a primeira alternativa a demonstração é análoga.

No segundo caso tem-se $(M_1\ M_2) : \langle \Gamma' \vdash \tau\rangle$ onde $\Gamma' = \Gamma.\omega^{\underline{n}}. \wedge_{k=1}^l \tau_k.nil$ para $n \geq 0$ tal que $|\Gamma'| = i$ e $\forall 1 \leq k \leq l$, $N : \langle nil \vdash \tau_k\rangle$. Pelo Lema 7.2.3.4 tem-se que $M_1 : \langle \Gamma' \vdash \omega \rightarrow \tau\rangle$ ou $\Gamma' = \Gamma' \wedge \Delta^1 \wedge \cdots \wedge \Delta^m$ tal que $M_1 : \langle \Gamma' \vdash \wedge_{j=1}^m \sigma_j \rightarrow \tau\rangle$ e $\forall 1 \leq j \leq m$, $M_2 : \langle \Delta^j \vdash \sigma_j\rangle$. Suponha s.p.d.g. que $\Gamma' = \Gamma$. Assim, para a segunda alternativa referente ao Lema 7.2.3.4 tem-se por (nil-σ) que $M_1\sigma^i N : \langle \Gamma \vdash \wedge_{j=1}^m \sigma_j \rightarrow \tau\rangle$ e $\forall 1 \leq j \leq m$, por ($\wedge$-nil-σ) ou (nil-σ) tem-se que $M_2\sigma^i N : \langle nil \vdash \sigma_j\rangle$. Portanto, por $\rightarrow_e$, $((M_1\sigma^i N)\ (M_2\sigma^i N)) : \langle \Gamma \vdash \tau\rangle$. Para a primeira alternativa a demonstração é análoga.

Se $|\Gamma| \geq i$, então pelo Lema 7.2.4.4 tem-se dois casos.

No primeiro caso tem-se que $(M_1\ M_2) : \langle \Gamma_{<i}.\omega.\Gamma_{\geq i} \vdash \tau\rangle$. Pelo Lema 7.2.3.4 tem-se que $M_1 : \langle \Gamma_{<i}.\omega.\Gamma_{\geq i} \vdash \omega \rightarrow \tau\rangle$ ou $\Gamma_{<i}.\omega.\Gamma_{\geq i} = \Gamma' \wedge \Delta^1 \wedge \cdots \wedge \Delta^m$ tal que $M_1 : \langle \Gamma' \vdash \wedge_{j=1}^m \sigma_j \rightarrow \tau\rangle$ e $\forall 1 \leq j \leq m$, $M_2 : \langle \Delta^j \vdash \sigma_j\rangle$. Suponha s.p.d.g. que $|\Gamma'| < i$ logo $\forall 1 \leq j \leq m$, se $|\Delta^j| > i$ então $\Delta^j_i = \omega$. Observe que, pelo Lema 7.2.2, $\forall 1 \leq j \leq m$, $|\Delta^j| \neq i$. Assim, para a segunda alternativa referente ao Lema 7.2.3.4 tem-se por (nil-σ) que $M_1\sigma^i N : \langle \Gamma' \vdash \wedge_{j=1}^m \sigma_j \rightarrow \tau\rangle$ e $\forall 1 \leq j \leq m$, se $|\Delta^j| > i$ então $M_2\sigma^i N : \langle \Delta^j_{<i}.\Delta^j_{>i} \vdash \sigma_j\rangle$ por (ω-σ) e se $|\Delta^j| < i$ então $M_2\sigma^i N : \langle \Delta^j \vdash \sigma_j\rangle$ por (nil-σ). Note que $\Gamma' \wedge (\Delta^1_{<i}.\Delta^1_{>i}) \wedge \cdots \wedge (\Delta^m_{<i}.\Delta^m_{>i}) = \Gamma_{<i}.\Gamma_{\geq i}$. Portanto, por $\rightarrow_e$ tem-se que $((M_1\sigma^i N)\ (M_2\sigma^i N)) : \langle \Gamma \vdash \tau\rangle$. Para a primeira alternativa a demonstração é análoga.

No segundo caso tem-se $\Gamma_{\geq i} = \Gamma' \wedge \Delta^1 \wedge \cdots \wedge \Delta^m$, para $|\Gamma_{\geq i}| > 0$, tal que $(M_1\ M_2) : \langle \Gamma_{<i}. \wedge_{j=1}^m \sigma_j.\Gamma' \vdash \tau\rangle$ e $\forall 1 \leq j \leq m$, $N : \langle \Delta^j \vdash \sigma_j\rangle$. Pelo Lema 7.2.3.4 tem-se que $M_1 : \langle \Gamma_{<i}. \wedge_{j=1}^m \sigma_j.\Gamma' \vdash \omega \rightarrow \tau\rangle$ ou $\Gamma_{<i}. \wedge_{j=1}^m \sigma_j.\Gamma' = \Gamma'' \wedge (\Delta')^1 \wedge \cdots \wedge (\Delta')^l$

tal que $M_1 : \langle \Gamma'' \vdash \wedge_{k=1}^{l} \sigma'_k \to \tau \rangle$ e $\forall 1 \leq k \leq l$, $M_2 : \langle (\Delta')^k \vdash \sigma'_k \rangle$. Assim, para a primeira alternativa referente ao Lema 7.2.3.4 tem-se por ($\wedge$-σ) que $M_1 \sigma^i N$: $\langle (\Gamma_{<i}.\Gamma') \wedge \underline{\omega^{i-1}}.(\Delta^1 \wedge \cdots \wedge \Delta^m) \vdash \omega \to \tau \rangle$ logo, por $\to_e^\omega$, $((M_1 \sigma^i N)\ (M_2 \sigma^i N))$: $\langle (\Gamma_{<i}.\Gamma') \wedge \underline{\omega^{i-1}}.(\Delta^1 \wedge \cdots \wedge \Delta^m) \vdash \tau \rangle$. Note que $(\Gamma_{<i}.\Gamma') \wedge \underline{\omega^{i-1}}.(\Delta^1 \wedge \cdots \wedge \Delta^m) = \Gamma_{<i}.(\Gamma' \wedge \Delta^1 \wedge \cdots \wedge \Delta^m)$. Para a segunda alternativa a demonstração é análoga.

- (σ-destruction): Pelo Lema 7.2.3.1 tem-se que $\underline{n} : \langle \underline{\omega^{n-1}}.\tau.nil \vdash \tau \rangle$. As possíveis tipagens de $\underline{n}\, \sigma^i N$ são analisadas a seguir.

  Se $n < i$ então por (nil-σ) tem-se que $\underline{n}\, \sigma^i N : \langle \underline{\omega^{n-1}}.\tau.nil \vdash \tau \rangle$ e $\underline{n}\, \sigma^i N \to \underline{n}$.

  Se $n = i$ então $\underline{n}\, \sigma^i N \to \varphi_0^i N$. Note que para $\underline{i}\, \sigma^i N$ ser tipável, N deve ter tipo τ. Se $N : \langle nil \vdash \tau \rangle$, então por ($\wedge$-$nil$-$\sigma$) tem-se que $\underline{i}\, \sigma^i N : \langle nil \vdash \tau \rangle$ e, por (nil-φ), $\varphi_0^i N : \langle nil \vdash \tau \rangle$. Se $N : \langle \Gamma \vdash \tau \rangle$, para $|\Gamma| > 0$, então por ($\wedge$-σ) tem-se que $\underline{i}\, \sigma^i N : \langle \underline{\omega^{i-1}}.\Gamma \vdash \tau \rangle$ e, por (ω-φ), $\varphi_0^i N : \langle \underline{\omega^{i-1}}.\Gamma \vdash \tau \rangle$.

  Se $n > i$ então por (ω-σ) tem-se que $\underline{n}\, \sigma^i N : \langle \underline{\omega^{n-2}}.\tau.nil \vdash \tau \rangle$ e $\underline{n}\, \sigma^i N \to \underline{n-1}$. Portanto, por var e varn tem-se que $\underline{n-1} : \langle \underline{\omega^{n-2}}.\tau.nil \vdash \tau \rangle$.

- (φ-λ-transition): demonstração análoga a demonstração de SR para (φ-λ-transition) em λs^{SM}, apresentada no Teorema 7.1.5.

- (φ-app-transition): Seja $\varphi_k^i (M_1\ M_2) : \langle \Gamma \vdash \tau \rangle$.

  Se $|\Gamma| \leq k$ então pelo Lema 7.2.4.1 tem-se $(M_1\ M_2) : \langle \Gamma \vdash \tau \rangle$. Pelo Lema 7.2.3.4 tem-se $M_1 : \langle \Gamma \vdash \omega \to \tau \rangle$ ou $\Gamma = \Gamma' \wedge \Delta^1 \wedge \cdots \wedge \Delta^m$ tal que $M_1 : \langle \Gamma' \vdash \wedge_{j=1}^m \sigma_j \to \tau \rangle$ e $\forall 1 \leq j \leq m$, $M_2 : \langle \Delta^j \vdash \sigma_j \rangle$. Tem-se que $max(|\Gamma'|, |\Delta^1|, \ldots, |\Delta^m|) = |\Gamma| \leq k$. Assim, para a segunda alternativa relativa ao Lema 7.2.3.4 tem-se, por (nil-φ), que $\varphi_k^i M_1 : \langle \Gamma' \vdash \wedge_{j=1}^m \sigma_j \to \tau \rangle$ e $\forall 1 \leq j \leq m$, $\varphi_k^i M_2 : \langle \Delta^j \vdash \sigma_j \rangle$. Portanto, por $\to_e$ tem-se que $((\varphi_k^i M_1)\ (\varphi_k^i M_2)) : \langle \Gamma' \wedge \Delta^1 \wedge \cdots \wedge \Delta^m \vdash \tau \rangle$. A demonstração para a primeira alternativa é similar.

  Se $|\Gamma| > k$ então pelo Lema 7.2.4.1 tem-se que $(M_1\ M_2) : \langle \Gamma_{\leq k}.\Gamma_{\geq k+i} \vdash \tau \rangle$ onde $\Gamma = \Gamma_{\leq k}.\underline{\omega^{i-1}}.\Gamma_{\geq k+i}$. Pelo Lema 7.2.3.4 tem-se que $M_1 : \langle \Gamma_{\leq k}.\Gamma_{\geq k+i} \vdash \omega \to \tau \rangle$ ou $\Gamma_{\leq k}.\Gamma_{\geq k+i} = \Gamma' \wedge \Delta^1 \wedge \cdots \wedge \Delta^m$ tal que $M_1 : \langle \Gamma' \vdash \wedge_{j=1}^m \sigma_j \to \tau \rangle$ e $\forall 1 \leq j \leq m$, $M_2 : \langle \Delta^j \vdash \sigma_j \rangle$. Seja $\Delta' = \Delta^1 \wedge \cdots \wedge \Delta^m$ e suponha s.p.d.g. que $|\Gamma'| \leq k$, logo $\Delta' = \Delta'_{\leq k}.\Gamma_{\geq k+i}$ onde $\Gamma' \wedge \Delta'_{\leq k} = \Gamma_{\leq k}$. Assim, para a segunda alternativa relativa ao Lema 7.2.3.4 tem-se por (nil-φ) que $\varphi^i_k M_1 : \langle \Gamma' \vdash \wedge^m_{j=1}\sigma_j \to \tau\rangle$ e $\forall 1 \leq j \leq m$, se $|\Delta^j| > k$ então $\varphi^i_k M_2 : \langle \Delta^j_{\leq k}.\underline{\omega^{i-1}}.\Delta^j_{>k} \vdash \sigma_j\rangle$ por (ω-φ) e se $|\Delta^j| \leq k$ então $\varphi^i_k M_2 : \langle \Delta^j \vdash \sigma_j\rangle$ por (nil-φ). Note que $\Delta^1_{>k} \wedge \cdots \wedge \Delta^m_{>k} = \Delta'_{>k} = \Gamma_{\geq k+i}$. Portanto, por $\to_e$, $(\varphi^i_k M_1\ \varphi^i_k M_2) : \langle \Gamma' \wedge (\Delta'_{\leq k}.\underline{\omega^{i-1}}.\Delta'_{>k}) \vdash \tau\rangle$, onde $\Gamma' \wedge (\Delta'_{\leq k}.\underline{\omega^{i-1}}.\Delta'_{>k}) = (\Gamma' \wedge \Delta'_{\leq k}).\underline{\omega^{i-1}}.\Gamma_{\geq k+i}$. Para a primeira alternativa a demonstração é análoga.

- (φ-destruction): As possíveis tipagens de $\varphi^i_k\,\underline{n}$ são analisadas a seguir, como o feito para (σ-destruction) acima. Seja $\underline{n} : \langle \underline{\omega^{n-1}}.\tau.nil \vdash \tau\rangle$.

  Se $n \leq k$ então por (nil-φ) tem-se que $\varphi^i_k\,\underline{n} : \langle \underline{\omega^{n-1}}.\tau.nil \vdash \tau\rangle$ e $\varphi^i_k\,\underline{n} \to \underline{n}$.

  Se $n > k$ então por (ω-φ) tem-se que $\varphi^i_k\,\underline{n} : \langle \underline{\omega^{n+i-2}}.\tau.nil \vdash \tau\rangle$ e $\varphi^i_k\,\underline{n} \to \underline{n{+}i{-}1}$. Portanto, por var e varn tem-se que $\underline{n{+}i{-}1} : \langle \underline{\omega^{n+i-2}}.\tau.nil \vdash \tau\rangle$.

- (σ-σ-transition): Seja $(M_1\sigma^i M_2)\,\sigma^j M_3 : \langle \Gamma \vdash \tau\rangle$, para $i \leq j$.

  Se $\Gamma = nil$, então pelo Lema 7.2.4.2 tem-se dois casos.

  No primeiro caso tem-se que $(M_1\sigma^i M_2) : \langle nil \vdash \tau\rangle$. Logo, pelo Lema 7.2.4.2 tem-se dois subcasos.

  No primeiro subcaso tem-se que $M_1 : \langle nil \vdash \tau\rangle$. Portanto, por ($nil$-$\sigma$) tem-se que $M_1\sigma^{j+1}M_3 : \langle nil \vdash \tau\rangle$ e $(M_1\sigma^{j+1}M_3)\,\sigma^i(M_2\,\sigma^{j-i+1}M_3) : \langle nil \vdash \tau\rangle$.

  No segundo subcaso tem-se que $M_1 : \langle \underline{\omega^{i-1}}.\wedge^m_{k=1}\sigma_k.nil \vdash \tau\rangle$ onde $\forall 1 \leq k \leq m$, $M_2 : \langle nil \vdash \sigma_k\rangle$. Tem-se que $i \leq j$, logo $i < j{+}1$. Assim, por (nil-σ) tem-se que $M_1\sigma^{j+1}M_3 : \langle \underline{\omega^{i-1}}.\wedge^m_{k=1}\sigma_k.nil \vdash \tau\rangle$ e, como $j{-}i{+}1 > 0$, $\forall 1 \leq k \leq m$, $M_2\,\sigma^{j-i+1}M_3 : \langle nil \vdash \sigma_k\rangle$. Portanto, por ($\wedge$-$nil$-$\sigma$) tem-se que $(M_1\sigma^{j+1}M_3)\,\sigma^i(M_2\,\sigma^{j-i+1}M_3) : \langle nil \vdash \tau\rangle$.

  No segundo caso tem-se que $(M_1\sigma^i M_2) : \langle \underline{\omega^{j-1}}.\wedge^m_{k=1}\sigma_k.nil \vdash \tau\rangle$ onde $\forall 1 \leq k \leq m$, $M_3 : \langle nil \vdash \sigma_k\rangle$. Tem-se que $i \leq j = |\underline{\omega^{j-1}}.\wedge^m_{k=1}\sigma_k.nil|$, logo pelo Lema 7.2.4.4 tem-se dois subcasos.

  No primeiro subcaso tem-se que $M_1 : \langle \underline{\omega^{j}}.\wedge^m_{k=1}\sigma_k.nil \vdash \tau\rangle$. Assim, por ($\wedge$-$nil$-$\sigma$) tem-se que $M_1\sigma^{j+1}M_3 : \langle nil \vdash \tau\rangle$ logo, por (nil-σ), $(M_1\sigma^{j+1}M_3)\,\sigma^i(M_2\,\sigma^{j-i+1}M_3) : \langle nil \vdash \tau\rangle$.

  No segundo subcaso tem-se que $(\underline{\omega^{j-1}}.\wedge^m_{k=1}\sigma_k.nil)_{\geq i} = \Gamma' \wedge \Delta^1 \wedge \cdots \wedge \Delta^{m'}$ tal que $M_1 : \langle (\underline{\omega^{j-1}}.\wedge^m_{k=1}\sigma_k.nil)_{<i}.\wedge^{m'}_{l=1}\sigma'_l.\Gamma' \vdash \tau\rangle$ e $\forall 1 \leq l \leq m'$, $M_2 : \langle \Delta^l \vdash \sigma'_l\rangle$. Observe que $(\underline{\omega^{j-1}}.\wedge^m_{k=1}\sigma_k.nil)_{<i} = \underline{\omega^{i-1}}$ e suponha s.p.d.g. que $\Gamma' = nil$ logo $\forall 1 \leq l \leq m'$, $\Delta^l =$

nil ou $|\Delta^l| = j-(i-1)$. Assim, por (nil-σ) tem-se $M_1\sigma^{j+1}M_3 : \langle \omega^{\underline{i-1}}.\wedge_{l=1}^{m'} \sigma'_l.nil \vdash \tau\rangle$ e $\forall 1 \leq l \leq m'$, se $|\Delta^l| = 0$ então $M_2\,\sigma^{j-i+1}M_3 : \langle nil \vdash \sigma'_l\rangle$ por (nil-σ) e se $|\Delta^l| > 0$ então $M_2\,\sigma^{j-i+1}M_3 : \langle nil \vdash \sigma'_l\rangle$ por ($\wedge$-nil-σ). Portanto, por ($\wedge$-nil-σ), $(M_1\sigma^{j+1}M_3)\,\sigma^i(M_2\,\sigma^{j-i+1}M_3) : \langle nil \vdash \tau\rangle$.

Se $0 < |\Gamma| < j$, então pelo Lema 7.2.4.3 tem-se dois casos.

No primeiro caso tem-se que $(M_1\sigma^i M_2) : \langle \Gamma \vdash \tau\rangle$. Note que $|\Gamma| < i$ ou $|\Gamma| \geq i$.

Se $|\Gamma| < i$ então pelo Lema 7.2.4.3 tem-se dois subcasos.

No primeiro subcaso tem-se que $M_1 : \langle \Gamma \vdash \tau\rangle$. Observe que $|\Gamma| < i \leq j < j+1$. Portanto, por ($nil$-$\sigma$) tem-se que $M_1\sigma^{j+1}M_3 : \langle \Gamma \vdash \tau\rangle$ e $(M_1\sigma^{j+1}M_3)\,\sigma^i(M_2\,\sigma^{j-i+1}M_3) : \langle \Gamma \vdash \tau\rangle$.

No segundo subcaso tem-se que $M_1 : \langle \Gamma' \vdash \tau\rangle$ onde $\Gamma' = \Gamma.\omega^{\underline{n}}.\wedge_{k=1}^{m} \sigma_k.nil$ para $n \geq 0$ tal que $|\Gamma'| = i$ e $\forall 1 \leq k \leq m$, $M_2 : \langle nil \vdash \sigma_k\rangle$. Tem-se que $i \leq j < j+1$ logo por (nil-σ) tem-se que $M_1\sigma^{j+1}M_3 : \langle \Gamma' \vdash \tau\rangle$ e $\forall 1 \leq k \leq m$, $M_2\,\sigma^{j-i+1}M_3 : \langle nil \vdash \sigma_k\rangle$. Portanto, por ($\wedge$-$\omega$-$\sigma$), $(M_1\sigma^{j+1}M_3)\,\sigma^i(M_2\,\sigma^{j-i+1}M_3) : \langle \Gamma \vdash \tau\rangle$.

Se $|\Gamma| \geq i$ então pelo Lema 7.2.4.4 tem-se dois subcasos.

No primeiro subcaso tem-se que $M_1 : \langle \Gamma_{<i}.\omega.\Gamma_{\geq i} \vdash \tau\rangle$. Observe que $|\Gamma_{<i}.\omega.\Gamma_{\geq i}| = |\Gamma|+1 < j+1$. Assim, por ($nil$-$\sigma$) tem-se que $M_1\sigma^{j+1}M_3 : \langle \Gamma_{<i}.\omega.\Gamma_{\geq i} \vdash \tau\rangle$ logo, por (ω-σ), $(M_1\sigma^{j+1}M_3)\,\sigma^i(M_2\,\sigma^{j-i+1}M_3) : \langle \Gamma \vdash \tau\rangle$.

No segundo subcaso tem-se que $\Gamma_{\geq i} = \Gamma' \wedge \Delta^1 \wedge \cdots \wedge \Delta^m$ para $|\Gamma_{\geq i}| > 0$ tal que $M_1 : \langle \Gamma_{<i}.\wedge_{k=1}^{m} \sigma_k.\Gamma' \vdash \tau\rangle$ e $\forall 1 \leq k \leq m$, $M_2 : \langle \Delta^k \vdash \sigma_k\rangle$. Note que $|\Gamma_{<i}.\wedge_{k=1}^{m} \sigma_k.\Gamma'| \leq |\Gamma|+1 < j+1$ e que $\forall 1 \leq k \leq m$, $|\Delta^k| \leq |\Gamma|-(i-1) < j-(i-1)$. Assim, por ($nil$-$\sigma$) tem-se $M_1\sigma^{j+1}M_3 : \langle \Gamma_{<i}.\wedge_{k=1}^{m} \sigma_k.\Gamma' \vdash \tau\rangle$ e $\forall 1 \leq k \leq m$, $M_2\,\sigma^{j-i+1}M_3 : \langle \Delta^k \vdash \sigma_k\rangle$. Portanto, por ($\wedge$-$\sigma$), $(M_1\sigma^{j+1}M_3)\,\sigma^i(M_2\,\sigma^{j-i+1}M_3) : \langle (\Gamma_{<i}.\Gamma') \wedge \omega^{\underline{i-1}}.(\Delta^1 \wedge \cdots \wedge \Delta^m) \vdash \tau\rangle$ onde $(\Gamma_{<i}.\Gamma') \wedge \omega^{\underline{i-1}}.(\Delta^1 \wedge \cdots \wedge \Delta^m) = \Gamma_{<i}.(\Gamma' \wedge \Delta^1 \wedge \cdots \wedge \Delta^m)$.

No segundo caso tem-se que $(M_1\sigma^i M_2) : \langle \Gamma' \vdash \tau\rangle$, onde $\Gamma' = \Gamma.\omega^{\underline{n}}.\wedge_{k=1}^{m} \sigma_k.nil$ para $n \geq 0$ tal que $|\Gamma'| = j$ e $\forall 1 \leq k \leq m$, $M_3 : \langle nil \vdash \sigma_k\rangle$. Note que $|\Gamma'| = j \geq i$. Logo, pelo Lema 7.2.4.4 tem-se dois subcasos.

No primeiro subcaso tem-se que $M_1 : \langle \Gamma'_{<i}.\omega.\Gamma'_{\geq i} \vdash \tau\rangle$. Suponha s.p.d.g. que $|\Gamma| < i$ logo $\Gamma'_{<i}.\omega.\Gamma'_{\geq i} = \Gamma.\omega^{\underline{n+1}}.\wedge_{k=1}^{m} \sigma_k.nil$. Assim, por ($\wedge$-$\omega$-$\sigma$), $M_1\sigma^{j+1}M_3 : \langle \Gamma \vdash \tau\rangle$. Portanto, por ($nil$-$\sigma$), $(M_1\sigma^{j+1}M_3)\,\sigma^i(M_2\,\sigma^{j-i+1}M_3) : \langle \Gamma \vdash \tau\rangle$.

No segundo subcaso tem-se que $\Gamma'_{\geq i}=\Gamma''\wedge(\Delta')^1\wedge\cdots\wedge(\Delta')^{m'}$ para $|\Gamma'_{\geq i}|>0$ tal que $M_1:\langle\Gamma'_{<i}.\wedge_{l=1}^{m'}\sigma'_l.\Gamma''\vdash\tau\rangle$ e $\forall 1\leq l\leq m'$, $M_2:\langle(\Delta')^l\vdash\sigma'_l\rangle$. Suponha s.p.d.g. que $|\Gamma|\geq i$ e $\Gamma''=\Gamma_{\geq i}$ logo $\Gamma'_{<i}.\wedge_{l=1}^{m'}\sigma'_l.\Gamma''=\Gamma_{<i}.\wedge_{l=1}^{m'}\sigma'_l.\Gamma_{\geq i}$ e $\forall 1\leq l\leq m'$, $(\Delta')^l=nil$ ou $(\Delta')^l=\omega^{\underline{j-i+1}}.u_l.nil$ onde $\omega\neq u_l\sqsubseteq\wedge_{k=1}^m\sigma_k$ e $u_1\wedge\cdots\wedge u_{m'}=\wedge_{k=1}^m\sigma_k$. Assim, por $(nil\text{-}\sigma)$ tem-se que $M_1\sigma^{j+1}M_3:\langle\Gamma_{<i}.\wedge_{l=1}^{m'}\sigma'_l.\Gamma_{\geq i}\vdash\tau\rangle$ e $\forall 1\leq l\leq m'$, se $(\Delta')^l=nil$ então $M_2\,\sigma^{j-i+1}M_3:\langle nil\vdash\sigma'_l\rangle$ por $(nil\text{-}\sigma)$ e se $|(\Delta')^l|=j-i+1$ então $M_2\,\sigma^{j-i+1}M_3:\langle nil\vdash\sigma'_l\rangle$ por $(\wedge\text{-}nil\text{-}\sigma)$. Note que $(\Gamma_{<i}.\Gamma_{\geq i})\wedge\omega^{\underline{i-1}}.(nil\wedge\cdots\wedge nil)=\Gamma_{<i}.(\Gamma_{\geq i}\wedge nil)=\Gamma$. Portanto, por $(\wedge\text{-}\sigma)$, $(M_1\sigma^{j+1}M_3)\,\sigma^i(M_2\,\sigma^{j-i+1}M_3):\langle\Gamma\vdash\tau\rangle$.

Se $|\Gamma|\geq j$, então pelo Lema 7.2.4.4 tem-se dois casos.

No primeiro caso tem-se que $(M_1\sigma^iM_2):\langle\Gamma_{<j}.\omega.\Gamma_{\geq j}\vdash\tau\rangle$. Note que $|\Gamma_{<j}.\omega.\Gamma_{\geq j}|=|\Gamma|+1\geq j+1>j\geq i$. Logo, pelo Lema 7.2.4.4 tem-se dois subcasos.

No primeiro subcaso tem-se que $M_1:\langle(\Gamma_{<j}.\omega)_{<i}.\omega.(\Gamma_{<j}.\omega)_{\geq i}.\Gamma_{\geq j}\vdash\tau\rangle$. Observe que $(\Gamma_{<j}.\omega)_{<i}=\Gamma_{<i}$ e $(\Gamma_{<j}.\omega)_{\geq i}=(\Gamma_{<j})_{\geq i}.\omega$. Assim, por $(\omega\text{-}\sigma)$ tem-se que $M_1\sigma^{j+1}M_3:\langle\Gamma_{<i}.\omega.(\Gamma_{<j})_{\geq i}.\Gamma_{\geq j}\vdash\tau\rangle$ onde $(\Gamma_{<j})_{\geq i}.\Gamma_{\geq j}=\Gamma_{\geq i}$. Portanto, por $(\omega\text{-}\sigma)$ tem-se que $(M_1\sigma^{j+1}M_3)\,\sigma^i(M_2\,\sigma^{j-i+1}M_3):\langle\Gamma_{<i}.\Gamma_{\geq i}\vdash\tau\rangle$.

No segundo subcaso tem-se que $(\Gamma_{<j})_{\geq i}.\omega.\Gamma_{\geq j}=\Gamma'\wedge\Delta^1\wedge\cdots\wedge\Delta^m$ tal que $M_1:\langle\Gamma_{<i}.\wedge_{k=1}^m\sigma_k.\Gamma'\vdash\tau\rangle$ e $\forall 1\leq k\leq m$, $M_2:\langle\Delta^k\vdash\sigma_k\rangle$. Suponha s.p.d.g. que $\Gamma'=\Gamma''.\omega.\Gamma_{\geq j}$ logo $\Gamma''\wedge\Delta^1\wedge\cdots\wedge\Delta^m=(\Gamma_{<j})_{\geq i}$. Assim, por $(\omega\text{-}\sigma)$ tem-se que $M_1\sigma^{j+1}M_3:\langle\Gamma_{<i}.\wedge_{k=1}^m\sigma_k.\Gamma''.\Gamma_{\geq j}\vdash\tau\rangle$ e por $(nil\text{-}\sigma)$ tem-se $\forall 1\leq k\leq m$, $M_2\,\sigma^{j-i+1}M_3:\langle\Delta^k\vdash\sigma_k\rangle$. Seja $\Delta'=\Delta^1\wedge\cdots\wedge\Delta^m$. Portanto, por $(\wedge\text{-}\sigma)$, $(M_1\sigma^{j+1}M_3)\,\sigma^i(M_2\,\sigma^{j-i+1}M_3):\langle(\Gamma_{<i}.\Gamma''.\Gamma_{\geq j})\wedge\omega^{\underline{i-1}}.\Delta'\vdash\tau\rangle$, onde $(\Gamma_{<i}.\Gamma''.\Gamma_{\geq j})\wedge\omega^{\underline{i-1}}.\Delta'=\Gamma_{<i}.((\Gamma''.\Gamma_{\geq j})\wedge\Delta')=\Gamma_{<i}.(\Gamma''\wedge\Delta').\Gamma_{\geq j}$.

No segundo caso tem-se que $\Gamma_{\geq j}=\Gamma'\wedge\Delta^1\wedge\cdots\wedge\Delta^m$ para $|\Gamma_{\geq j}|>0$ tal que $(M_1\sigma^iM_2):\langle\Gamma_{<j}.\wedge_{k=1}^m\sigma_k.\Gamma'\vdash\tau\rangle$ e $\forall 1\leq k\leq m$, $M_3:\langle\Delta^k\vdash\sigma_k\rangle$. Logo, pelo Lema 7.2.4.4 tem-se dois subcasos, que são similares aos subcasos do primeiro caso tratado logo acima.

- $(\sigma\text{-}\varphi\text{-transition 1})$: Seja $(\varphi_k^iM)\,\sigma^jN:\langle\Gamma\vdash\tau\rangle$, para $k<j<k+i$.

  Se $\Gamma=nil$, então pelo Lema 7.2.4.2 tem-se que $\varphi_k^iM:\langle nil\vdash\tau\rangle$. Logo, pelo Lema 7.2.4.1 tem-se que $M:\langle nil\vdash\tau\rangle$. Portanto, por $(nil\text{-}\varphi)$ tem-se que $\varphi_k^{i-1}M:\langle nil\vdash\tau\rangle$. Observe que a segunda possibilidade pelo Lema 7.2.4.2 é tal que $\varphi_k^iM:$

$\langle \omega^{\underline{j-1}}.\wedge_{l=1}^{m} \sigma_l.nil \vdash \tau\rangle$ e $\forall 1 \leq l \leq m$, $N:\langle nil \vdash \sigma_l\rangle$. Porém, pelo Lema 7.2.4.1 tem-se que $k+(i-1) \leq j-1$ logo $k+i \leq j$ e por hipótese tem-se $j < k+i$.

Se $0 < |\Gamma| < j$, então pelo Lema 7.2.4.3 tem-se que $\varphi_k^i M : \langle \Gamma \vdash \tau\rangle$. Logo, pelo Lema 7.2.4.1 tem-se que $|\Gamma| \leq k$ logo $M:\langle \Gamma \vdash \tau\rangle$. Portanto, por ($nil$-$\varphi$), $\varphi_k^{i-1} M:\langle \Gamma \vdash \tau\rangle$. Note que se $|\Gamma| > k$ então, pelo Lema 7.2.4.1, $k+i \leq |\Gamma| < j < k+i$. Observe que a segunda possibilidade pelo Lema 7.2.4.3 é tal que $\varphi_k^i M : \langle \Gamma' \vdash \tau\rangle$, onde $\Gamma' = \Gamma.\omega^{\underline{n}}.\wedge_{l=1}^{m} \sigma_l.nil$ para $n \geq 0$ tal que $|\Gamma'| = j$ e $\forall 1 \leq l \leq m$, $N:\langle nil \vdash \sigma_l\rangle$. Porém, pelo Lema 7.2.4.1 tem-se que $k+i \leq |\Gamma'| = j$ e por hipótese tem-se $j < k+i$.

Se $|\Gamma| \geq j$, então pelo Lema 7.2.4.4 tem-se que $\varphi_k^i M : \langle \Gamma_{<j}.\omega.\Gamma_{\geq j} \vdash \tau\rangle$.Logo, pelo Lema 7.2.4.1 e pela hipótese de que $k < j < k+i$, tem-se que $M:\langle \Gamma_{\leq k}.\Gamma_{\geq k+i} \vdash \tau\rangle$ onde $(\Gamma_{<j})_{>k}.\omega.(\Gamma_{<k+i})_{\geq j} = \omega^{\underline{i-1}}$. Portanto, $(\Gamma_{<j})_{>k}.(\Gamma_{<k+i})_{\geq j} = (\Gamma_{<k+i})_{>k} = \omega^{\underline{i-2}}$. Assim, por ($\omega$-$\varphi$) tem-se que $M\varphi_k^{i-1} M:\langle \Gamma_{\leq k}.\omega^{\underline{i-2}}.\Gamma_{\geq k+i} \vdash \tau\rangle$, onde $\Gamma_{\leq k}.\omega^{\underline{i-2}}.\Gamma_{\geq k+i} = \Gamma_{\leq k}.(\Gamma_{<k+i})_{>k}.\Gamma_{\geq k+i} = \Gamma$. Note que a segunda possibilidade pelo Lema 7.2.4.4 é tal que $\Gamma_{\geq j} = \Gamma' \wedge \Delta^1 \wedge \cdots \wedge \Delta^m$ para $|\Gamma_{\geq j}| > 0$ tal que $\varphi_k^i M : \langle \Gamma_{<j}.\wedge_{l=1}^{m} \sigma_l.\Gamma' \vdash \tau\rangle$ e $\forall 1 \leq l \leq m$, $N : \langle \Delta^l \vdash \sigma_l\rangle$. Porém, pelo Lema 7.2.4.1 e pela hipótese de que $k < j < k+i$, tem-se que $(\Gamma_{<j}.\wedge_{l=1}^{m} \sigma_l.\Gamma')_j = \omega$.

- (σ-φ-transition 2): Seja $(\varphi_k^i M)\,\sigma^j N:\langle \Gamma \vdash \tau\rangle$, para $k+i \leq j$.

  Se $\Gamma = nil$, então pelo Lema 7.2.4.2 tem-se dois casos.

  No primeiro caso tem-se que $\varphi_k^i M:\langle nil \vdash \tau\rangle$. Logo, pelo Lema 7.2.4.2 tem-se que $M:\langle nil \vdash \tau\rangle$. Portanto, por ($nil$-$\sigma$) tem-se que $M\sigma^{j-i+1}N:\langle nil \vdash \tau\rangle$ e, por (nil-φ), que $\varphi_k^i(M\sigma^{j-i+1}N):\langle nil \vdash \tau\rangle$.

  No segundo caso tem-se que $\varphi_k^i M : \langle \omega^{\underline{j-1}}.\wedge_{l=1}^{m} \sigma_l.nil \vdash \tau\rangle$ e $\forall 1 \leq l \leq m$, $N : \langle nil \vdash \sigma_l\rangle$. Logo, pelo Lema 7.2.4.1 e pela hipótese de que $k+i \leq j$ tem-se que $M : \langle \omega^{\underline{(j-1)-(i-1)}}.\wedge_{l=1}^{m} \sigma_l.nil \vdash \tau\rangle$. Portanto, por ($\wedge$-$nil$-$\sigma$) tem-se que $M\sigma^{j-i+1}N : \langle nil \vdash \tau\rangle$ e por (nil-φ) tem-se que $\varphi_k^i(M\sigma^{j-i+1}N):\langle nil \vdash \tau\rangle$.

  Se $0 < |\Gamma| < j$, então pelo Lema 7.2.4.3 tem-se dois casos.

  No primeiro caso tem-se que $\varphi_k^i M:\langle \Gamma \vdash \tau\rangle$. Note que $|\Gamma| \leq k$ ou $|\Gamma| > k$.

  Se $|\Gamma| \leq k$, então pelo Lema 7.2.4.1 tem-se que $M:\langle \Gamma \vdash \tau\rangle$. Tem-se que $k+i \leq j$ logo $|\Gamma| \leq k \leq j-i < j-i+1$. Portanto, por ($nil$-$\sigma$) tem-se que $M\sigma^{j-i+1}N:\langle \Gamma \vdash \tau\rangle$ e por (nil-φ) tem-se que $\varphi_k^i(M\sigma^{j-i+1}N):\langle \Gamma \vdash \tau\rangle$.

Se $|\Gamma| > k$, então pelo Lema 7.2.4.1 tem-se que $M : \langle \Gamma_{\leq k}.\Gamma_{\geq k+i} \vdash \tau \rangle$, onde $\Gamma = \Gamma_{\leq k}.\underline{\omega^{i-1}}.\Gamma_{\geq k+i}$. Tem-se que $|\Gamma_{\leq k}.\Gamma_{\geq k+i}| = |\Gamma| - (i-1) < j - (i-1)$. Portanto, por ($nil$-$\sigma$) tem-se que $M\sigma^{j-i+1}N : \langle \Gamma_{\leq k}.\Gamma_{\geq k+i} \vdash \tau \rangle$ e por (ω-φ) tem-se que $\varphi^i_k(M\sigma^{j-i+1}N) : \langle \Gamma_{\leq k}.\underline{\omega^{i-1}}.\Gamma_{\geq k+i} \vdash \tau \rangle$.

No segundo caso tem-se que $\varphi^i_k M : \langle \Gamma' \vdash \tau \rangle$, onde $\Gamma' = \Gamma.\underline{\omega^{n}}.\wedge^m_{l=1} \sigma_l.nil$ para $n \geq 0$ tal que $|\Gamma'| = j$ e $\forall 1 \leq l \leq m$, $N : \langle nil \vdash \sigma_l \rangle$. Note que pelo Lema 7.2.2 $\Gamma_{|\Gamma|} \neq \omega$, logo pelo Lema 7.2.4.1 tem-se que $k + i \leq |\Gamma|$ ou $|\Gamma| < k$. Suponha s.p.d.g. que $k + i \leq |\Gamma|$. Assim, $M : \langle \Gamma_{\leq k}.\Gamma_{\geq k+i}.\underline{\omega^{n}}.\wedge^m_{l=1} \sigma_l.nil \vdash \tau \rangle$, onde $\Gamma = \Gamma_{\leq k}.\underline{\omega^{i-1}}.\Gamma_{\geq k+i}$. Portanto, por ($\wedge$-$\omega$-$\sigma$) tem-se que $M\sigma^{j-i+1}N : \langle \Gamma_{\leq k}.\Gamma_{\geq k+i} \vdash \tau \rangle$ e, por (ω-φ), $\varphi^i_k(M\sigma^{j-i+1}N) : \langle \Gamma_{\leq k}.\underline{\omega^{i-1}}.\Gamma_{\geq k+i} \vdash \tau \rangle$.

Se $|\Gamma| \geq j$, então pelo Lema 7.2.4.4 tem-se dois casos.

No primeiro caso tem-se que $\varphi^i_k M : \langle \Gamma_{<j}.\omega.\Gamma_{\geq j} \vdash \tau \rangle$. Tem-se que $k+i \leq j$ logo, pelo Lema 7.2.4.1, $M : \langle \Gamma_{\leq k}.(\Gamma_{<j})_{\geq k+i}.\omega.\Gamma_{\geq j} \vdash \tau \rangle$ onde $\Gamma_{<j} = \Gamma_{\leq k}.\underline{\omega^{i-1}}.(\Gamma_{<j})_{\geq k+i}$. Portanto, por ($\omega$-$\sigma$) tem-se que $M\sigma^{j-i+1}N : \langle \Gamma_{\leq k}.(\Gamma_{<j})_{\geq k+i}.\Gamma_{\geq j} \vdash \tau \rangle$ e por (ω-φ) tem-se $\varphi^i_k(M\sigma^{j-i+1}N) : \langle \Gamma_{\leq k}.\underline{\omega^{i-1}}.(\Gamma_{<j})_{\geq k+i}.\Gamma_{\geq j} \vdash \tau \rangle$, onde $\Gamma_{\leq k}.\underline{\omega^{i-1}}.(\Gamma_{<j})_{\geq k+i}.\Gamma_{\geq j} = \Gamma$.

No segundo caso tem-se que $\Gamma_{\geq j} = \Gamma' \wedge \Delta^1 \wedge \cdots \wedge \Delta^m$ para $|\Gamma_{\geq j}| > 0$ tal que $\varphi^i_k M : \langle \Gamma_{<j}.\wedge^m_{l=1} \sigma_l.\Gamma' \vdash \tau \rangle$ e $\forall 1 \leq l \leq m$, $N : \langle \Delta^l \vdash \sigma_l \rangle$. A demonstração de $\varphi^i_k(M\sigma^{j-i+1}N) : \langle \Gamma \vdash \tau \rangle$ é similar ao primeiro caso apresentado logo acima.

- (φ-σ-transition): Seja $\varphi^i_k(M\sigma^j N) : \langle \Gamma \vdash \tau \rangle$, para $j \leq k+1$.

  Se $|\Gamma| \leq k$, então pelo Lema 7.2.4.1 tem-se que $M\sigma^j N : \langle \Gamma \vdash \tau \rangle$.

  Se $\Gamma = nil$ então pelo Lema 7.2.4.2 tem-se dois subcasos.

  No primeiro subcaso tem-se que $M : \langle nil \vdash \tau \rangle$. Portanto, por ($nil$-$\varphi$) tem-se $\varphi^i_{k+1} M : \langle nil \vdash \tau \rangle$ e, por (nil-σ), $(\varphi^i_{k+1} M)\, \sigma^j (\varphi^i_{k+1-j} N) : \langle nil \vdash \tau \rangle$.

  No segundo subcaso tem-se que $M : \langle \underline{\omega^{j-1}}.\wedge^m_{l=1} \sigma_l.nil \vdash \tau \rangle$ onde $\forall 1 \leq l \leq m$, $N : \langle nil \vdash \sigma_l \rangle$. Assim, por ($nil$-$\varphi$) tem-se $\varphi^i_{k+1} M : \langle \underline{\omega^{j-1}}.\wedge^m_{l=1} \sigma_l.nil \vdash \tau \rangle$ e $\forall 1 \leq l \leq m$, $\varphi^i_{k+1-j} N : \langle nil \vdash \sigma_l \rangle$. Portanto, por ($\wedge$-$nil$-$\sigma$), $(\varphi^i_{k+1} M)\, \sigma^j (\varphi^i_{k+1-j} N) : \langle nil \vdash \tau \rangle$.

  Se $0 < |\Gamma| < j$ então pelo Lema 7.2.4.3 tem-se dois subcasos.

  No primeiro subcaso tem-se que $M : \langle \Gamma \vdash \tau \rangle$. Note que $|\Gamma| \leq k < k+1$ logo, por (nil-φ), $\varphi^i_{k+1} M : \langle \Gamma \vdash \tau \rangle$. Portanto, por ($nil$-$\sigma$), $(\varphi^i_{k+1} M)\, \sigma^j (\varphi^i_{k+1-j} N) : \langle \Gamma \vdash \tau \rangle$.

No segundo subcaso tem-se que $M:\langle\Gamma'\vdash\tau\rangle$ onde $\Gamma'=\Gamma.\omega^{\underline{n}}.\wedge_{l=1}^{m}\sigma_l.nil$ para $n\geq 0$ tal que $|\Gamma'|=j$ e $\forall 1\leq l\leq m$, $N:\langle nil\vdash\sigma_l\rangle$. Note que $|\Gamma'|=j\leq k{+}1$. Assim, por (nil-φ) tem-se $\varphi^i_{k+1}M:\langle\Gamma'\vdash\tau\rangle$ e $\forall 1\leq l\leq m$, $\varphi^i_{k+1-j}N:\langle nil\vdash\sigma_l\rangle$. Portanto, por ($\wedge$-$\omega$-$\sigma$), $(\varphi^i_{k+1}M)\,\sigma^j(\varphi^i_{k+1-j}N):\langle\Gamma\vdash\tau\rangle$.

Se $|\Gamma|\geq j$ então pelo Lema 7.2.4.4 tem-se dois subcasos.

No primeiro caso tem-se que $M:\langle\Gamma_{<j}.\omega.\Gamma_{\geq j}\vdash\tau\rangle$. Note que $|\Gamma_{<j}.\omega.\Gamma_{\geq j}|=|\Gamma|{+}1\leq k{+}1$. Assim, por ($nil$-$\varphi$) tem-se $\varphi^i_{k+1}M:\langle\Gamma_{<j}.\omega.\Gamma_{\geq j}\vdash\tau\rangle$. Portanto, por ($\omega$-$\sigma$), $(\varphi^i_{k+1}M)\,\sigma^j(\varphi^i_{k+1-j}N):\langle\Gamma_{<j}.\Gamma_{\geq j}\vdash\tau\rangle$.

No segundo caso tem-se que $\Gamma_{\geq j}=\Gamma'\wedge\Delta^1\wedge\cdots\wedge\Delta^m$ para $|\Gamma_{\geq j}|>0$ tal que $M:\langle\Gamma_{<j}.\wedge_{l=1}^{m}\sigma_l.\Gamma'\vdash\tau\rangle$ e $\forall 1\leq l\leq m$, $N:\langle\Delta^l\vdash\sigma_l\rangle$. Observe que $\forall 1\leq l\leq m$, $|\Delta^l|\leq|\Gamma|-(j{-}1)\leq k-(j{-}1)=k{+}1-j$. Assim, por ($nil$-$\varphi$) tem-se $\varphi^i_{k+1}M:\langle\Gamma_{<j}.\wedge_{l=1}^{m}\sigma_l.\Gamma'\vdash\tau\rangle$ e $\forall 1\leq l\leq m$, $\varphi^i_{k+1-j}N:\langle\Delta^l\vdash\sigma_l\rangle$. Portanto, por ($\wedge$-$\sigma$), $(\varphi^i_{k+1}M)\,\sigma^j(\varphi^i_{k+1-j}N):\langle(\Gamma_{<j}.\Gamma')\wedge\omega^{\underline{j-1}}.(\Delta^1\wedge\cdots\wedge\Delta^m)\vdash\tau\rangle$, onde $(\Gamma_{<j}.\Gamma')\wedge\omega^{\underline{j-1}}.(\Delta^1\wedge\cdots\wedge\Delta^m)=\Gamma_{<j}.(\Gamma'\wedge\Delta^1\wedge\cdots\wedge\Delta^m)$.

Se $|\Gamma|>k$, então pelo Lema 7.2.4.1 tem-se que $M\sigma^jN:\langle\Gamma_{\leq k}.\Gamma_{\geq k+i}\vdash\tau\rangle$, onde $\Gamma=\Gamma_{\leq k}.\omega^{\underline{i-1}}.\Gamma_{\geq k+i}$. Note que $|\Gamma_{\leq k}.\Gamma_{\geq k+i}|\geq k{+}1\geq j$ logo, pelo Lema 7.2.4.4, tem-se dois casos.

No primeiro caso, $M:\langle(\Gamma_{\leq k}.\Gamma_{\geq k+i})_{<j}.\omega.(\Gamma_{\leq k}.\Gamma_{\geq k+i})_{\geq j}\vdash\tau\rangle$, onde $(\Gamma_{\leq k}.\Gamma_{\geq k+i})_{<j}=\Gamma_{<j}$ e $(\Gamma_{\leq k}.\Gamma_{\geq k+i})_{\geq j}=(\Gamma_{\leq k})_{\geq j}.\Gamma_{\geq k+i}$. Note que $|\Gamma_{<j}.\omega.(\Gamma_{\leq k})_{\geq j}.|=k{+}1$ logo, por (ω-φ), $\varphi^i_{k+1}M:\langle\Gamma_{<j}.\omega.(\Gamma_{\leq k})_{\geq j}.\omega^{\underline{i-1}}.\Gamma_{\geq k+i}\vdash\tau\rangle$. Portanto, por ($\omega$-$\sigma$) tem-se que $(\varphi^i_{k+1}M)\,\sigma^j(\varphi^i_{k+1-j}N):\langle\Gamma_{<j}.(\Gamma_{\leq k})_{\geq j}.\omega^{\underline{i-1}}.\Gamma_{\geq k+i}\vdash\tau\rangle$ onde $\Gamma_{<j}.(\Gamma_{\leq k})_{\geq j}=\Gamma_{\leq k}$.

No segundo caso tem-se que $(\Gamma_{\leq k}.\Gamma_{\geq k+i})_{\geq j}=(\Gamma_{\leq k})_{\geq j}.\Gamma_{\geq k+i}=\Gamma'\wedge\Delta^1\wedge\cdots\wedge\Delta^m$ para $|(\Gamma_{\leq k}.\Gamma_{\geq k+i})_{\geq j}|>0$ tal que $M:\langle\Gamma_{<j}.\wedge_{l=1}^{m}\sigma_l.\Gamma'\vdash\tau\rangle$ e $\forall 1\leq l\leq m$, $N:\langle\Delta^l\vdash\sigma_l\rangle$. Suponha s.p.d.g. que $\Gamma'=nil$ logo $(\Gamma_{\leq k})_{\geq j}=(\Delta^1\wedge\cdots\wedge\Delta^m)_{\leq k+1-j}$ e $\Gamma_{\geq k+i}=(\Delta^1\wedge\cdots\wedge\Delta^m)_{>k+1-j}$. Assim, por ($nil$-$\varphi$) tem-se que $\varphi^i_{k+1}M:\langle\Gamma_{<j}.\wedge_{l=1}^{m}\sigma_l.nil\vdash\tau\rangle$ e $\forall 1\leq l\leq m$, se $|\Delta^l|\leq k{+}1-j$ então $\varphi^i_{k+1-j}N:\langle\Delta^l\vdash\sigma_l\rangle$ por (nil-φ) e se $|\Delta^l|>k{+}1-j$ então $\varphi^i_{k+1-j}N:\langle\Delta^l_{\leq k+1-j}.\omega^{\underline{i-1}}.\Delta^l_{>k+1-j}\vdash\sigma_l\rangle$ por (ω-φ). Note que a interseção Δ' dos contextos de $\varphi^i_{k+1-j}N$ pode ser descrita como $\Delta'=(\Delta^1\wedge\cdots\wedge\Delta^m)_{\leq k+1-j}.\omega^{\underline{i-1}}.(\Delta^1\wedge\cdots\wedge\Delta^m)_{>k+1-j}$ pois $\forall 1\leq l\leq m$, se $|\Delta^l|\leq k{+}1-j$ então $\Delta^l_{>k+1-j}=nil$. Logo, $\Delta'=(\Gamma_{\leq k})_{\geq j}.\omega^{\underline{i-1}}.\Gamma_{\geq k+i}$. Portanto, por ($\wedge$-$\sigma$) tem-se que $(\varphi^i_{k+1}M)\,\sigma^j(\varphi^i_{k+1-j}N):\langle(\Gamma_{<j}.nil)\wedge\omega^{\underline{j-1}}.\Delta'\vdash\tau\rangle$ onde $(\Gamma_{<j}.nil)\wedge\omega^{\underline{j-1}}.\Delta'=\Gamma_{<j}.\Delta'=\Gamma_{<j}.(\Gamma_{\le k})_{\ge j}.\omega^{\underline{i-1}}.\Gamma_{\ge k+i}=\Gamma_{\le k}.\omega^{\underline{i-1}}.\Gamma_{\ge k+i}=\Gamma$.

- (φ-φ-transition 1): Seja $\varphi^i_k(\varphi^j_l M):\langle\Gamma\vdash\tau\rangle$, para $l+j\le k$.

  Se $|\Gamma|\le k$, então pelo Lema 7.2.4.1 tem-se que $\varphi^j_l M:\langle\Gamma\vdash\tau\rangle$. Logo, pelo Lema 7.2.4.1 tem-se dois casos.

  Se $|\Gamma|\le l$, então pelo Lema 7.2.4.1 tem-se $M:\langle\Gamma\vdash\tau\rangle$. Note que $k+1-j\ge(l+j)+1-j=l+1>l>|\Gamma|$. Portanto, por ($nil$-$\varphi$) tem-se que $\varphi^i_{k+1-j}M:\langle\Gamma\vdash\tau\rangle$ e $\varphi^j_l(\varphi^i_{k+1-j}M):\langle\Gamma\vdash\tau\rangle$.

  Se $|\Gamma|>l$ então, pelo Lema 7.2.4.1, $M:\langle\Gamma_{\le l}.\Gamma_{\ge l+j}\vdash\tau\rangle$ onde $\Gamma_{\le l}.\omega^{\underline{j-1}}.\Gamma_{\ge l+j}=\Gamma$. Tem-se que $|\Gamma_{\le l}.\Gamma_{\ge l+j}|=|\Gamma|-(j-1)\le k-(j-1)$. Portanto, por ($nil$-$\varphi$) tem-se que $\varphi^i_{k+1-j}M:\langle\Gamma_{\le l}.\Gamma_{\ge l+j}\vdash\tau\rangle$ e, por (ω-φ), $\varphi^j_l(\varphi^i_{k+1-j}M):\langle\Gamma_{\le l}.\omega^{\underline{j-1}}.\Gamma_{\ge l+j}\vdash\tau\rangle$.

  Se $|\Gamma|>k$, então pelo Lema 7.2.4.1 tem-se que $\varphi^j_l M:\langle\Gamma_{\le k}.\Gamma_{\ge k+i}\vdash\tau\rangle$ onde $\Gamma=\Gamma_{\le k}.\omega^{\underline{i-1}}.\Gamma_{\ge k+i}$. Tem-se que $|\Gamma|>k\ge l+j>l$. Logo, pelo Lema 7.2.4.1 tem-se que $M:\langle(\Gamma')_{\le l}.(\Gamma')_{\ge l+j}\vdash\tau\rangle$ onde $\Gamma'=(\Gamma')_{\le l}.\omega^{\underline{j-1}}.(\Gamma')_{\ge l+j}=\Gamma_{\le k}.\Gamma_{\ge k+i}$. Note que $(\Gamma')_{\le l}=\Gamma_{\le l}$ e que $(\Gamma')_{\ge l+j}=(\Gamma_{\le k})_{\ge l+j}.\Gamma_{\ge k+i}$. Assim, $(\Gamma')_{\le l}.(\Gamma')_{\ge l+j}=\Gamma_{\le l}.(\Gamma_{\le k})_{\ge l+j}.\Gamma_{\ge k+i}$ e $|\Gamma_{\le l}.(\Gamma_{\le k})_{\ge l+j}|=l+(k-(l+j-1))=k+1-j$. Portanto, por ($\omega$-$\varphi$) tem-se que $\varphi^i_{k+1-j}M:\langle\Gamma_{\le l}.(\Gamma_{\le k})_{\ge l+j}.\omega^{\underline{i-1}}.\Gamma_{\ge k+i}\vdash\tau\rangle$ e $\varphi^j_l(\varphi^i_{k+1-j}M):\langle\Gamma_{\le l}.\omega^{\underline{j-1}}.(\Gamma_{\le k})_{\ge l+j}.\omega^{\underline{i-1}}.\Gamma_{\ge k+i}\vdash\tau\rangle$, onde $\Gamma_{\le l}.\omega^{\underline{j-1}}.(\Gamma_{\le k})_{\ge l+j}=\Gamma_{\le k}$.

- (φ-φ-transition 2): Seja $\varphi^i_k(\varphi^j_l M):\langle\Gamma\vdash\tau\rangle$, para $l\le k<l+j$.

  Se $|\Gamma|\le k$, então pelo Lema 7.2.4.1 tem-se que $\varphi^j_l M:\langle\Gamma\vdash\tau\rangle$. Observe que se $|\Gamma|>l$ então pelo Lema 7.2.4.1 tem-se que $|\Gamma|>l+(j-1)$ logo $|\Gamma|\ge l+j>k$. Assim, para $|\Gamma|\le l$ tem-se, pelo Lema 7.2.4.1, que $M:\langle\Gamma\vdash\tau\rangle$. Portanto, por ($nil$-$\varphi$), $\varphi^{j+i-1}_l M:\langle\Gamma\vdash\tau\rangle$.

  Se $|\Gamma|>k$, então pelo Lema 7.2.4.1 tem-se que $\varphi^j_l M:\langle\Gamma_{\le k}.\Gamma_{\ge k+i}\vdash\tau\rangle$ onde $\Gamma=\Gamma_{\le k}.\omega^{\underline{i-1}}.\Gamma_{\ge k+i}$. Tem-se que $|\Gamma_{\le k}.\Gamma_{\ge k+i}|>k\ge l$. Assim, pelo Lema 7.2.4.1 tem-se que $M:\langle(\Gamma_{\le k}.\Gamma_{\ge k+i})_{\le l}.(\Gamma_{\le k}.\Gamma_{\ge k+i})_{\ge l+j}\vdash\tau\rangle$ onde $(\Gamma_{\le k}.\Gamma_{\ge k+i})_{\le l}=\Gamma_{\le l}$, $(\Gamma_{\le k})_{>l}=\omega^{\underline{k-l}}$, $(\Gamma_{\le k}.\Gamma_{\ge k+i})_{\ge l+j}=(\Gamma_{\ge k+i})_{\ge l+j-k}=\Gamma_{\ge l+(j+i)-1}$ e $(\Gamma_{<l+(j+i)-1})_{\ge k+i}=\omega^{\underline{(j-1)-(k-l)}}$. Portanto, por ($\omega$-$\varphi$) tem-se $\varphi^{j+i-1}_l M:\langle\Gamma_{\le l}.\omega^{\underline{j+i-2}}.\Gamma_{\ge l+(j+i)-1}\vdash\tau\rangle$,

onde $\Gamma_{\leq l}.\omega^{\underline{j+i-2}}.\Gamma_{\geq l+(j+i)-1} = \Gamma_{\leq l}.(\Gamma_{\leq k})_{>l}.\omega^{\underline{i-1}}.(\Gamma_{<l+(j+i)-1})_{\geq k+i}.\Gamma_{\geq l+(j+i)-1}$. Observe que $\Gamma_{\leq l}.(\Gamma_{\leq k})_{>l} = \Gamma_{\leq k}$ e $(\Gamma_{<l+(j+i)-1})_{\geq k+i}.\Gamma_{\geq l+(j+i)-1} = \Gamma_{\geq k+i}$, portanto $\Gamma_{\leq l}.\omega^{\underline{j+i-2}}.\Gamma_{\geq l+(j+i)-1} = \Gamma$. $\square$

Capítulo 8

Tipos com interseção para o $\lambda\sigma$-calculus

Assim como para o λs_e-calculus no Capítulo 7, utilizamos o conjunto de tipos $\mathcal{T}$ da Definição 5.1.1.1 nos sistemas de tipos com interseção propostos para $\lambda\sigma$. Portanto, os Lemas 5.1.2 e 5.1.3, sobre as propriedade dos tipos e sobre contextos respectivamente, são válidos no presente capítulo.

Os sistemas são baseados em λ_{dB}^{SM}, que é estendido tal que os sistemas obtidos possam inferir uma tipagem aos objetos da classe de substituições. Assim, a primeira abordagem é acrescentar ao λ_{dB}^{SM} as regras de inferência de $\lambda\sigma^{\rightarrow}$ para substituições e o *closure*, obtendo o sistema $\lambda\sigma_r^{\wedge}$. Todas as formas $\lambda\sigma$-normais são tipáveis em $\lambda\sigma_r^{\wedge}$ porém o sistema é muito restrito para todo o cálculo. Assim, a partir das regras de reescrita do $\lambda\sigma$-calculus e tendo como referência a propriedade de SR estendemos o $\lambda\sigma_r^{\wedge}$, obtendo então o sistema $\lambda\sigma^{\wedge}$.

P.-A. Melliès apresentou em [74] um contraexemplo para $\lambda\sigma$ para a propriedade de PSN onde um termo em $\lambda\sigma$, que corresponde a um termo tipável em $\lambda^{\rightarrow}$, tem estratégia de redução infinita em $\lambda\sigma$. Assim, outra semelhança em relação aos sistemas propostos para o λs_e é a de que as considerações e propostas de mudanças nas regras de tipagens têm como objetivo a garantia da propriedade de SR para um sistema de tipos com interseção que seja "o mais restrito possível". Assim, o sistema obtido para $\lambda\sigma$ com a propriedade de SR tem uma relação com a propriedade de relevância, apresentada no Lema 8.2.2.

Diferentemente do Capítulo 7, apresentamos a proposta para $\lambda\sigma_r^{\wedge}$ e, logo a seguir, fazemos as considerações para obtenção de $\lambda\sigma^{\wedge}$. Depois de feita essa análise, apresentamos nas Seções 8.1 e 8.2 as propriedades de $\lambda\sigma_r^{\wedge}$ e $\lambda\sigma^{\wedge}$, respectivamente.

Como mencionado, primeiramente estendemos λ_{dB}^{SM} com as regras de inferência de $\lambda\sigma^{\rightarrow}$. Portanto, o sistema $\lambda\sigma_r^{\wedge}$ é obtido com o acréscimo das seguintes regras:

$$\frac{S:\langle\Gamma\rhd\Gamma'\rangle \qquad M:\langle\Gamma'\vdash\tau\rangle}{M[S]:\langle\Gamma\vdash\tau\rangle}\ (\text{clos})$$

$$\frac{}{id:\langle\Gamma\rhd\Gamma\rangle}\ (\text{id}) \qquad\qquad \frac{}{\uparrow:\langle\omega.\Gamma\rhd\Gamma\rangle}\ (\omega\text{-shift})$$

$$\frac{M:\langle\Gamma\vdash\tau\rangle \qquad S:\langle\Delta\rhd\Delta'\rangle}{M.S:\langle\Gamma\wedge\Delta\rhd\tau.\Delta'\rangle}\ (\text{cons}) \qquad \frac{S:\langle\Gamma\rhd\Gamma''\rangle \qquad S':\langle\Gamma''\rhd\Gamma'\rangle}{S'\circ S:\langle\Gamma\rhd\Gamma'\rangle}\ (\text{comp})$$

Observe que a regra (ω-shift) acima é a única regra diferente das regras apresentadas na Definição 3.3.9, para o sistema $\lambda\sigma^{\rightarrow}$. Se usarmos a mesma regra de $\lambda\sigma^{\rightarrow}$, temos a tipagem $\underline{1}[\uparrow]:\langle\beta.\alpha.nil\vdash\alpha\rangle$. Como o temo $\underline{1}[\uparrow]$ representa $\underline{2}$ em $\lambda\sigma$, o axioma $\uparrow:\langle\tau.\Gamma\vdash\Gamma\rangle$ de $\lambda\sigma^{\rightarrow}$ admite uma lei de redundância no sistema.

O Sistema $\lambda\sigma_r^{\wedge}$ é muito restrito, mesmo para o subconjunto formado pelas expressões em $\lambda\sigma$ geradas pelas reduções de termos correspondentes a termos do λ-calculus. Por exemplo, para algum $|\Gamma|>0$ suponha que

$$\frac{\dfrac{M:\langle\omega.\Gamma\vdash\tau\rangle}{\lambda.M:\langle\Gamma\vdash\omega\rightarrow\tau\rangle} \qquad M':\langle\Delta\vdash\sigma\rangle}{(\lambda.M\ M'):\langle\Gamma\wedge\Delta\vdash\tau\rangle}$$

Pela regra (*Beta*) de $\lambda\sigma$ temos que $(\lambda.M\ M')\rightarrow M[M'.id]$. Por (id) e (cons) como definidos acima temos que $M'.id:\langle\Delta\wedge\Gamma\vdash\sigma.\Gamma\rangle$. Pelo lema de geração para $\lambda\sigma_r^{\wedge}$ provado mais adiante temos que $M:\langle\omega.\Gamma\vdash\tau\rangle$ sempre que $\lambda.M:\langle\Gamma\vdash\omega\rightarrow\tau\rangle$ para $|\Gamma|>0$. Portanto, $M[M'.id]$ não é tipável com as regras de $\lambda\sigma_r^{\wedge}$. Ocorre um problema similar quando o β-redex é tipado com a regra $\rightarrow_e$, para $n>1$.

A primeira tentativa de estender as regras de inferência propostas acima, a fim de resolver os problemas supracitados, é substituir a regra (cons) pelas duas regras a seguir:

$$\frac{M:\langle\Gamma\vdash\tau\rangle \qquad S:\langle\Delta\rhd\Delta'\rangle}{M.S:\langle\Gamma\wedge\Delta\rhd\omega.\Delta'\rangle}\ (\omega\text{-cons})$$

$$\frac{M:\langle\Delta^1\vdash\sigma_1\rangle\ \ldots\ M:\langle\Delta^n\vdash\sigma_n\rangle \qquad S:\langle\Delta\rhd\Delta'\rangle}{M.S:\langle\Delta^1\wedge\cdots\wedge\Delta^n\wedge\Delta\rhd(\wedge_{i=1}^n\sigma_i).\Delta'\rangle}\ (\wedge\text{-cons})$$

Agora, suponha que

$$\frac{\lambda.\underline{2}:\langle\alpha.nil\vdash\omega\rightarrow\alpha\rangle \qquad \underline{1}\,.id:\langle\alpha.nil\rhd\alpha.nil\rangle}{(\lambda.\underline{2}\,)[\,\underline{1}\,.\,id]:\langle\alpha.nil\vdash\omega\rightarrow\alpha\rangle}$$

onde $\underline{2}$ é uma abreviação para $\underline{1}[\uparrow]$. Temos que $(\lambda.\underline{2})[\underline{1}\,.\,id] \rightarrow_{(Abs)} \lambda.\underline{2}[\underline{1}\,.(\underline{1}\,.\,id)\circ\uparrow]$. Seja $\Gamma = \alpha.nil$. Pelas regras (ω-shift) e (comp) tem-se que $(\underline{1}\,.\,id)\circ\uparrow : \langle\omega.\Gamma \rhd \Gamma\rangle$. Observe que, pela regra (var), $\underline{1} : \langle\sigma.nil \vdash \sigma\rangle$ para qualquer $\sigma \in \mathcal{T}$. Portanto, pela regra (ω-cons), $\underline{1}\,.(\underline{1}\,.\,id)\circ\uparrow : \langle\sigma.\Gamma \rhd \omega.\Gamma\rangle$. Por um lema de geração teremos que $\underline{2} : \langle\omega.\Gamma \vdash \alpha\rangle$ logo, pelas regras (clos) e $\rightarrow_i$, teremos que $\lambda.\underline{2}[\underline{1}\,.(\underline{1}\,.\,id)\circ\uparrow] : \langle\Gamma \vdash \sigma\rightarrow\alpha\rangle$.

Consequentemente, além de não possuir SR, uma forma de redundância é introduzida no sistema de tipos com as presentes regras de inferência. Isso ocorre por causa da semântica de *cons*, cujo contexto na tipagem da substituição representa o contexto na tipagem da σ-nf correspondente.

A solução é *"esquecer"* a informação do contexto de um termo associado ao ω, assim que adicionado à substituição. Portanto, introduzimos a seguir a nova regra (ω-cons):

$$\frac{M : \langle\Gamma \vdash \tau\rangle \qquad S : \langle\Delta \rhd \Delta'\rangle}{M.S : \langle\Delta \rhd \omega.\Delta'\rangle}\ (\omega\text{-cons})$$

Note que nesse caso o M representa um termo que corresponde a um índice de de Bruijn que não ocorre. Em outras palavras, quando a substituição $M.S$ é aplicada a um termo M' tipado com o contexto $\omega.\Delta'$, nenhum índice livre de M' será substituído por M. Assim, M e a informação de tipos contida em seu contexto desaparece em $\sigma(M'[M.S])$.

Além disso, seja

$$\frac{\dfrac{M : \langle nil \vdash \tau\rangle}{\lambda.M : \langle nil \vdash \omega\rightarrow\tau\rangle} \qquad M' : \langle\Gamma \vdash \sigma\rangle}{(\lambda.M\ M') : \langle\Gamma \vdash \tau\rangle}$$

Pela regra (ω-cons) temos que $M'.id : \langle nil \rhd \omega.nil\rangle$ e, por um lema de geração, temos que $M : \langle nil \vdash \tau\rangle$ sempre que $\lambda.M : \langle nil \vdash \omega\rightarrow\tau\rangle$. Portanto, $M[M'.id]$ não é tipável com as regras de inferência apresentadas até agora. Existem duas maneiras de resolver esse problema.

A primeira é adicionar à regra (ω-cons) a premissa $\Delta' \neq \omega^{\underline{n}}$, para todo $n \in \mathbb{N}$, introduzindo a regra (nil-cons):

$$\frac{M : \langle\Gamma \vdash \tau\rangle \qquad S : \langle\Delta \rhd nil\rangle}{M.S : \langle\Delta \rhd nil\rangle}\ (nil\text{-cons})$$

Além disso, fazer uma mudança semelhante para (ω-shift), obtendo as duas regras:

$$\frac{\Gamma \neq \omega^{\underline{n}}}{\uparrow : \langle\omega.\Gamma \rhd \Gamma\rangle}\ (\omega\text{-shift}) \qquad\qquad \uparrow : \langle nil \rhd nil\rangle\ (nil\text{-shift})$$

A mudança em (ω-shift) é necessária, caso contrário teríamos

$$\frac{\underline{1}:\langle \alpha.nil \vdash \alpha\rangle \qquad \uparrow:\langle \omega.nil \rhd nil\rangle}{\underline{1}\,.\!\uparrow:\langle \omega.nil \rhd nil\rangle}$$

e $\underline{1}\,.\!\uparrow \rightarrow id$ pela regra ($VarShift$) de $\lambda\sigma$.

A segunda maneira é a de mudar as regras (clos) e (comp) como a seguir:

$$\frac{S:\langle \Gamma \rhd \Gamma' \wedge \omega^{\underline{n}}\rangle \qquad M:\langle \Gamma' \vdash \tau\rangle}{M[S]:\langle \Gamma \vdash \tau\rangle}\ (\omega\text{-clos})$$

$$\frac{S:\langle \Gamma \rhd \Gamma' \wedge \omega^{\underline{n}}\rangle \qquad S':\langle \Gamma' \rhd \Gamma''\rangle}{S' \circ S:\langle \Gamma \rhd \Gamma''\rangle}\ (\omega\text{-comp})$$

Sejam $\lambda\sigma^{\wedge}_{SM}$ o sistema com as regras *nil* e $\lambda\sigma^{\wedge}_{\omega}$ o sistema com as regras (ω-cons) e (ω-clos). Dado um termo M em σ-nf, a principal diferença entre esses dois sistemas é o tipo designado para a subexpressão de uma substituição aplicada a M, que corresponda a índices acima de $sup(M^*)$. Em $\lambda\sigma^{\wedge}_{SM}$ o tipo atribuído para tal subexpressão é *nil*, enquanto em $\lambda\sigma^{\wedge}_{\omega}$ o tipo atribuído é um contexto omega. O sistema $\lambda\sigma^{\wedge}_{\omega}$ permite que seja preservada a correspondência entre o comprimento do tipo de uma substituição, que é um contexto, e o número de *cons*.

Exemplo 8.0.2: Sejam $M \equiv \underline{1}$ e $S \equiv \underline{1}\,.(\lambda.\underline{1}\,).\!\uparrow$ e suponha que $\underline{1}:\langle \tau.nil \vdash \tau\rangle$. Assim, para a inferência de tipagem de $M[S]$ tem-se $S:\langle \tau.nil \rhd \tau.nil\rangle$ em $\lambda\sigma^{\wedge}_{SM}$ com a seguinte derivação:

$$\frac{\underline{1}:\langle \tau.nil \vdash \tau\rangle \qquad \dfrac{\lambda.\underline{1}:\langle nil \vdash \alpha \to \alpha\rangle \qquad \uparrow:\langle nil \rhd nil\rangle}{(\lambda.\underline{1}\,).\!\uparrow:\langle nil \rhd nil\rangle}}{\underline{1}\,.(\lambda.\underline{1}\,).\!\uparrow:\langle \tau.nil \rhd \tau.nil\rangle}$$

e $S:\langle \tau.\omega.nil \rhd \tau.\omega.nil\rangle$ em $\lambda\sigma^{\wedge}_{\omega}$ com a seguinte derivação:

$$\frac{\underline{1}:\langle \tau.nil \vdash \tau\rangle \qquad \dfrac{\lambda.\underline{1}:\langle nil \vdash \alpha \to \alpha\rangle \qquad \uparrow:\langle \omega.nil \rhd nil\rangle}{(\lambda.\underline{1}\,).\!\uparrow:\langle \omega.nil \rhd \omega.nil\rangle}}{\underline{1}\,.(\lambda.\underline{1}\,).\!\uparrow:\langle \tau.\omega.nil \rhd \tau.\omega.nil\rangle}$$

As duas abordagens têm propriedades semelhantes. Com propriedades similares, ambos os sistemas têm problemas similares. Ao checar SR para esses dois sistemas, a propriedade vale para todas as regras do $\lambda\sigma$-calculus menos uma. A falha ocorre para a regra ($MapEnv$), que em [74] é apontada como a causa da estratégia infinita de redução. Seja $(M.S) \circ S'$ tal que

$$\frac{M.S:\langle \Gamma' \rhd \omega.\Gamma''\rangle \qquad S':\langle \Gamma \rhd \Gamma'\rangle}{(M.S) \circ S':\langle \Gamma \rhd \omega.\Gamma''\rangle}$$

no sistema $\lambda\sigma^{\wedge}_{SM}$ ou no sistema $\lambda\sigma^{\wedge}_{\omega}$. Pela regra ($\omega$-cons) temos que $S:\langle\Gamma' \triangleright \Gamma''\rangle$ e que M é tipável. Assim, $S \circ S' : \langle\Gamma \triangleright \omega.\Gamma''\rangle$ mas não temos garantia de que $M[S']$ seja tipável em qualquer um dos sistemas. A seguir, um contraexemplo em $\lambda\sigma^{\wedge}_{\omega}$.

Exemplo 8.0.3: Sejam $A \equiv (\underline{1}\ \underline{1})$ a autoaplicação, $S_1 \equiv A.id$ e $S_2 \equiv \lambda.A.id$. Temos então que $S_1 : \langle\omega.nil \triangleright_{\lambda\sigma^{\wedge}_{\omega}} \omega^{\underline{2}}.nil\rangle$ e $S_2 : \langle nil \triangleright_{\lambda\sigma^{\wedge}_{\omega}} \omega.nil\rangle$. Portanto, pela regra ($\omega$-comp), $S_1 \circ S_2 : \langle nil \triangleright \omega^{\underline{2}}.nil\rangle$. Observe que $S_1 \circ S_2 \rightarrow_{(MapEnv)} A[S_2].(id \circ S_2)$ e $id \circ S_2 : \langle nil \vdash_{\lambda\sigma^{\wedge}_{\omega}} nil\rangle$. A tipabilidade de $A[S_2]$ depende da unificação do contexto da tipagem de A e do tipo da tipagem de S_2, que é reduzido a unificação de $(\alpha \rightarrow \beta) \wedge \alpha$ e $(\alpha' \rightarrow \beta') \wedge \alpha' \rightarrow \beta'$. Portanto, o problema da tipabilidade de $A[S_2]$ se reduz ao da tipabilidade do autorreprodutor $R \equiv \big(\lambda.(\underline{1}\ \underline{1})\ \lambda.(\underline{1}\ \underline{1})\big)$ em sistemas de tipos com interseção.

As mesmas expressões em $\lambda\sigma$ compõem um contraexemplo análogo para $\lambda\sigma^{\wedge}_{SM}$. Observe que o exemplo dado acima não ocorreria como uma subexpressão em termos originados de reduções no $\lambda\sigma$ a partir de termos tipados correspondentes ao λ-calculus. Seja M qualquer tal que $M : \langle nil \vdash_{\lambda\sigma^{\wedge}_{\omega}} \tau\rangle$. Assim, para S_1 e S_2 como no Exemplo 8.0.3 acima, temos $(M[S_1])[S_2] : \langle nil \vdash_{\lambda\sigma^{\wedge}_{\omega}} \tau\rangle$ e $\lambda.M : \langle nil \vdash_{\lambda\sigma^{\wedge}_{\omega}} \omega \rightarrow \tau\rangle$. Analisando as regras de $\lambda\sigma$, o único jeito de obtermos $(M[S_1])[S_2]$ é a partir do termo $M' \equiv \big(\lambda.\big(\lambda.M\ A\big)\ \lambda.A\big)$, onde A é a autoaplicação. Seja

$$\dfrac{\dfrac{\lambda.M : \langle nil \vdash \omega \rightarrow \tau\rangle \qquad A : \langle(\alpha \rightarrow \beta) \wedge \alpha.nil \vdash \beta\rangle}{(\lambda.M\ A) : \langle(\alpha \rightarrow \beta) \wedge \alpha.nil \vdash \tau\rangle}}{\lambda.(\lambda.M\ A) : \langle nil \vdash ((\alpha \rightarrow \beta) \wedge \alpha) \rightarrow \tau\rangle}$$

Note que a tipabilidade de M' reduz ao problema de tipabilidade para o autorreprodutor em IT logo M' não é tipável. Essa não tipabilidade de M' em ambos os sistemas é devida à adição da informação de tipos proveniente do contexto da tipagem do termo aplicado a ω pela regra $\rightarrow'_e$.

Podemos então mudar as regras (ω-cons) e (nil-cons), retirando das respectivas premissas que o termo M deve ser tipável, obtendo então:

$$(\omega\text{-cons})\ \frac{S : \langle\Delta \triangleright nil\rangle}{M.S : \langle\Delta \triangleright nil\rangle} \qquad (nil\text{-cons})\frac{S : \langle\Delta \triangleright \Delta'\rangle}{M.S : \langle\Delta \triangleright \omega.\Delta'\rangle}, \text{onde } \Delta' \neq \omega^{\underline{n}}$$

Apesar do fato de expressões WN em $\lambda\sigma$, que correspondem a termos do λ-calculus, não serem tipáveis em $\lambda\sigma^{\wedge}_{\omega}$ nem em $\lambda\sigma^{\wedge}_{SM}$, o exemplo de Melliés [74], que mostra que o $\lambda\sigma$-calculus simplesmente tipado não é PSN, é tipável em ambos.

Portanto, por não ser possível uma caracterização de termos SN em $\lambda\sigma$, substituímos a regra $\to_e'$ pela regra $\to_e^{\omega}$ a seguir:

$$\frac{M:\langle\Gamma\vdash\omega\to\tau\rangle}{(M\ N):\langle\Gamma\vdash\tau\rangle}\ \to_e^{\omega}$$

Assim como a mudança para $\lambda s_e{}^{\wedge}$ em relação λs^{SM}, o fato de não termos nenhuma informação de tipo sobre os índices livres de um termo aplicado a ω, faz com que percamos qualquer relação direta entre contextos de tipagens e propriedades sintáticas do termo correspondente. Na seção 8.2 apresentamos o sistema $\lambda\sigma^{\wedge}$, baseado em $\lambda\sigma_{SM}^{\wedge}$ e com a regra $\to_e^{\omega}$ descrita acima. Apesar de não se ter a noção exata do que seria relevância para $\lambda\sigma^{\wedge}$, provamos uma propriedade similar ao Lema 7.2.2 para $\lambda s_e{}^{\wedge}$, sobre o último elemento de um contexto não nulo.

8.1 O sistema $\lambda\sigma_r^{\wedge}$ e propriedades

Nesta seção apresentaremos algumas propriedades do sistema $\lambda\sigma_r^{\wedge}$. Apesar de não possuir SR, provamos que toda forma $\lambda\sigma$-normal é tipável com esse sistema.

Definição 8.1.1 (O sistema $\lambda\sigma_r^{\wedge}$): As regras de tipagem do sistema $\lambda\sigma_r^{\wedge}$ são dadas a seguir:

$$\frac{}{\underline{1}:\langle\tau.nil\vdash\tau\rangle}\ (\text{var}) \qquad \frac{M:\langle u.\Gamma\vdash\tau\rangle}{\lambda.M:\langle\Gamma\vdash u\to\tau\rangle}\ \to_i$$

$$\frac{M_1:\langle\Gamma\vdash\omega\to\tau\rangle \qquad M_2:\langle\Delta\vdash\sigma\rangle}{(M_1\ M_2):\langle\Gamma\wedge\Delta\vdash\tau\rangle}\ \to_e' \qquad \frac{M:\langle nil\vdash\tau\rangle}{\lambda.M:\langle nil\vdash\omega\to\tau\rangle}\ \to_i'$$

$$\frac{M_1:\langle\Gamma\vdash\wedge_{i=1}^{n}\sigma_i\to\tau\rangle \qquad M_2:\langle\Delta^1\vdash\sigma_1\rangle\ \ldots\ M_2:\langle\Delta^n\vdash\sigma_n\rangle}{(M_1\ M_2):\langle\Gamma\wedge\Delta^1\wedge\cdots\wedge\Delta^n\vdash\tau\rangle}\ \to_e$$

$$\frac{S:\langle\Gamma\rhd\Gamma'\rangle \qquad M:\langle\Gamma'\vdash\tau\rangle}{M[S]:\langle\Gamma\vdash\tau\rangle}\ (\text{clos})$$

$$\frac{}{id:\langle\Gamma\rhd\Gamma\rangle}\ (\text{id}) \qquad \frac{}{\uparrow:\langle\omega.\Gamma\rhd\Gamma\rangle}\ (\omega\text{-shift})$$

$$\frac{M:\langle\Gamma\vdash\tau\rangle \qquad S:\langle\Delta\rhd\Delta'\rangle}{M.S:\langle\Gamma\wedge\Delta\rhd\tau.\Delta'\rangle}\ (\text{cons}) \qquad \frac{S:\langle\Gamma\rhd\Gamma''\rangle \qquad S':\langle\Gamma''\rhd\Gamma'\rangle}{S'\circ S:\langle\Gamma\rhd\Gamma'\rangle}\ (\text{comp})$$

Os lemas de geração mencionados no início da seção, durante as considerações sobre as mudanças nas regra de inferência para a obtenção de um sistema de IT apropriado para $\lambda\sigma$, são apresentados a seguir.

Lema 8.1.2 (Geração para $\lambda\sigma_r^\wedge$):

1. $\uparrow^n : \langle \Gamma \rhd \Gamma' \rangle$ se, e somente se, $\Gamma = \omega^{\underline{n}}.\Gamma'$.

2. $\underline{1}\,[\uparrow^n] : \langle \Gamma \vdash \tau \rangle$ se, e somente se, $\Gamma = \omega^{\underline{n}}.\tau.nil$.

3. Se $\lambda.M : \langle nil \vdash \tau \rangle$, então $\tau = \omega \to \sigma$ e $M : \langle nil \vdash_{\lambda\sigma_r^\wedge} \sigma \rangle$ ou $\tau = \wedge_{i=1}^n \sigma_i \to \sigma$, $n > 0$, e $M : \langle \wedge_{i=1}^n \sigma_i.nil \vdash_{\lambda\sigma_r^\wedge} \sigma \rangle$ onde $\sigma, \sigma_1, \ldots, \sigma_n \in \mathcal{T}$.

4. Se $\lambda.M : \langle \Gamma \vdash \tau \rangle$ e $|\Gamma| > 0$, então $\tau = u \to \sigma$ para algum $u \in \mathcal{U}$ e $\sigma \in \mathcal{T}$, onde $M : \langle u.\Gamma \vdash_{\lambda\sigma_r^\wedge} \sigma \rangle$.

Demonstração. 1. Por indução em n.

2. Suponha que $\underline{1}\,[\uparrow^n] : \langle \Gamma \vdash \tau \rangle$. Por (clos) tem-se que $\uparrow : \langle \Gamma \rhd \Gamma' \rangle$ e $\underline{1} : \langle \Gamma' \vdash \tau \rangle$. Pelo item 1 acima tem-se que $\uparrow : \langle \Gamma \rhd \Gamma' \rangle$ sse $\Gamma = \omega^{\underline{n}}.\Gamma'$ e, por (var), tem-se que $\underline{1} : \langle \Gamma' \vdash \tau \rangle$ sse $\Gamma' = \tau.nil$.

3. Por análise de casos na derivação de $\lambda.M : \langle nil \vdash \tau \rangle$.

4. Por análise de casos na derivação de $\lambda.M : \langle \Gamma \vdash \tau \rangle$, para $|\Gamma| > 0$. □

Pela Proposição 3.3.7 temos que as formas σ-normais correspondem exatamente aos termo em λ_{dB}. O lema a seguir mostra que a teoria de $\lambda\sigma_r^\wedge$ coincide com a teoria de λ_{dB}^{SM} para esses termos.

Lema 8.1.3: Seja M uma σ-nf. Então $M : \langle \Gamma \vdash_{\lambda\sigma_r^\wedge} \tau \rangle$ se, e somente se, $M^* : \langle \Gamma \vdash_{\lambda_{dB}^{SM}} \tau \rangle$.

Demonstração. Por indução na estrutura de M, descrita no Lema 3.3.5. Note que, pela Proposição 3.3.7, a aplicação $*$ introduzida na Definição 3.3.6 é um isomorfismo entre as σ-nfs e Λ_{dB}.

- $M \equiv \underline{1}$: nada há provar.

- $M \equiv \underline{1}[\uparrow^n]$: Suponha que $\underline{1}[\uparrow^n]:\langle\Gamma \vdash \tau\rangle$. Temos que $(\underline{1}[\uparrow^n])^* = \underline{n{+}1}$. Pelo Lema 8.1.2.2 tem-se que $\Gamma = \omega^{\underline{n}}.\tau.nil$ logo, por (var) e (varn) em λ_{dB}^{SM}, $\underline{n{+}1}:\langle\Gamma \vdash \tau\rangle$. Se $\underline{n{+}1}:\langle\Gamma' \vdash \sigma\rangle$ então, pelos Lemas 5.1.7 e 5.1.8.1, $\Gamma' = \omega^{\underline{n}}.\sigma.nil$. Logo, por (var), ($\omega$-shift) e (clos) em $\lambda\sigma_r^{\wedge}$, $\underline{1}[\uparrow^n]:\langle\Gamma' \vdash \sigma\rangle$.

- $M \equiv \lambda.M_1$. Suponha que $\lambda.M_1:\langle\Gamma \vdash \tau\rangle$.

  Se $\Gamma = nil$ então pelo Lema 8.1.2.3, $\tau = \omega \to \tau'$ e $M_1:\langle nil \vdash \tau'\rangle$ ou $\tau = \wedge_{i=1}^m \sigma_i \to \tau'$ e $M_1:\langle \wedge_{i=1}^m \sigma_i.nil \vdash \tau'\rangle$. No primeiro caso tem-se, por HI, que $M_1:\langle nil \vdash_{\lambda\sigma_r^{\wedge}} \tau'\rangle$ sse $M_1^*:\langle nil \vdash_{\lambda_{dB}^{SM}} \tau'\rangle$. Logo, pela regra $\to_i'$ de cada sistema tem-se $\lambda.M_1:\langle nil \vdash_{\lambda\sigma_r^{\wedge}} \omega \to \tau'\rangle$ e $\lambda.M_1^*:\langle nil \vdash_{\lambda_{dB}^{SM}} \omega \to \tau'\rangle$. O segundo caso é similar, com o uso da regra $\to_i$ de cada sistema.

  Se $|\Gamma| > 0$, então pelo Lema 8.1.2.4, $\tau = u \to \tau'$ e $M_1:\langle u.\Gamma \vdash \tau'\rangle$. Assim, por HI, $M_1:\langle u.\Gamma \vdash_{\lambda\sigma_r^{\wedge}} \tau'\rangle$ sse $M_1^*:\langle u.\Gamma \vdash_{\lambda_{dB}^{SM}} \tau'\rangle$. Logo, pela regra $\to_i$ de cada sistema, $\lambda.M_1:\langle\Gamma \vdash_{\lambda\sigma_r^{\wedge}} u \to \tau'\rangle$ e $\lambda.M_1^*:\langle\Gamma \vdash_{\lambda_{dB}^{SM}} u \to \tau'\rangle$.

  Note que a prova de suficiência nos caso descritos acima é devido a derivação de $\lambda.M_1^*:\langle\Gamma \vdash_{\lambda_{dB}^{SM}} \tau\rangle$ ser tratada por casos similares aos descritos acima, a partir dos Lemas 5.1.8.3 e 5.1.8.4

- $M \equiv (M_1\ M_2)$: Suponha que $(M_1\ M_2):\langle\Gamma \vdash \tau\rangle$. Por análise de casos tem-se duas possibilidades para o último passo na derivação.

  Na primeira possibilidade, $M_1:\langle\Gamma^1 \vdash \omega \to \tau\rangle$ e $M_2:\langle\Gamma^2 \vdash \sigma\rangle$, onde $\Gamma = \Gamma^1 \wedge \Gamma^2$. Assim, por HI, $M_1:\langle\Gamma^1 \vdash_{\lambda\sigma_r^{\wedge}} \omega \to \tau\rangle$ sse $M_1^*:\langle\Gamma^1 \vdash_{\lambda_{dB}^{SM}} \omega \to \tau\rangle$ e $M_2:\langle\Gamma^2 \vdash_{\lambda\sigma_r^{\wedge}} \sigma\rangle$ sse $M_2^*:\langle\Gamma^2 \vdash_{\lambda_{dB}^{SM}} \sigma\rangle$. Portanto, por $\to_e'$ nos respectivos sistemas, $(M_1\ M_2):\langle\Gamma \vdash_{\lambda\sigma_r^{\wedge}} \tau\rangle$ e $(M_1^*\ M_2^*):\langle\Gamma \vdash_{\lambda_{dB}^{SM}} \tau\rangle$.

  Na segunda possibilidade, $M_1:\langle\Gamma' \vdash \wedge_{i=1}^m \sigma_i \to \tau\rangle$ e $\forall 1 \le i \le m$, $M_2:\langle\Gamma^i \vdash \sigma_i\rangle$ onde $\Gamma = \Gamma' \wedge \Gamma^1 \wedge \cdots \wedge \Gamma^m$. Assim, por HI $M_1:\langle\Gamma' \vdash_{\lambda\sigma_r^{\wedge}} \omega \to \tau\rangle$ sse $M_1^*:\langle\Gamma' \vdash_{\lambda_{dB}^{SM}} \omega \to \tau\rangle$ e $\forall 1 \le i \le m$, $M_2:\langle\Gamma^i \vdash_{\lambda\sigma_r^{\wedge}} \sigma_i\rangle$ sse $M_2^*:\langle\Gamma^i \vdash_{\lambda_{dB}^{SM}} \sigma_i\rangle$. Portanto, com as regras $\to_e$ dos respectivos sistemas o resultado é valido.

  As provas de suficiência decorrem das possibilidades do ultimos passo de $(M_1^*\ M_2^*):\langle\Gamma \vdash_{\lambda_{dB}^{SM}} \tau\rangle$ corresponderem aos passos descritos acima. □

Para finalizar, apresentamos o lema que estabelece a tipabilidade de toda $\lambda\sigma$-nf.

Lema 8.1.4: Toda forma $\lambda\sigma$-normal é tipável em $\lambda\sigma_r^{\wedge}$.

Demonstração. Por indução na estrutura das $\lambda\sigma$-nfs, descrita no Lema 3.3.8.

- Se $N \equiv \underline{1}$, nada há provar.
- Sejam $N \equiv \underline{1}\,[\uparrow^n]$ e $\tau \in \mathcal{T}$. Pelo Lema 8.1.2.2, $\underline{1}\,[\uparrow^n] : \langle \omega^{\underline{n}}.\tau.nil \vdash \tau \rangle$.
- Seja $N \equiv \underline{1}\, N_1 \cdots N_m$. Por HI tem-se que $\forall 1 \le i \le m$, N_i é tipável em $\lambda\sigma_r^{\wedge}$ então seja $N_i : \langle \Gamma^i \vdash \sigma_i \rangle$. Seja $\tau = \sigma_1 \to \cdots \to \sigma_m \to \beta$ então, por (var), $\underline{1} : \langle \tau.nil \vdash \tau \rangle$. Assim, pela regra $\to_e$ aplicada m-vezes, $\underline{1}\, N_1 \cdots N_m : \langle (\omega^{\underline{n}}.\tau.nil) \wedge \Gamma^1 \wedge \cdots \wedge \Gamma^m \vdash \beta \rangle$.
- Seja $N \equiv \underline{1}\,[\uparrow^n]\, N_1 \cdots N_m$. Por HI, $\forall 1 \le i \le m$, N_i é tipável em $\lambda\sigma_r^{\wedge}$ então seja $N_i : \langle \Gamma^i \vdash \sigma_i \rangle$ e seja $\tau = \sigma_1 \to \cdots \to \sigma_m \to \beta$. Portanto, pelo Lema 8.1.2.2 tem-se que $\underline{1}\,[\uparrow^n] : \langle \omega^{\underline{n}}.\tau.nil \vdash \tau \rangle$ e, por m aplicações de $\to_e$, $\underline{1}\,[\uparrow^n]\, N_1 \cdots N_m : \langle (\omega^{\underline{n}}.\tau.nil) \wedge \Gamma^1 \wedge \cdots \wedge \Gamma^m \vdash \beta \rangle$.
- Seja $N \equiv \lambda.N'$. Por HI, $N' : \langle \Gamma' \vdash \sigma \rangle$. Se $\Gamma' = u.\Gamma$ então, por $\to_i$, $\lambda.N' : \langle \Gamma \vdash u \to \sigma \rangle$. Senão, por $\to_i'$, $\lambda.N' : \langle nil \vdash \omega \to \sigma \rangle$.
- Seja $S \equiv N_1. \cdots .N_m.\uparrow^n$, onde $m \neq n$. Por HI tem-se que $\forall 1 \le i \le m$, N_i é tipável então seja $N_i : \langle \Gamma^i \vdash \sigma_i \rangle$. Se $n > 0$ então, pelo Lema 8.1.21, $\uparrow^n : \langle \omega^{\underline{n}}.\Gamma \rhd \Gamma \rangle$ para qualquer contexto Γ. Assim, pela regra (cons) m-vezes, $N_1. \cdots .N_m.\uparrow^n : \langle \Gamma^1 \wedge \cdots \wedge \Gamma^m \wedge (\omega^{\underline{n}}.\Gamma) \rhd \sigma_1. \cdots \sigma_m.\Gamma \rangle$. Analogamente, com a regras (id) e (cons), $N_1. \cdots .N_m.id : \langle \Gamma^1 \wedge \cdots \wedge \Gamma^m \wedge \Gamma \rhd \sigma_1. \cdots \sigma_m.\Gamma \rangle$. □

8.2 O sistema $\lambda\sigma^{\wedge}$ e propriedades

Introduzimos nesta seção o sistema de IT proposto ao $\lambda\sigma$-calculus, o $\lambda\sigma^{\wedge}$, onde algumas de suas propriedades são apresentadas. Ao contrário do sistema $\lambda\sigma_r^{\wedge}$, o sistema $\lambda\sigma^{\wedge}$ possui a propriedade de SR. Além disso, estabelecemos uma propriedade relacionada à relevância. Assim como para o sistema $\lambda s^{\wedge}$ em relação ao sistema λs^{SM}, introduzidos no Capítulo 7, as alterações propostas ao sistema $\lambda\sigma_r^{\wedge}$ para a obtenção de um sistema de IT com a propriedade de SR para o $\lambda\sigma$, impedem a caracterização de SN de uma $\lambda\sigma$-expressão através da tipabilidade no sistema. A seguir, apresentamos a definição do sistema $\lambda\sigma^{\wedge}$.

Definição 8.2.1 (O sistema $\lambda\sigma^\wedge$): As regras para tipagem no sistema $\lambda\sigma^\wedge$ são dadas a seguir, onde $m > 0$ e $n \geq 0$:

$$\frac{}{\underline{1}:\langle \tau.nil \vdash \tau\rangle}\ (\text{var}) \qquad \frac{M:\langle u.\Gamma \vdash \tau\rangle}{\lambda.M:\langle \Gamma \vdash u\to\tau\rangle}\ \to_i$$

$$\frac{M_1:\langle \Gamma \vdash \omega\to\tau\rangle}{(M_1\ M_2):\langle \Gamma \vdash \tau\rangle}\ \to_e^\omega \qquad \frac{M:\langle nil \vdash \tau\rangle}{\lambda.M:\langle nil \vdash \omega\to\tau\rangle}\ \to_i'$$

$$\frac{M_1:\langle \Gamma \vdash \wedge_{i=1}^m \sigma_i\to\tau\rangle \qquad M_2:\langle \Delta^1 \vdash \sigma_1\rangle\ \ldots\ M_2:\langle \Delta^m \vdash \sigma_m\rangle}{(M_1\ M_2):\langle \Gamma\wedge\Delta^1\wedge\cdots\wedge\Delta^m \vdash \tau\rangle}\ \to_e$$

$$(\text{clos})\ \frac{S:\langle \Gamma \rhd \Gamma'\rangle \qquad M:\langle \Gamma' \vdash \tau\rangle}{M[S]:\langle \Gamma \vdash \tau\rangle}$$

$$(\wedge\text{-cons})\ \frac{M:\langle \Delta^1 \vdash \sigma_1\rangle\ \ldots\ M:\langle \Delta^m \vdash \sigma_m\rangle \qquad S:\langle \Delta \rhd \Delta'\rangle}{M.S:\langle \Delta\wedge\Delta^1\wedge\cdots\wedge\Delta^m \rhd (\wedge_{i=1}^m\sigma_i).\Delta'\rangle}$$

$$(\text{id})\ \frac{\Gamma \neq \Delta.\underline{\omega^m}}{id:\langle \Gamma \rhd \Gamma\rangle} \qquad (\text{comp})\ \frac{S:\langle \Gamma \rhd \Gamma''\rangle \qquad S':\langle \Gamma'' \rhd \Gamma'\rangle}{S'\circ S:\langle \Gamma \rhd \Gamma'\rangle}$$

$$(nil\text{-shift})\ \frac{}{\uparrow:\langle nil \rhd nil\rangle} \qquad (nil\text{-cons})\ \frac{S:\langle \Delta \rhd nil\rangle}{M.S:\langle \Delta \rhd nil\rangle}$$

$$(\omega\text{-shift})\ \frac{\Gamma \neq \Delta.\underline{\omega^n}}{\uparrow:\langle \omega.\Gamma \rhd \Gamma\rangle} \qquad (\omega\text{-cons})\ \frac{S:\langle \Delta \rhd \Delta'\rangle}{M.S:\langle \Delta \rhd \omega.\Delta'\rangle},\ \Delta' \neq \underline{\omega^n}$$

Note que a premissa para a regra (ω-shift) foi alterada de forma a excluir, além de qualquer contexto omega, qualquer contexto que termine com um contexto omega. Uma premissa similar sobre contextos foi incluída para (id). Note que (id) não exclui $\Gamma = nil$ enquanto que (ω-shift) exclui essa possibilidade. Essas condições extra garantem que toda substituição, e não apenas as aplicadas a termos, tenham a propriedade apresentada no lema a seguir.

Lema 8.2.2: Se $M:\langle \Gamma \vdash_{\lambda\sigma^\wedge} \tau\rangle$ e $m = |\Gamma| > 0$, então $\Gamma_m \neq \omega$. Em particular, se $S:\langle \Gamma \rhd \Gamma'\rangle$ e $m = |\Gamma| > 0$ então $\Gamma_m \neq \omega$ e se $m' = |\Gamma'| > 0$ então $\Gamma'_{m'} \neq \omega$.

Demonstração. A demonstração é feita por indução em $M:\langle \Gamma \vdash \tau\rangle$, com subindução em $S:\langle \Gamma \rhd \Gamma'\rangle$. A prova para as regras (var), $\to_i$, $\to_e$ e $\to_e^\omega$ são similares as provas no Lema 7.2.2 para $\lambda s_e{}^\wedge$. Apresentamos a prova para as regras restantes.

- Se $\dfrac{\Gamma \neq \Delta.\omega^{\underline{m}}}{id : \langle \Gamma \rhd \Gamma \rangle}$, nada há provar.

- Se $\uparrow : \langle nil \rhd nil \rangle$, nada há provar.

- Se $\dfrac{\Gamma \neq \Delta.\omega^{\underline{n}}}{\uparrow : \langle \omega.\Gamma \rhd \Gamma \rangle}$, nada há provar.

- Seja $\dfrac{S : \langle \Delta \rhd nil \rangle}{M.S : \langle \Delta \rhd nil \rangle}$. Se $m = |\Delta| > 0$, então por HI tem-se que $\Delta_m \neq \omega$.

- Seja $\dfrac{S : \langle \Delta \rhd \Delta' \rangle}{M.S : \langle \Delta \rhd \omega.\Delta' \rangle}$, $\Delta' \neq \omega^{\underline{n}}$. Por HI tem-se que se $m = |\Delta| > 0$, então $\Delta_m \neq \omega$ e para $m' = |\Delta'|$ tem-se que $\Delta'_{m'} \neq \omega$. Assim, $|\omega.\Delta'| = m'+1$ e $(\omega.\Delta')_{m'+1} = \Delta'_{m'} \neq \omega$.

- Seja $\dfrac{M : \langle \Delta^1 \vdash \sigma_1 \rangle \ \ldots \ M : \langle \Delta^m \vdash \sigma_m \rangle \qquad S : \langle \Delta \rhd \Delta' \rangle}{M.S : \langle \Delta \wedge \Delta^1 \wedge \cdots \wedge \Delta^m \rhd (\wedge_{i=1}^m \sigma_i).\Delta' \rangle}$. Por HI tem-se que se $m' = |\Delta'| > 0$ então $\Delta'_{m'} \neq \omega$. Note que $|\Delta \wedge \Delta^1 \wedge \cdots \wedge \Delta^m| = max(|\Delta|, |\Delta^1|, \ldots, |\Delta^m|)$. Assim, se $m = |\Delta \wedge \Delta^1 \wedge \cdots \wedge \Delta^m| > 0$, então basta tomar algum contexto de comprimento m e mostrar que ele satisfaz a propriedade. Suponha s.p.d.g. que $|\Delta| = m$. Portanto, por HI tem-se que se $m = |\Delta| > 0$, então $\Delta_m \neq \omega$.

- Seja $\dfrac{S : \langle \Gamma \rhd \Gamma'' \rangle \qquad S' : \langle \Gamma'' \rhd \Gamma' \rangle}{S' \circ S : \langle \Gamma \rhd \Gamma' \rangle}$. Por HI tem-se que se $m = |\Gamma| > 0$ então $\Gamma_m \neq \omega$ e se $m' = |\Gamma'| > 0$ então $\Gamma'_{m'} \neq \omega$.

- Seja $\dfrac{S : \langle \Gamma \rhd \Gamma' \rangle \qquad M : \langle \Gamma' \vdash \tau \rangle}{M[S] : \langle \Gamma \vdash \tau \rangle}$. Por HI tem-se que se $m = |\Gamma| > 0$ então $\Gamma_m \neq \omega$.

Note que a prova para substituições tem como base as tipagens de id e $\uparrow$. Portanto, as condições para contextos adicionadas nas premissas é essencial. $\square$

Corolário 8.2.3: Se $M : \langle \Gamma \vdash_{\lambda\sigma^{\wedge}} \tau \rangle$, então $\Gamma \neq \Delta.\omega^{\underline{m}}$, para quaisquer contexto Δ e $m > 0$. Em particular, se $S : \langle \Gamma \rhd_{\lambda\sigma^{\wedge}} \Gamma' \rangle$ então $\Gamma \neq \Delta.\omega^{\underline{m}}$ e $\Gamma' \neq \Delta'.\omega^{\underline{m}}$, para quaisquer contextos Δ e Δ' e $m > 0$.

As novas condições para contexto garantem a propriedade apresentada no Corolário 8.2.3 acima, para toda substituição tipável em $\lambda\sigma^{\wedge}$. Essa propriedade não é necessária para garantir a propriedade de SR. Para tal, basta adicionar a premissa $\Gamma \neq \omega^{\underline{m}}$ para (id), onde $m > 0$. Nesse caso o Lema 8.2.2, e consequentemente o corolário acima, é válido para termos e apenas para substituições aplicadas a termos. A seguir apresentamos os lemas de geração para o sistema, separados em geração para substituições e para termos.

Lema 8.2.4 (Geração para substituições em $\lambda\sigma^{\wedge}$):

1. $S:\langle nil \rhd nil\rangle$ para qualquer substituição S.
2. Se $M.S:\langle\Gamma \rhd nil\rangle$ então $S:\langle\Gamma \rhd nil\rangle$.
3. Se $M.S:\langle\Gamma \rhd \omega.\Gamma'\rangle$ então $S:\langle\Gamma \rhd \Gamma'\rangle$ e $\Gamma' \neq \omega^{\underline{n}}$.
4. Se $M.S:\langle\Gamma \rhd \Gamma'\rangle$ e $\Gamma' = \wedge_{i=1}^{m}\sigma_i.\Gamma''$ então $S:\langle\Gamma''' \rhd \Gamma''\rangle$ e $\forall 1 \leq i \leq m$, $M:\langle\Gamma^i \vdash \sigma_i\rangle$ tal que $\Gamma = \Gamma''' \wedge \Gamma^1 \wedge \cdots \wedge \Gamma^m$.
5. Se $S:\langle\Gamma \rhd nil\rangle$ então $\Gamma = nil$.
6. $\uparrow^m:\langle\Gamma \rhd \Gamma'\rangle$ se, e somente se, $\Gamma = \Gamma' = nil$ ou $\Gamma = \omega^{\underline{m}}.\Gamma'$, onde $\Gamma' \neq \Delta.\omega^{\underline{n}}$.
7. Se $S:\langle\Gamma \rhd \Gamma'\rangle$ e $S:\langle\Delta \rhd \Delta'\rangle$ então $S:\langle\Gamma \wedge \Delta \rhd \Gamma' \wedge \Delta'\rangle$.
8. Se $S:\langle\Gamma \rhd \Delta^1 \wedge \Delta^2\rangle$ para $\Delta^1 \neq \Delta'.\omega^{\underline{m}}$ e $\Delta^2 \neq \Delta''.\omega^{\underline{m}}$, então $\Gamma = \Gamma^1 \wedge \Gamma^2$ tal que $S:\langle\Gamma^1 \rhd \Delta^1\rangle$ e $S:\langle\Gamma^2 \rhd \Delta^2\rangle$.

Demonstração. 1. Por indução na estrutura de S.

2. Suponha que $M.S:\langle\Gamma \rhd nil\rangle$. Por análise de casos, a única possibilidade para o último passo da derivação é a regra (*nil*-cons). Portanto, $S:\langle\Gamma \rhd nil\rangle$.
3. Suponha que $M.S:\langle\Gamma \rhd \omega.\Gamma'\rangle$. Por análise de casos, o último passo é a regra (ω-cons). Portanto, $S:\langle\Gamma \rhd \Gamma'\rangle$ e $\Gamma \neq \omega^{\underline{n}}$.
4. Por análise de casos a única regra possível no último passo da dedução será a regra ($\wedge$-cons), validando a afirmação.
5. Por indução na estrutura de S. Suponha que $S:\langle\Gamma \rhd nil\rangle$.
   - Se $S \equiv id$ ou $S \equiv \uparrow$, nada há provar.
   - Se $S \equiv S_1 \circ S_2$, então por (comp) tem-se que $S_1:\langle\Gamma' \vdash nil\rangle$ e $S_2:\langle\Gamma \vdash \Gamma'\rangle$. Por HI em S_1 tem-se que $\Gamma' = nil$ e, por HI em S_2, tem-se $\Gamma = nil$.
   - Se $S \equiv M.S'$ então, pelo item 2, $S':\langle\Gamma \rhd nil\rangle$. Assim, por HI, $\Gamma = nil$.

6. Se $\Gamma'\!=\!nil$ então pelo item 5 acima tem-se que $\Gamma\!=\!nil$. Senão a hipótese é provada por indução em m.

7. Por indução na estrutura de S. Suponha que $S\!:\!\langle\Gamma \rhd \Gamma'\rangle$ e que $S\!:\!\langle\Delta \rhd \Delta'\rangle$. Observe que se $\Gamma'\!=\!nil$ ou $\Delta'\!=\!nil$ então, pelo item 5, $\Gamma\!=\!nil$ e $\Delta\!=\!nil$, respectivamente, e a afirmação é verdadeira. Assim, consideram-se apenas os casos onde $\Gamma', \Delta'\!\neq\! nil$.

   - Se $S \equiv id$, então por (id) tem-se que $\Gamma\!=\!\Gamma'$ e $\Delta\!=\!\Delta'$ e $\Gamma\!\neq\!\Gamma''.\omega^{\underline{m}}$ e $\Delta\!\neq\!\Delta''.\omega^{\underline{m}}$, para qualquer $m\!>\!0$. Assim, $\Gamma\!\wedge\!\Delta\!\neq\!\Gamma''.\omega^{\underline{m}}$ logo, por (id), $id\!:\!\langle\Gamma \wedge \Delta \vdash \Gamma' \wedge \Delta'\rangle$ onde $\Gamma' \wedge \Delta'\!=\!\Gamma \wedge \Delta$.
   - Seja $S \equiv\uparrow$ e suponha que $\Gamma', \Delta' \neq nil$. Então pelo item 6 tem-se que $\Gamma\!=\!\omega.\Gamma'$ e $\Delta\!=\!\omega.\Delta'$, onde $\Gamma'\!\neq\!\Gamma''.\omega^{\underline{n}}$ e $\Delta'\!\neq\!\Delta''.\omega^{\underline{n}}$. Tem-se que $(\omega.\Gamma')\!\wedge\!(\omega.\Delta')\!=\!\omega.(\Gamma'\!\wedge\!\Delta')$ logo, por (ω-shift), $\uparrow\!:\!\langle\Gamma \wedge \Delta \rhd \Gamma' \wedge \Delta'\rangle$.
   - Seja $S \equiv S_1 \circ S_2$. Por (comp) tem-se que $S_1 : \langle\Gamma'' \rhd \Gamma'\rangle$, $S_2 : \langle\Gamma \rhd \Gamma''\rangle$, $S_1 : \langle\Delta'' \rhd \Delta'\rangle$ e $S_2 : \langle\Delta \rhd \Delta''\rangle$. Por HI tem-se que $S_1 : \langle\Gamma'' \wedge \Delta'' \rhd \Gamma' \wedge \Delta'\rangle$ e $S_2\!:\!\langle\Gamma \wedge \Delta \rhd \Gamma'' \wedge \Delta''\rangle$ logo, por (comp), $S_1 \circ S_2\!:\!\langle\Gamma \wedge \Delta \rhd \Gamma' \wedge \Delta'\rangle$.
   - Seja $S \equiv M.S'$. Suponha s.p.d.g. que $\Gamma' = \omega.\Gamma''$ e $\Delta' = (\wedge_{i=1}^m \sigma_i).\Delta''$. Pelo item 3 tem-se que $S' : \langle\Gamma \rhd \Gamma''\rangle$ para $\Gamma'' \neq \omega^{\underline{n}}$. Pelo item 4 tem-se que $S' : \langle\Delta''' \rhd \Delta''\rangle$ onde $\Delta = \Delta''' \wedge \Delta^1 \wedge \cdots \wedge \Delta^m$ e $\forall 1 \leq i \leq m$, $M : \langle\Delta^i \vdash \sigma_i\rangle$. Assim, por HI, $S' : \langle\Gamma \wedge \Delta''' \rhd \Gamma'' \wedge \Delta''\rangle$ e por ($\wedge$-cons) tem-se que $M.S' : \langle(\Gamma \wedge \Delta''') \wedge \Delta^1 \wedge \cdots \wedge \Delta^m \rhd \wedge_{i=1}^m \sigma_i.(\Gamma'' \wedge \Delta'')\rangle$. Observe que $(\Gamma \wedge \Delta''') \wedge \Delta^1 \wedge \cdots \wedge \Delta^m = \Gamma \wedge (\Delta''' \wedge \Delta^1 \wedge \cdots \wedge \Delta^m) = \Gamma \wedge \Delta$.

8. Por indução na estrutura de S. Suponha que $S\!:\!\langle\Gamma\!\rhd\!\Delta^1 \wedge \Delta^2\rangle$. Note que se $\Delta^j = nil$ para $j\!\in\!\{1,2\}$ então, pelo item 1, $S\!:\!\langle\Gamma^j \rhd \Delta^j\rangle$ para $\Gamma^j\!=\!nil$ e a afirmação é trivial. Consideram-se os casos onde $\Delta^1, \Delta^2\!\neq\! nil$.

   - Se $S \equiv id$, então por (id) tem-se que $\Gamma = \Delta^1 \wedge \Delta^2$. Assim, seja $\Gamma^j = \Delta^j$ para $j\!\in\!\{1,2\}$. Logo por (id) tem-se que $id\!:\!\langle\Delta^j \rhd \Delta^j\rangle$.
   - Seja $S \equiv\uparrow$. Pelo item 6 tem-se que $\Gamma = \omega.(\Delta^1 \wedge \Delta^2)$. Assim, para $\Gamma^1\!=\!\omega.\Delta^1$ e $\Gamma^2\!=\!\omega.\Delta^2$ tem-se o resultado.
   - Seja $S \equiv S_1 \circ S_2$. Por (comp) tem-se que $S_1 : \langle\Gamma' \rhd \Delta^1 \wedge \Delta^2\rangle$ e $S_2 : \langle\Gamma \rhd \Gamma'\rangle$. Por HI tem-se que $\Gamma'\!=\!(\Gamma')^1 \wedge (\Gamma')^2$ tal que $S_1\!:\!\langle(\Gamma')^1 \rhd \Delta^1\rangle$ e $S_1\!:\!\langle(\Gamma')^2 \rhd \Delta^2\rangle$.

Pelo Corolário 1 tem-se que $(\Gamma')^1 \neq (\Gamma'')^1.\omega^{\underline{m}}$ e $(\Gamma')^2 \neq (\Gamma'')^2.\omega^{\underline{m}}$, para qualquer $m > 0$. Assim, por HI tem-se que $\Gamma = \Gamma^1 \wedge \Gamma^2$ tal que $S_2 : \langle \Gamma^1 \rhd (\Gamma')^1 \rangle$ e $S_2 : \langle \Gamma^2 \rhd (\Gamma')^2 \rangle$. Portanto, por (comp), $S_1 \circ S_2 : \langle \Gamma^1 \rhd \Delta^1 \rangle$ e $S_1 \circ S_2 : \langle \Gamma^2 \rhd \Delta^2 \rangle$.

- Seja $S \equiv M.S'$. Suponha s.p.d.g. que $\Delta^1 = \omega.\Delta'$ e que $\Delta^2 = (\wedge_{i=1}^m \sigma_i).\Delta''$. Pelo item 4 tem-se que $S' : \langle \Gamma' \rhd \Delta' \wedge \Delta'' \rangle$ onde $\Gamma = \Gamma' \wedge \Gamma^1 \wedge \cdots \wedge \Gamma^m$ e $\forall 1 \leq i \leq m$, $M : \langle \Gamma^i \vdash \sigma_i \rangle$. Assim, por HI, $\Gamma' = (\Gamma')^1 \wedge (\Gamma')^2$ onde $S' : \langle (\Gamma')^1 \rhd \Delta' \rangle$ e $S' : \langle (\Gamma')^2 \rhd \Delta'' \rangle$. Por ($\omega$-*cons*) tem-se $M.S' : \langle (\Gamma')^1 \rhd \omega.\Delta' \rangle$ e, por ($\wedge$-*cons*), $M.S' : \langle (\Gamma')^2 \wedge \Gamma^1 \wedge \cdots \wedge \Gamma^m \rhd (\wedge_{i=1}^m \sigma_i).\Delta'' \rangle$. Portanto, tomando $\Gamma^1 = (\Gamma')^1$ e $\Gamma^2 = (\Gamma')^2 \wedge \Gamma^1 \wedge \cdots \wedge \Gamma^m$ tem-se o resultado. $\square$

Lema 8.2.5 (Geração para termos em $\lambda\sigma^{\wedge}$):

1. $\underline{1}\,[\uparrow^m] : \langle \Gamma \vdash_{\lambda\sigma^{\wedge}} \tau \rangle$ se, e somente se, $\Gamma = \omega^{\underline{m}}.\tau.nil$.
2. Se $\lambda.M : \langle nil \vdash_{\lambda\sigma^{\wedge}} \tau \rangle$, então $\tau = \omega \to \sigma$ e $M : \langle nil \vdash_{\lambda\sigma^{\wedge}} \sigma \rangle$ ou $\tau = \wedge_{i=1}^n \sigma_i \to \sigma$, $n > 0$, e $M : \langle \wedge_{i=1}^n \sigma_i.nil \vdash_{\lambda\sigma^{\wedge}} \sigma \rangle$ onde $\sigma, \sigma_1, \ldots, \sigma_n \in \mathcal{T}$.
3. Se $\lambda.M : \langle \Gamma \vdash_{\lambda\sigma^{\wedge}} \tau \rangle$ e $|\Gamma| > 0$, então $\tau = u \to \sigma$ para algum $u \in \mathcal{U}$ e $\sigma \in \mathcal{T}$, onde $M : \langle u.\Gamma \vdash_{\lambda\sigma^{\wedge}} \sigma \rangle$.
4. Se $(M_1\ M_2) : \langle \Gamma \vdash_{\lambda\sigma^{\wedge}} \tau \rangle$ então $M_1 : \langle \Gamma \vdash_{\lambda\sigma^{\wedge}} \omega \to \tau \rangle$ ou $M_1 : \langle \Gamma' \vdash_{\lambda\sigma^{\wedge}} \wedge_{i=1}^m \sigma_i \to \tau \rangle$ e $\forall 1 \leq i \leq m$, $N_i : \langle \Gamma^i \vdash_{\lambda\sigma^{\wedge}} \sigma_i \rangle$ onde $\Gamma = \Gamma' \wedge \Gamma^1 \wedge \cdots \wedge \Gamma^m$.

Demonstração. 1. Suponha que $\underline{1}\,[\uparrow^m] : \langle \Gamma \vdash \tau \rangle$. Por (comp) tem-se que $\uparrow : \langle \Gamma \rhd \Gamma' \rangle$ e $\underline{1} : \langle \Gamma' \vdash \tau \rangle$. Assim, por (var) tem-se que $\Gamma' = \tau.nil$ logo, pelo Lema 8.2.4.6, $\Gamma = \omega^{\underline{m}}.\tau.nil$.

2. Por análise de casos na derivação de $\lambda.M : \langle nil \vdash \tau \rangle$.

3. Por análise de casos na derivação de $\lambda.M : \langle \Gamma \vdash \tau \rangle$, para $|\Gamma| > 0$.

4. Por análise de casos na derivação de $(M_1\ M_2) : \langle \Gamma \vdash \tau \rangle$. $\square$

8.2.1 Redução de sujeito

Com os lemas de geração apresentados acima, podemos então estabelecer o propriedade de SR para $\lambda\sigma^{\wedge}$.

Teorema 8.2.6 (SR para $\lambda\sigma^{\wedge}$): Seja M um termo em $\lambda\sigma$. Se $M:\langle\Gamma\vdash_{\lambda\sigma^{\wedge}}\tau\rangle$ e $M\longrightarrow_{\lambda\sigma} M'$ então $M':\langle\Gamma\vdash_{\lambda\sigma^{\wedge}}\tau\rangle$. Em particular, para uma substituição S em $\lambda\sigma$, se $S:\langle\Gamma\rhd_{\lambda\sigma^{\wedge}}\Gamma'\rangle$ e $S\longrightarrow_{\lambda\sigma} S'$ então $S':\langle\Gamma\rhd_{\lambda\sigma^{\wedge}}\Gamma'\rangle$.

Demonstração. A prova é feita através da análise da propriedade em cada regra de reescrita do cálculo.

- *Beta*: Suponha que $(\lambda.M\ N):\langle\Gamma\vdash\tau\rangle$. Pelo Lema 8.2.5.4 tem-se duas possibilidades.

  Suponha que $\lambda.M:\langle\Gamma\vdash\omega\rightarrow\tau\rangle$. Se $\Gamma=nil$, então pelo Lema 8.2.5.2 tem-se que $M:\langle nil\vdash\tau\rangle$. Por (id) e ($nil$-cons), $N.id:\langle nil\rhd nil\rangle$. Assim, por (clos), $M[N.id]:\langle nil\vdash\tau\rangle$. Se $|\Gamma|>0$ então, pelo Lema 8.2.5.3, $M:\langle\omega.\Gamma\vdash\tau\rangle$. Por (id) e ($\omega$-cons), $N.id:\langle\Gamma\rhd\omega.\Gamma\rangle$ logo, por (clos), $M[N.id]:\langle\Gamma\vdash\tau\rangle$.

  Suponha que $\lambda.M:\langle\Gamma'\vdash\wedge_{i=1}^{m}\sigma_i\rightarrow\tau\rangle$ e $\forall 1\leq i\leq m$, $N:\langle\Gamma^i\vdash\sigma_i\rangle$ onde $\Gamma=\Gamma'\wedge\Gamma^1\wedge\cdots\wedge\Gamma^m$. Pelos Lemas 8.2.5.2 e 8.2.5.3, $M:\langle(\wedge_{i=1}^{m}\sigma_i).\Gamma'\vdash\tau\rangle$. Assim, por (id) e ($\wedge$-cons), $N.id:\langle\Gamma'\wedge\Gamma^1\wedge\cdots\wedge\Gamma^m\rhd(\wedge_{i=1}^{m}\sigma_i).\Gamma'\rangle$. Portanto, por (clos), $M[N.id]:\langle\Gamma\vdash\tau\rangle$.

- *App*: Suponha que $(M_1\ M_2)[S]:\langle\Gamma\vdash\tau\rangle$. Por (clos), $S:\langle\Gamma\rhd\Delta\rangle$ e $(M_1\ M_2):\langle\Delta\vdash\tau\rangle$. Pelo Lema 8.2.5.4 tem-se duas possibilidades para a tipagem da aplicação, como descrito no caso acima.

  Suponha que $M_1:\langle\Delta\vdash\omega\rightarrow\tau\rangle$. Por (clos) tem-se que $M_1[S]:\langle\Gamma\vdash\omega\rightarrow\tau\rangle$. Portanto, por $\rightarrow_e^{\omega}$, $(M_1[S]\ M_2[S]):\langle\Gamma\vdash\tau\rangle$.

  Suponha que $M_1:\langle\Delta'\vdash\wedge_{i=1}^{m}\sigma_i\rightarrow\tau\rangle$ e $\forall 1\leq i\leq m$, $M_2:\langle\Delta^i\vdash\sigma_i\rangle$ onde $\Delta=\Delta'\wedge\Delta^1\wedge\cdots\wedge\Delta^m$. Pelo Corolário 8.2.3 tem-se que $\Delta',\Delta^1,\ldots,\Delta^m\neq\Delta'''.\omega^{\underline{n}}$, para quaisquer contexto Δ''' e $n>0$. Assim, por indução em m e o Lema 8.2.4.8 tem-se que $\Gamma=\Gamma'\wedge\Gamma^1\wedge\cdots\wedge\Gamma^m$ tal que $S:\langle\Gamma'\rhd\Delta'\rangle$ e $\forall 1\leq i\leq m$, $S:\langle\Gamma^i\rhd\Delta^i\rangle$. Assim, por (clos), $M_1[S]:\langle\Gamma'\vdash(\wedge_{i=1}^{m}\sigma_i)\rightarrow\tau\rangle$ e $\forall 1\leq i\leq m$, $M_2[S]:\langle\Gamma^i\vdash\sigma_i\rangle$. Portanto, por $\rightarrow_e$, $(M_1[S]\ M_2[S]):\langle\Gamma\vdash\tau\rangle$.

- *Abs*: Suponha que $(\lambda.M)[S]:\langle\Gamma\vdash\tau\rangle$. Por (clos), $S:\langle\Gamma\rhd\Gamma'\rangle$ e $\lambda.M:\langle\Gamma'\vdash\tau\rangle$.

  Se $\Gamma'=nil$ então, pelo Lema 8.2.5.2, $\tau=\omega\rightarrow\tau'$ e $M:\langle nil\vdash\tau'\rangle$ ou $\tau=\wedge_{i=1}^{m}\sigma_i\rightarrow\tau'$ e $M:\langle\wedge_{i=1}^{m}\sigma_i.nil\vdash\tau'\rangle$. Além disso, pelo Lema 8.2.4.5, $\Gamma=nil$. Assim, por (nil-shift) e (comp), $S\circ\uparrow:\langle nil\rhd nil\rangle$. Portanto, para $\tau=\omega\rightarrow\tau'$ tem-se por (nil-cons) que $\underline{1}.(S\circ\uparrow) : \langle nil \rhd nil\rangle$. Logo, por (clos), $M[\underline{1}.(S\circ\uparrow)] : \langle nil \vdash \tau'\rangle$ e por $\to_i'$ tem-se que $\lambda.M[\underline{1}.(S\circ\uparrow)] : \langle nil \vdash \tau\rangle$. Para $\tau = (\wedge_{i=1}^m \sigma_i) \to \tau'$ tem-se por (var) e ($\wedge$-cons) que $\underline{1}.(S\circ\uparrow) : \langle \wedge_{i=1}^m \sigma_i.nil \rhd \wedge_{i=1}^m \sigma_i.nil\rangle$. Assim, por (clos) e $\to_i$ tem-se que $\lambda.M[\underline{1}.(S\circ\uparrow)] : \langle nil \vdash \tau\rangle$.

Se $|\Gamma'|>0$ então pelo Corolário 8.2.3 tem-se que Γ' não é um contexto omega e pelo Lema 8.2.5.3 tem-se que $\tau = u \to \tau'$ tal que $M : \langle u.\Gamma' \vdash \tau'\rangle$. Se $\Gamma = nil$ então por (nil-shift) e (comp), $S\circ\uparrow : \langle nil \rhd \Gamma'\rangle$. Se $u=\omega$ então por (ω-$cons$) tem-se $\underline{1}.(S\circ\uparrow) : \langle nil \rhd \omega.\Gamma'\rangle$ logo, por (clos) e $\to_i'$, $\lambda.M[\underline{1}.(S\circ\uparrow)] : \langle nil \vdash \omega\to\tau'\rangle$. Se $u = \wedge_{i=1}^m \sigma_i$ então por ($\wedge$-cons) tem-se $\underline{1}.(S\circ\uparrow) : \langle \wedge_{i=1}^m \sigma_i.nil \rhd \wedge_{i=1}^m \sigma_i.\Gamma'\rangle$ logo, por (clos) e $\to_i$, $\lambda.M[\underline{1}.(S\circ\uparrow)] : \langle nil \vdash \wedge_{i=1}^m \sigma_i \to \tau'\rangle$. Se $|\Gamma|>0$ então, pelo Corolário 8.2.3, $\Gamma \neq \Delta.\omega^{\underline{m}}$ logo, por (ω-shift) e (comp), $S\circ\uparrow : \langle \omega.\Gamma \rhd \Gamma'\rangle$. Análogo ao caso anterior tem-se por (ω-$cons$) ou ($\wedge$-cons), dependendo de u, que $\underline{1}.(S\circ\uparrow) : \langle u.\Gamma \rhd u.\Gamma'\rangle$. Assim, por (clos) e $\to_i$, $\lambda.M[\underline{1}.(S\circ\uparrow)] : \langle \Gamma \vdash u\to\tau'\rangle$.

- *Clos*: Suponha que $(M[S])[S'] : \langle \Gamma \vdash \tau\rangle$. Por (clos), $S' : \langle \Gamma \rhd \Gamma'\rangle$ e $M[S] : \langle \Gamma' \vdash \tau\rangle$ e, por (clos) novamente, $S : \langle \Gamma' \rhd \Gamma''\rangle$ e $M : \langle \Gamma'' \vdash \tau\rangle$. Portanto, por (comp) tem-se que $S\circ S' : \langle \Gamma \rhd \Gamma''\rangle$ logo por (clos) tem-se que $M[S\circ S'] : \langle \Gamma \vdash \tau\rangle$.

- *VarCons*: Suponha que $\underline{1}[M.S] : \langle \Gamma \vdash \tau\rangle$. Por (clos), $M.S : \langle \Gamma \rhd \Gamma'\rangle$ e $\underline{1} : \langle \Gamma' \vdash \tau\rangle$. Por (var) tem-se que $\Gamma' = \tau.nil$. Assim, pelo Lema 8.2.4.4 tem-se que $S : \langle \Gamma^1 \rhd nil\rangle$ onde $\Gamma = \Gamma^1 \wedge \Gamma^2$ e $M : \langle \Gamma^2 \vdash \tau\rangle$. Pelo Lema 8.2.4.5 tem-se que $\Gamma^1 = nil$ logo $\Gamma^2 = \Gamma$.

- *Id*: Suponha que $M[id] : \langle \Gamma \vdash \tau\rangle$. Por (clos), $id : \langle \Gamma \rhd \Gamma'\rangle$ e $M : \langle \Gamma' \vdash \tau\rangle$. Por (id) tem-se que $\Gamma' = \Gamma$.

- *AssEnv*: Suponha que $(S_1\circ S_2)\circ S_3 : \langle \Gamma \rhd \Gamma'\rangle$. Por (comp) tem-se que $S_3 : \langle \Gamma \rhd \Gamma''\rangle$ e $S_1\circ S_2 : \langle \Gamma'' \rhd \Gamma'\rangle$ e, por (comp) novamente, $S_2 : \langle \Gamma'' \vdash \Gamma'''\rangle$ e $S_1 : \langle \Gamma''' \rhd \Gamma'\rangle$. Portanto, por (comp) tem-se que $S_2\circ S_3 : \langle \Gamma \vdash \Gamma'''\rangle$ logo, por (comp), tem-se que $S_1\circ(S_2\circ S_3) : \langle \Gamma \rhd \Gamma'\rangle$

- *MapEnv*: Suponha que $(M.S)\circ S' : \langle \Gamma \rhd \Gamma'\rangle$. Por (comp) tem-se que $S' : \langle \Gamma \rhd \Gamma''\rangle$ e $M.S : \langle \Gamma'' \rhd \Gamma'\rangle$.

  Se $\Gamma' = nil$ então pelo Lema 8.2.4.2 tem-se que $S : \langle \Gamma'' \rhd nil\rangle$. Assim, por (comp) tem-se $S\circ S' : \langle \Gamma \rhd nil\rangle$ e por (nil-$cons$) tem-se que $M[S'].(S\circ S') : \langle \Gamma \rhd nil\rangle$.

Se $\Gamma' = \omega.\Delta$ então pelo Lema 8.2.4.3 tem-se que $S:\langle \Gamma'' \rhd \Delta\rangle$ onde $\Delta \neq \omega^{\underline{n}}$. Assim, por (comp) tem-se $S \circ S' : \langle \Gamma \rhd \Delta\rangle$ e por (ω-*cons*) tem-se que $M[S'].(S \circ S') : \langle \Gamma \rhd \omega.\Delta\rangle$.

Se $\Gamma' = \wedge_{i=1}^{m} \sigma_i.\Delta$ então pelo Lema 8.2.4.4 tem-se que $S : \langle \Delta''' \rhd \Delta\rangle$ onde $\Gamma'' = \Delta''' \wedge \Delta^1 \wedge \cdots \wedge \Delta^m$ e $\forall 1 \leq i \leq m$, $M : \langle \Delta^i \vdash \sigma_i\rangle$. Pelo Corolário 8.2.3 tem-se que $\Delta''', \Delta^1, \ldots, \Delta^m \neq \Gamma'''.\omega^{\underline{n}}$, para quaisquer contexto Γ''' e $n > 0$. Assim, por indução em m e o Lemma 8.2.4.8 tem-se que $\Gamma = \Gamma''' \wedge \Gamma^1 \wedge \cdots \wedge \Gamma^m$ tal que $S' : \langle \Gamma''' \rhd \Delta'''\rangle$ e $\forall 1 \leq i \leq m$, $S' : \langle \Gamma^i \rhd \Delta^i\rangle$. Assim, por (comp) tem-se $S \circ S' : \langle \Gamma''' \rhd \Delta\rangle$ e por (clos) tem-se que $\forall 1 \leq i \leq m$, $M[S'] : \langle \Gamma^i \vdash \sigma^i\rangle$. Portanto, por ($\wedge$-*cons*) tem-se que $M[S'].(S \circ S') : \langle \Gamma''' \wedge \Gamma^1 \wedge \cdots \wedge \Gamma^m \rhd \wedge_{i=1}^{m} \sigma_i.\Delta\rangle$

- IdL: Suponha que $id \circ S : \langle \Gamma \rhd \Gamma'\rangle$. Por (comp) tem-se que $S : \langle \Gamma \rhd \Gamma''\rangle$ e $id : \langle \Gamma'' \rhd \Gamma'\rangle$. Assim, por (id) tem-se que $\Gamma'' = \Gamma'$.

- IdR: Suponha que $S \circ id : \langle \Gamma \rhd \Gamma'\rangle$. Por (comp) tem-se que $id : \langle \Gamma \rhd \Gamma''\rangle$ e $S : \langle \Gamma'' \rhd \Gamma'\rangle$. Assim, por (id) tem-se que $\Gamma'' = \Gamma$.

- $ShiftCons$: Suponha que $\uparrow \circ (M.S) : \langle \Gamma \rhd \Gamma'\rangle$. Por (comp) tem-se que $M.S : \langle \Gamma \rhd \Gamma''\rangle$ e $\uparrow : \langle \Gamma'' \rhd \Gamma'\rangle$. Pelo Lema 8.2.4.6 tem-se que $\Gamma' = \Gamma'' = nil$ ou $\Gamma'' = \omega.\Gamma'$ onde $\Gamma' \neq \Delta.\omega^{\underline{m}}$.

  Se $\Gamma' = \Gamma'' = nil$ então pelo Lema 8.2.4.2 tem-se que $S : \langle \Gamma \rhd nil\rangle$.

  Se $\Gamma'' = \omega.\Gamma'$ então pelo Lema 8.2.4.3 tem-se que $S : \langle \Gamma \rhd \Gamma'\rangle$.

- $VarShift$: Suponha que $\underline{1}.\uparrow : \langle \Gamma \rhd \Gamma'\rangle$. Se $\Gamma' = nil$ então pelo Lema 8.2.4.5 tem-se que $\Gamma = nil$ logo, por (id), $id : \langle nil \rhd nil\rangle$.

  Se $\Gamma' = \omega.\Gamma''$ então pelo Lema 8.2.4.3 tem-se que $\uparrow : \langle \Gamma \rhd \Gamma''\rangle$ onde $\Gamma'' \neq \omega^{\underline{n}}$. Assim, pelo Lema 8.2.4.6 tem-se que $\Gamma = \omega.\Gamma''$ onde $\Gamma'' \neq \Delta.\omega^{\underline{n}}$ logo, por (id), $id : \langle \Gamma \rhd \Gamma'\rangle$ onde $\Gamma' = \omega.\Gamma'' = \Gamma$.

  Se $\Gamma' = \wedge_{i=1}^{m} \sigma_i.\Gamma''$ então pelo Lema 8.2.4.4 tem-se que $\uparrow : \langle \Gamma''' \vdash \Gamma''\rangle$ onde $\Gamma = \Gamma''' \wedge \Gamma^1 \wedge \cdots \wedge \Gamma^m$ e $\forall 1 \leq i \leq m$, $\underline{1} : \langle \Gamma^i \vdash \sigma_i\rangle$. Por (var) tem-se que $\forall 1 \leq i \leq m$, $\Gamma^i = \sigma_i.nil$. Pelo Lema 8.2.4.6 tem-se que $\Gamma''' = \Gamma'' = nil$ ou $\Gamma''' = \omega.\Gamma''$ para $\Gamma'' \neq \Delta.\omega^{\underline{n}}$. Em ambos os casos tem-se que $\Gamma''' \wedge \Gamma^1 \wedge \cdots \wedge \Gamma^m = \Gamma''' \wedge (\wedge_{i=1}^{m} \sigma_i.nil) = \wedge_{i=1}^{m} \sigma_i.\Gamma''$. Portanto, $id : \langle \Gamma \rhd \Gamma'\rangle$.

- $Scons$: Suponha que $\underline{1}[S].(\uparrow \circ S) : \langle \Gamma \rhd \Gamma'\rangle$. Se $\Gamma' = nil$ então pelo Lema 8.2.4.5 tem-se que $\Gamma = nil$ e, pelo Lema 8.2.4.1, $S : \langle nil \rhd nil\rangle$.

Se $\Gamma' = \omega.\Gamma''$ então pelo Lema 8.2.4.3 tem-se que $\uparrow\circ S : \langle \Gamma \rhd \Gamma'' \rangle$ onde $\Gamma'' \neq \omega^{\underline{n}}$. Por (comp) tem-se que $S : \langle \Gamma \rhd \Delta \rangle$ e $\uparrow : \langle \Delta \rhd \Gamma'' \rangle$. Assim, pelo Lema 8.2.4.6 tem-se $\Delta = \omega.\Gamma''$ portanto $S : \langle \Gamma \rhd \Gamma' \rangle$.

Se $\Gamma' = \wedge_{i=1}^{m} \sigma_i.\Gamma''$ então pelo Lema 8.2.4.4 tem-se que $\uparrow\circ S : \langle \Gamma''' \vdash \Gamma'' \rangle$ onde $\Gamma = \Gamma''' \wedge \Gamma^1 \wedge \cdots \wedge \Gamma^m$ e $\forall 1 \leq i \leq m$, $\underline{1}[S] : \langle \Gamma^i \vdash \sigma_i \rangle$. Por (cons) e (var) tem-se que $\forall 1 \leq i \leq m$, $S : \langle \Gamma^i \rhd \sigma_i.nil \rangle$. Por (comp) tem-se que $S : \langle \Gamma''' \rhd \Delta \rangle$ e $\uparrow : \langle \Delta \rhd \Gamma'' \rangle$. Pelo Lema 8.2.4.6 tem-se que $\Delta = \Gamma'' = nil$ ou $\Delta = \omega.\Gamma''$ onde $\Gamma'' \neq \Delta'.\omega^{\underline{n}}$. Portanto, em ambos os caso tem-se pelo Lema 8.2.4.7 que $S : \langle \Gamma''' \wedge \Gamma^1 \wedge \cdots \wedge \Gamma^m \rhd \wedge_{i=1}^{m} \sigma_i.\Gamma'' \rangle$. $\square$

Capítulo 9

Conclusão e trabalhos futuros

Apresentamos neste trabalho o primeiro sistema de tipos com interseção (IT) para dois cálculos de substituições explíctas (ES), o $\lambda\sigma$ no Capítulo 8 e o λs_e no Capítulo 7, e provamos a propriedade básica de redução de sujeito (SR) para ambos. Os sistemas de IT têm sido estudados como uma alternativa a sistemas de tipos baseados em Hindley/Milner [75], utilizado por exemplo pela implementação do Standard ML [45], para o tratamento do polimorfismo em sistemas de tipos computacionais. O tratamento finitário para o polimorfismo em IT, listando os tipos assumidos por um mesmo procedimento, é conveniente computacionalmente com propriedades como a tipagem principal (PT), que permite atributos como a compilação separada e a recompilação inteligente. A propriedade de relevância em IT é a forma de manter o sistema com o mínimo de suposições possíveis para a inferência do par formado por um tipo e um contexto de tipo, chamado de tipagem. Além disso, sistemas de IT são ferramentas importantes na investigação da semântica do λ-calculus, um estudo que ainda precisa ser realizado tanto para o $\lambda\sigma$ quanto para o λs_e. Os dois cálculos de ES estudados usam uma notação à la de Bruijn, portanto estudamos o λ_{dB}-calculus com dois sistemas de tipos com interseção, apresentados no Capítulo 5.

Resultados

Este trabalho apresentou um sistema de IT para os cálculos de ES estudados, com a proposta de ser uma base para extensões que incluam, *e.g*, a interseção como um construtor geral para tipos, o ω como um tipo universal, uma ordem para tipos e as variáveis de expansão [57]. Estas últimas estão relacionadas com um estudo de PT em sistemas de IT para cálculos de ES. Dessa forma, no Capítulo 4 apresentamos a noção de PT introduzida em [98] para os cálculos λ_{dB}, λs_e e $\lambda\sigma$ com sistemas de tipos simples, provados corretos e completos de acordo à noção de PT apresentada por J. Wells em [103].

O sistema λ_{dB}^{SM}, introduzido em [101] e apresentado na Seção 5.1, é uma versão à la de Bruijn do sistema λ^{SM} de E. Sayag e M. Mauny em [37], com um regra de designação de tipos para índices mais geral. Analisamos a propriedade de SR para o cálculo e provamos a propriedade para a contração β através de um conceito aplicável em sistemas de tipos relevantes, a restrição para contextos.

No Capítulo 6, apresentamos uma caracterização sintática de PT introduzida em [101] para formas β-normais de λ_{dB} no sistema $\lambda_{dB}^{SM_r}$, uma restrição do sistema λ_{dB}^{SM}, seguindo a linha do trabalho de Sayag e Mauny. Além da caracterização sintática de pares principais, similar à de [37], conseguimos uma correspondência bijetiva, módulo renomeamento, com PT para formas β-normais e formalizamos a correspondência do contexto de uma PT e o multiconjunto de índices livres de uma forma β-normal.

O sistema $\lambda_{dB}^{\sqcap}$, introduzido em [100], é apresentado na Seção 5.2, sendo este uma versão à la de Bruijn do sistema $\lambda^{\sqcap}$ de F. Kamareddine e K. Nour em [56]. Apresentamos a prova de SR para a redução β neste sistema, usando uma noção de restrição de contexto mais fraca que a utilizada para o λ_{dB}^{SM}. Os sistemas λ_{dB}^{SM} e $\lambda_{dB}^{SM_r}$ são relevantes como os sistemas originais, enquanto o sistema $\lambda_{dB}^{\sqcap}$ perde a propriedade de relevância presente no sistema $\lambda^{\sqcap}$, permitindo uma forma limitada de redundância. Isso se deve à combinação da presença de ω nos contextos, de forma a prover a estrutura sequencial necessária em sistemas de tipos para o λ_{dB}, e a relação binária para os tipos. A relação de ordem para os tipos induzida por esta relação binária é utilizada no sistema original em [56] para provar a completitude em relação a semântica proposta. Assim, baseamos os sistemas propostos para os cálculos de ES estudados no sistema λ_{dB}^{SM}, de forma a tentar obter um sistema de IT restrito tal que as propriedades de SR e relevância fossem satisfeitas.

No Capítulo 7 introduzimos os sistemas λs^{SM} e $\lambda s_e{}^{\wedge}$ de IT para o λs e λs_e, respectivamente. A análise feita para a adição de IT para o λs_e-calculus apresenta o estudo do sistema para λs como uma etapa preliminar onde, além da propriedade de SR, a relevância do sistema de tipos é considerada. A propriedade de relevância é satisfeita para o sistema

λs^{SM} com relação a uma propriedade sintática bem definida, os índices disponíveis. Por outro lado, uma propriedade relacionada à relevância é satisfeita por $\lambda {s_e}^{\wedge}$, sem uma definição geral para uma relação com características sintáticas. A propriedade de SR para o sistema λs^{SM} é provada para o s-calculus e, com restrição de contexto, para a simulação de contração β. A propriedade de SR é provada para $\lambda {s_e}^{\wedge}$ em relação à redução β.

No Capítulo 8 apresentamos o estudo referente a adição de IT para o $\lambda\sigma$, introduzindo o sistema $\lambda^{\wedge}$. O sistema, a exemplo do sistema de IT para o λs_e, tem uma propriedade relativa à relevância sem uma relação geral com características sintáticas das expressões em $\lambda\sigma$. O sistema tem a propriedade de SR para a redução β.

Trabalhos futuros

Apesar da definição para PT de formas β-normais apresentada no Capítulo 6 ser similar a definição de PT para o sistema de tipos simples $\lambda_{dB}^{\rightarrow}$, a correspondência com a noção geral de PT não é direta. Assim, fica como trabalho futuro demonstrar a correspondência entre a noção de PT no sistema $\lambda_{dB}^{SM_r}$ e a noção geral de PT de Wells.

Para os sistemas de IT para o λ_{dB} baseados no sistema λ^{SM}, podemos trocar $\rightarrow_e'$ do sistema λ_{dB}^{SM} por $\rightarrow_e^{\omega}$ abaixo

$$\frac{M : \langle \Gamma \vdash \omega \rightarrow \tau \rangle}{(M\ N) : \langle \Gamma \vdash \tau \rangle} \rightarrow_e^{\omega}$$

Seja $\lambda_{dB}^{\wedge}$ o sistema obtido com essa troca de regras. Este sistema é uma versão à la de Bruijn do sistema em [38, 39]. Com lemas de geração apropriados, a exemplo dos sistemas para cálculos de ES propostos neste trabalho e que possuem uma regra $\rightarrow_e^{\omega}$, acreditamos na possibilidade de recuperação das propriedades de redução, e expansão, de sujeito. A partir deste resultado, poderemos estender o resultado de PT para termos terminantes (WN), na linha do trabalho realizado originalmente em [38]. Assim, o $\lambda_{dB}^{\wedge}$ e uma restrição no conjunto de tipos, a exemplo de $\mathbb{U}^{+}$ em [56], podem ser utilizados em uma caracterização de WN para o λ_{dB}. Um sistema desta forma seria um primeiro passo para um sistema com variáveis de expansão, formalizando tanto a substituição quanto a expansão para o sistema de tipos correspondente.

Para o λs_e-calculus, fica como trabalho futuro uma possível prova de caracterização para WN através da inferência de tipos em $\lambda {s_e}^{\wedge}$ e a investigação de PT para o cálculo. Para o λs fica a questão de uma possível caracterização de SN através do sistema λs^{SM}.

Podemos ainda investigar sistemas de IT para o $\lambda\upsilon$ de P. Lescanne [70], permitindo uma comparação com o sistema $\lambda\sigma^{\wedge}$ análoga à comparação entre propriedades para os sistemas de λs e λs_e, presentes neste documento. Uma questão interessante é se um sistema de IT para a variante do $\lambda\sigma$ sem a regra $(VarShift)$ tem alguma propriedade diferente do que para a versão analisada no presente trabalho. Além disso, ficam as questões de caracterização de WN para o $\lambda\sigma$ através de $\lambda\sigma^{\wedge}$ e PT em sistemas com IT para $\lambda\sigma$.

Por fim, o uso de sistemas de IT para o estudo da semântica dos cálculos à la de Bruijn, investigados no presente trabalho, devem utilizar como base os sistemas aqui introduzidos. Estes sistemas devem ser estendidos de forma a incluir uma regra de inferência para a introdução da interseção sendo esta tratada como um construtor de tipos mais geral, o ω ser tratado como um tipo universal e a inclusão de uma ordem para tipos com uma regra de inferência associada. Portanto, os sistemas assim obtidos devem estar relacionados ao sistema $\lambda_{dB}^{\sqcap}$, da Seção 5.2.

Referências Bibliográficas

[1] Martín Abadi, Luca Cardelli, Pierre-Louis Curien, and Jean-Jacques Lévy. Explicit substitutions. *Journal of Functional Programming*, 1(4):375–416, 1991.

[2] Fabio Alessi, Franco Barbanera, and Mariangiola Dezani-Ciancaglini. Intersection types and lambda models. *Theoretical Computer Science*, 355(2):108–126, 2006.

[3] Mauricio Ayala-Rincón and Fairouz Kamareddine. Unification via λs_e-style of explicit substitution. *The Logical Journal of the Interest Group in Pure and Applied Logics*, 9(4):489–523, 2001.

[4] Mauricio Ayala-Rincón and César Muñoz. Explicit substitutions and all that. *Colombian Journal of Computation*, 1(1):47–71, 2000.

[5] Franz Baader and Tobias Nipkow. *Term rewriting and all that.* Cambridge University Press, New York, NY, USA, 1998.

[6] Henk Barendregt. *The Lambda Calculus: Its Syntax and Semantics (revised edition)*, volume 103 of *Studies in Logic and the Foundations of Mathematics.* Elsevier Science, Amsterdan, The Netherlands, 1984.

[7] Henk Barendregt. Lambda calculi with types. In *Handbook of logic in computer science (vol. 2): background: computational structures*, pages 117–309. Oxford University Press, New York, NY, USA, 1992.

[8] Henk Barendregt. The impact of the lambda calculus. *Bulletin of Symbolic Logic*, 3(2):181–215, 1997.

[9] Henk Barendregt, Mario Coppo, and Mariangiola Dezani-Ciancaglini. A filter lambda model and the completeness of type assignment. *Bulletin of Symbolic Logic*, 48:931–940, 1983.

[10] Yves Bertot and Pierre Castéran. *Interactive Theorem Proving and Program Development. Coq'Art: The Calculus of Inductive Constructions.* Texts in Theoretical Computer Science. An EATCS Series. Springer, 2004.

[11] Roel Bloo and Kristoffer H. Rose. Preservation of strong normalization in named lambda calculi with explicit substitution and garbage collection. In *CSN-95: Computer Science in the Netherlands*, pages 62–72, 1995.

[12] Eduardo Bonelli. The polymorphic lambda calculus with explicit substitutions. In *International Workshop on Explicit Substitutions: Theory and Applications to Programs and Proofs. Trento, Italy*, 1999.

[13] Eduardo Bonelli. Perpetuality in a named lambda calculus with explicit substitutions. *Mathematical Structures in Computer Science*, 11(1):47–90, 2001.

[14] Peter Borovanskỳ. Implementation of higher-order unification based on calculus of explicit substitution. In *SOFSEM '95: Proceedings of the 22nd Seminar on Current Trends in Theory and Practice of Informatics*, pages 363–368, London, UK, 1995. Springer-Verlag.

[15] Sébastien Carlier and J. B. Wells. Type inference with expansion variables and intersection types in system E and an exact correspondence with β-reduction. In *PPDP '04: Proceedings of the 6th ACM SIGPLAN international conference on Principles and practice of declarative programming*, pages 132–143, New York, NY, USA, 2004. ACM.

[16] Alonzo Church. A set of postulates for the foundation of logic. *Annals of Mathematics*, 33(2):346–366, 1932.

[17] Alonzo Church. An unsolvable problem of elementary number theory. *American Journal of Mathematics*, 58:354–363, 1936.

[18] Alonzo Church. A formulation of the simple theory of types. *Journal of Symbolic Logic*, 5:56–68, 1940.

[19] Alonzo Church and Barkley Rosser. Some properties of conversion. *Transactions of the American Mathematical Society*, 39:472–482, 1936.

[20] R. L. Constable, S. F. Allen, H. M. Bromley, W. R. Cleaveland, J. F. Cremer, R. W. Harper, D. J. Howe, T. B. Knoblock, N. P. Mendler, P. Panangaden, J. T. Sasaki, and S. F. Smith. *Implementing mathematics with the Nuprl proof development system*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1986.

[21] Mario Coppo and Mariangiola Dezani-Ciancaglini. A new type-assignment for lambda terms. *Archiv für Mathematische Logik und Grundlagenforschung*, 19:139–156, 1978.

[22] Mario Coppo and Mariangiola Dezani-Ciancaglini. An extension of the basic functionality theory for the λ-calculus. *Notre Dame, Journal of Formal Logic*, 21:685–693, 1980.

[23] Mario Coppo, Mariangiola Dezani-Ciancaglini, and Patrick Sallé. Functional characterization of some semantic equalities inside lambda-calculus. In *ICALP'79: Proceedings of the 6th Colloquium, on Automata, Languages and Programming*, volume 71 of *LNCS*, pages 133–146, London, UK, 1979. Springer-Verlag.

[24] Mario Coppo, Mariangiola Dezani-Ciancaglini, and Betti Venneri. Principal type schemes and λ-calculus semantics. In *J.P. Seldin and J.R. Hindley, editors, To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 536–560. Academic Press, 1980.

[25] Mario Coppo, Mariangiola Dezani-Ciancaglini, and Betti Venneri. Functional characters of solvable terms. *Zeitschrift für mathematiche Logik und Grundlagen der Mathematik*, 27:45–58, 1981.

[26] Pierre-Louis Curien, Thérèse Hardin, and Jean-Jacques Lévy. Confluence properties of weak and strong calculi of explicit substitutions. *Journal of the ACM*, 43(2):362–397, 1996.

[27] Haskell Curry and Robert Feys. *Combinatory Logic volume I.* North-Holland Co., Netherlands, 1958.

[28] Luis Damas and Robin Milner. Principal type-schemes for functional programs. In *POPL '82: Proceedings of the 9th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 207–212, New York, NY, USA, 1982. ACM.

[29] Ferruccio Damiani and Paola Giannini. A decidable intersection type system based on relevance. In *TACS'94*, LNCS 789, pages 707–725. Springer, 1994.

[30] N. G. de Bruijn. Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation, with application to the Church-Rosser theorem. *Indagationes Mathematicae*, 34:381–392, 1972.

[31] N. G. de Bruijn. A namefree lambda calculus with facilities for internal definition of expressions and segments. Technical Report T.H.-Report 78-WSK-03, Technische Hogeschool Eindhoven, Nederland, 1978.

[32] Flávio L. C. de Moura, Mauricio Ayala-Rincón, and Fairouz Kamareddine. Higher-order unification: A structural relation between Huet's method and the one based on explicit substitutions. *Journal of Applied Logic*, 6(1):72–108, 2008.

[33] Gilles Dowek, Thérèse Hardin, and Claude Kirchner. Higher-order unification via explicit substitutions. In *LICS '95: Proceedings of the 10th Annual IEEE Symposium on Logic in Computer Science*, page 366, Washington, DC, USA, 1995. IEEE Computer Society.

[34] Gilles Dowek, Thérèse Hardin, and Claude Kirchner. Higher-order unification via explicit substitutions. *Information and Computation*, 157(1/2):183–235, 2000.

[35] Roy Dyckhoff and Christian Urban. Strong normalization of herbelin's explicit substitution calculus with substitution propagation. *Journal of Logic and Computation*, 13(5):689–706, 2003.

[36] H.-D. Ebbinghaus, J. Flum, and W. Thomas. *Mathematical Logic*. Undergraduate texts in Mathematics. Springer, Berlin, second edition, 1994.

[37] Émilie Sayag and Michel Mauny. Characterization of principal types of normal forms in an intersection type system. In *Proceedings of Foundations of Software Technology and Theoretical Computer Science*, volume 1180, pages 335–346. Springer Verlag, 1996.

[38] Émilie Sayag and Michel Mauny. A presentation of the intersection type discipline through principal typings of normal forms. Technical Report RR-2998, INRIA, 1996.

[39] Émilie Sayag and Michel Mauny. Structural properties of intersection types. In *Proceedings of the 8th International Conference on Logic and Computer Science – Theoretical Foundations of Computing (LIRA)*, pages 167–175, Novi Sad, Yugoslavia, September 1997.

[40] Maria C. F. Ferreira, Delia Kesner, and Laurence Puel. Lambda-calculi with explicit substitutions preserving strong normalization. *Applicable Algebra in Engineering, Communication and Computing.*, 9(4):333–371, 1999.

[41] Gottlob Frege. *Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens.* Louis Nebert, Halle, Germany, 1879.

[42] Gottlob Frege. *Grundgesetze der Arithmetik, volume I.* Verlag Hermann Pohle, Jena, Germany, 1892.

[43] Jean-Yves Girard. *Interprétation fonctionelle et élimination des coupures de l'arithmétique d'ordre supérieur.* These d'Etat, Université Paris VII, 1972.

[44] Kurt Gödel. *On Undecidable Propositions of Formal Mathematical Systems(lecture notes taken by Kleene and Rosser at the Institute for Advanced Study).* Davis, 1965.

[45] M. Gordon, R. Milner, L. Morris, M. Newey, and C. Wadsworth. A metalanguage for interactive proof in LCF. In *POPL '78: Proceedings of the 5th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*, pages 119–130, New York, NY, USA, 1978. ACM.

[46] Jean Goubault-Larrecq. Lambda-calcul, logique et machines. Online notes of course delivered at École Normale Supérieure de Cachan, 2001.

[47] Bruno Guillaume. The λse-calculus does not preserve strong normalisation. *Journal of Functional Programming*, 10(4):321–325, July 2000.

[48] Hugo Herbelin. A lambda-calculus structure isomorphic to Gentzen-style sequent calculus structure. In *CSL '94: Selected Papers from the 8th International Workshop on Computer Science Logic*, pages 61–75, London, UK, 1995. Springer-Verlag.

[49] J. Roger Hindley. *Basic Simple Type Theory.* Cambridge University Press, Cambridge, UK, 1997.

[50] Furio Honsell and Marina Lenisa. Semantical analysis of perpetual strategies in lambda-calculus. *Theoretical Computer Science*, 212(1–2):183–209, 1999.

[51] Gérard P. Huet. Unification in typed lambda calculus. In *Lambda-Calculus and Computer Science Theory, Proceedings of the Symposium Held in Rome, March 25-27, 1975*, volume 37 of *LNCS*, pages 192–212. Springer, 1975.

[52] Trevor Jim. What are principal typings and what are they good for? In *POPL*, pages 42–53, 1996.

[53] Fairouz Kamareddine, editor. *Thirty Five Years of Automating Mathematics: A Volume Dedicated to De Bruijn's Automath*, volume 28 of *Applied Logic Series*. Kluwer Academic Publishers, Norwell, MA, USA, 2004.

[54] Fairouz Kamareddine, Twan Laan, and Rob Nederpelt. Automath and pure type systems. *Electronic Notes in Theoretical Computer Science*, 85(7), 2003.

[55] Fairouz Kamareddine, Twan Laan, and Rob Nederpelt. *A Modern Perspective on Type Theory - From its Origins until Today*, volume 29 of *Applied Logic Series*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2004.

[56] Fairouz Kamareddine and Karim Nour. A completeness result for a realisability semantics for an intersection type system. *Annals of Pure and Applied Logic*, 146(2-3):180–198, 2007.

[57] Fairouz Kamareddine, Karim Nour, Vincent Rahli, and J. B. Wells. A complete realisability semantics for intersection types and arbitrary expansion variables. In *Theoretical Aspects of Computing - ICTAC 2008, 5th International Colloquium, Istanbul, Turkey, September 1-3, 2008. Proceedings*, volume 5160 of *LNCS*, pages 171–185. Springer, 2008.

[58] Fairouz Kamareddine and Alejandro Ríos. A lambda-calculus à la de Bruijn with explicit substitutions. In *PLILP '95: Proceedings of the 7th International Symposium on Programming Languages: Implementations, Logics and Programs*, pages 45–62, New York, USA, 1995. Springer.

[59] Fairouz Kamareddine and Alejandro Ríos. Extending a lambda-calculus with explicit substitution which preserves strong normalisation into a confluent calculus on open terms. *Journal of Functional Programming*, 7(4):395–420, 1997.

[60] Fairouz Kamareddine and Alejandro Ríos. Relating the $\lambda\sigma$- and λs-styles of explicit substitutions. *Journal of Logic and Computation*, 10(3):349–380, 2000.

[61] Fairouz Kamareddine and Alejandro Ríos. Pure type systems with de Bruijn indices. *The Computer Journal*, 45(2):187–201, 2002.

[62] Delia Kesner. The theory of calculi with explicit substitutions revisited. In *Computer Science Logic, 21st International Workshop, CSL 2007, 16th Annual Conference of the EACSL, Lausanne, Switzerland, September 11-15, 2007*, volume 4646 of *LNCS*, pages 238–252, 2007.

[63] Delia Kesner. A theory of explicit substitutions with safe and full composition. *Logical Methods in Computer Science*, 5(3), 2009.

[64] Assaf J. Kfoury and J. B. Wells. Principality and type inference for intersection types using expansion variables. *Theoretical Computer Science*, 311(1-3):1–70, 2004.

[65] Kentaro Kikuchi. Simple proofs of characterizing strong normalization for explicit substitution calculi. In *Term Rewriting and Applications, 18th International Conference, RTA 2007, Paris, France, June 26-28, 2007*, volume 4533 of *LNCS*, pages 257–272. Springer, 2007.

[66] Stephen Kleene. Lambda-definability and recursiveness. *Duke Mathematical Journal*, 2:340–353, 1936.

[67] Stephen Kleene and Barkley Rosser. The inconsistency of certain formal logics. *Annals of Mathematics*, 36(2):630–636, 1935.

[68] Jean-Louis Krivine. *Lambda-calculus, types and models.* Ellis Horwood, 1993.

[69] Stéphane Lengrand, Pierre Lescanne, Dan Dougherty, Mariangiola Dezani-Ciancaglini, and Steffen van Bakel. Intersection types for explicit substitutions. *Information and Computation*, 189:17–42, 2004.

[70] Pierre Lescanne. From $\lambda\sigma$ to $\lambda\upsilon$: a journey through calculi of explicit substitutions. In *POPL '94: Proceedings of the 21st ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 60–69, New York, USA, 1994. ACM.

[71] Rafael D. Lins. A new formula for the execution of categorical combinators. In *8th International Conference on Automated Deduction*, volume 230 of *LNCS*, pages 89–98. Springer Berlin / Heidelberg, 1986.

[72] Michel Mauny. *Compilation des langages fonctionnels dans les combinateurs catégoriques. Application au langage ML.* Thèse de Doctorat d'Université, Université Paris VII, 1985.

[73] John McCarthy. Recursive functions of symbolic expressions and their computation by machine, part I. *Communications of the ACM*, 3(4):184–195, 1960.

[74] Paul-André Melliès. Typed lambda-calculi with explicit substitutions may not terminate. In *TLCA '95: Proceedings of the Second International Conference on Typed Lambda Calculi and Applications*, LNCS, pages 328–334, London, UK, 1995. Springer-Verlag.

[75] Robin Milner. A theory of type polymorphism in programming. *Journal of Computer and System Sciences*, 17:348–375, 1978.

[76] César Muñoz. A left-linear variant of $\lambda\sigma$. In *Proc. International Conference PLILP/ALP/HOA'97*, volume 1298 of *LNCS*, pages 224–239, Southampton (England), September 1997. Springer. Also available as INRIA Technical Report RR-3107.

[77] Rob P. Nederpelt, J. H. Geuvers, and R. C. de Vrijer. *Selected Papers on Automath*, volume 133 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, Amsterdan, 1994.

[78] Tobias Nipkow, Lawrence C. Paulson, and Markus Wenzel. *Isabelle/HOL - A Proof Assistant for Higher-Order Logic*, volume 2283 of *LNCS*. Springer, 2002.

[79] Sam Owre and Natarajan Shankar. The formal semantics of PVS. Technical Report NASA/CR-1999-209321, NASA STI, 1999.

[80] Benjamin C. Pierce. *Types and Programming Languages*. The MIT Press, Cambridge, Massachusetts USA, 2002.

[81] Garrel Pottinger. A type assignment for the strongly normalizable λ-terms. In *J.P. Seldin and J.R. Hindley, editors, To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 561–578. Academic Press, 1980.

[82] J. C. Reynolds. Towards a theory of type structure. In *Programming Symposium, Proceedings Colloque sur la Programmation*, volume 19 of *LNCS*, pages 408–423, London, UK, 1974. Springer-Verlag.

[83] Alejandro Ríos. *Contributions à l'étude des λ-calculs avec des substitutions explicites.* Thèse de Doctorat d'Université, Université Paris VII, 1993.

[84] Eike Ritter. Characterising explicit substitutions which preserve termination. In *TLCA '99: Proceedings of the 4th International Conference on Typed Lambda Calculi and Applications*, pages 325–339, London, UK, 1999. Springer-Verlag.

[85] John Alan Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM*, 12(1):23–41, 1965.

[86] Simona Ronchi Della Rocca. Principal type scheme and unification for intersection type discipline. *Theoretical Computer Science*, 59:1–29, 1988.

[87] Simona Ronchi Della Rocca and Betti Venneri. Principal type scheme for an extended type theory. *Theoretical Computer Science*, 28:151–169, 1984.

[88] Kristoffer Høgsbro Rose. Explicit cyclic substitutions. In *CTRS '92: Proceedings of the Third International Workshop on Conditional Term Rewriting Systems*, pages 36–50, London, UK, 1993. Springer-Verlag.

[89] Bertrand Russell. Mathematical logic as based on the theory of types. *American Journal of Mathematics*, 30:222–262, 1908.

[90] Patrick Sallé. Une extension de la théorie des types en λ-calcul. In *Fifth International Conference on Automata, Languages and Programming*, volume 62, pages 398–410. Springer Berlin / Heidelberg, 1978.

[91] Morten Heine Sørensen and Pawel Urzyczyn. *Lectures on the Curry-Howard Isomorphism*, volume 149 of *Studies in Logic and the Foundations of Mathematics.* Elsevier, New York, NY, USA, 2006.

[92] Alan Turing. On computable numbers, with an application to the Entscheidungsproblem. In *Proceedings of the London Mathematical Society*, volume 42(2), pages 230–265, 1936.

[93] Alan Turing. Computability and lambda-definability. *Journal of Symbolic Logic*, 2:153–163, 1937.

[94] Steffen van Bakel. Principal type schemes for the strict type assignment system. *Journal of Logic and Computation*, 3(6):643–670, 1993.

[95] Steffen van Bakel. Intersection type assignment systems. *Theoretical Computer Science*, 151(2):385–435, 1995.

[96] Lambert S. van Benthem Jutting. *Checking Landau's "Grundlagen" in the AUTOMATH System*. PhD thesis, Technische Universiteit Eindhoven, 1977.

[97] Daniel L. Ventura. Cálculos de substituições explícitas que preservam a propriedade de redução de sujeito. Master's thesis, Universidade de Brasília, fevereiro 2006.

[98] Daniel L. Ventura, Mauricio Ayala-Rincón, and Fairouz Kamareddine. Principal typings for explicit substitutions calculi. In *Logic and Theory of Algorithms, 4th Conference on Computability in Europe, CiE 2008, Athens, Greece, June 15-20, 2008, Proceedings*, volume 5028 of *LNCS*, pages 567–578. Springer, 2008.

[99] Daniel L. Ventura, Mauricio Ayala-Rincón, and Fairouz Kamareddine. Explicit substitutions calculi with one step eta-reduction decided explicitly. *Logic Journal of the IGPL*, 17(6):697–718, 2009.

[100] Daniel L. Ventura, Mauricio Ayala-Rincón, and Fairouz Kamareddine. Intersection type system with de Bruijn indices. In *The Many Sides of Logic*, volume 21 of *Studies in Logic*, pages 557–576. College Publication, 2009.

[101] Daniel L. Ventura, Mauricio Ayala-Rincón, and Fairouz Kamareddine. Principal typings in a restricted intersection type system for beta normal forms with de Bruijn indices. In *9th International Workshop on Reduction Strategies in Rewriting and Programming, Brasília, Brazil, 2009*, volume 15 of *EPTCS*, pages 69–82, 2010.

[102] J. B. Wells. Typability and type checking in the second-order lambda-calculus are equivalent and undecidable. In *Proceedings of the 9th Annual IEEE Symposium on Logic in Computer Science*, pages 176–185, San Antonio, Texas, USA, 1994.

[103] J. B. Wells. The essence of principal typings. In *LNCS: Proceedings of the 29th International Colloquium on Automata, Languages and Programming*, volume 2380, pages 913–925, London, UK, 2002. Springer-Verlag.

[104] Ernst Zermelo. Untersuchungen über die grundlagen der mengenlehre. I. *Mathematische Annalen*, 65:261–281, 1908.

Índice Remissivo