Habitantes principais normais em TA$_\lambda$

Sabine Broda, Luíis Damas

Technical Report Series: DCC-00-04

Departamento de Ciência de Computadores – Faculdade de Ciências

&

Laboratório de Inteligência Artificial e Ciência de Computadores

Universidade do Porto

Rua do Campo Alegre, 823 4150 Porto, Portugal

Tel: +351+2+6078830 – Fax: +351+2+6003654

http://www.ncc.up.pt/fcup/DCC/Pubs/treports.html

Habitantes principais normais em $\mathbf{TA}_\lambda$

Sabine Broda, Luís Damas
DCC & LIACC, Universidade do Porto
R. do Campo Alegre 823, 4150-180 Porto, Portugal
Phone: 351 2 6078830, Fax: 351 2 6003654
e-mail: {sbb,luis}@ncc.up.pt

Resumo

Neste trabalho estudamos o problema de contar e enumerar efectivamente os λ-termos em forma normal-β que são habitantes principais de um dado tipo no λ-calculus simplesmente tipado. Apresentamos resultados e algoritmos para resolver este problema respectivamente no λ-calculus completo e nos seus três subsistemas principais. Apresentamos ainda um resultado que permite adaptar para o λI-calculus o algoritmo definido por Ben-Yelles em [6] para contar e enumerar habitantes normais. Com este conjunto de resultados resolvemos cinco problemas em aberto nesta área, ainda recentemente referidos por Hindley em [16].

1 Introdução

Versões tipadas de sistemas de λ-calculus têm sido estudadas desde o seu aparecimento no início deste século em [10] e [8] devido à sua importância para diversas áreas da lógica matemática e ultimamente também para a ciência de computadores. Uma das sub-áreas onde encontra aplicação directa é a programação funcional (e.g. as linguagens ML, Miranda ou Haskell), um exemplo paradigmático duma área na ciência de computadores onde a interligação estreita entre teoria e prática levou a um crescimento frutuoso de ambas: os desenvolvimentos na teoria levam a aplicações práticas directas e as necessidades da implementação suscitam novas abordagens teóricas (cf. [25], [24]). De facto, a programação funcional baseia-se na representação de funções computáveis/algoritmos por λ-termos e a execução de programas corresponde à redução de λ-termos. Uma atribuição de tipos a λ-termos fornece então uma especificação parcial dos algoritmos que são representados, e serve para mostrar a correcção parcial dos programas definidos. Por outro lado, usa-se os tipos também para melhorar a eficiência da compilação de programas funcionais (termos), identificando e implementando de maneira adequada, por exemplo, partes (sub-termos) com tipo especial (e.g. aritmético). Na base de praticamente todos os sistemas utilizados encontra-se o sistema $\mathbf{TA}_\lambda$ o que justifica a atenção que tem recebido no meio científico. Apresentamos aqui alguns resultados sobre habitantes (principais) normais em $\mathbf{TA}_\lambda$, que nos permitem resolver cinco problemas em aberto nesta área, ainda recentemente referidos por Hindley em [16] (cf. pág. 127). O trabalho apresentado corresponde essencialmente ao exposto em [2], [3] e [4].

Se um tipo τ pode ser inferido para um λ-termo M, então dizemos que M é um habitante de τ, e chamamos-lhe ainda normal ou principal se estiver em forma normal-β ou se τ é um tipo principal para M. Em [6], Ben-Yelles apresentou um algoritmo, também descrito em [16], que dado um tipo τ determina o número de λ-termos em forma normal-β que são habitantes de τ em $\mathbf{TA}_\lambda$. Para cada tipo τ o algoritmo decide num número finito de passos, se o número de habitantes normais de τ é finito ou infinito, determina este número no caso de ser finito, e produz e enumera todos os termos relevantes em ambos os casos. Outro algoritmo de contagem, ainda para a mesma questão mas baseado em pontos fixos de equações polinomiais, foi estudado por Zaionc e apresentado em [29] e [30]. Hirokawa mostrou em [18] que o problema, de decidir se o número de habitantes normais de um tipo é infinito ou não, é completo para PSPACE. Relacionados com este problema, estão os problemas, de contar os habitantes normais principais de um tipo τ tanto no λ-calculus completo como nos seus sub-sistemas principais, nomeadamente no λI-calculus, no **BCK**-calculus e no **BCI**-calculus. Apresentamos neste trabalho algoritmos adequados para os quatro casos, bem como um resultado que permite adaptar o algoritmo de Ben-Yelles para contar e enumerar habitantes normais (não necessariamente principais) no λI-calculus.

Na secção 2 introduzimos os conceitos e resultados básicos do λ-calculus simplesmente tipado que serão usados no resto do trabalho. Na secção 3 apresentamos um algoritmo que decide se um dado par $(\mathcal{B}, \alpha)$ é ou não um par principal para algum **BCK**-termo M em forma normal-β. Se este for o caso, então o algoritmo constrói ainda um termo M' com $M' =_\alpha M$. Concluímos da correcção do algoritmo e da unicidade do seu resultado, que todo o par $(\mathcal{B}, \tau)$ (e em particular todo o tipo τ) tem quando muito um habitante principal normal no **BCK**-calculus e que este é calculado pelo algoritmo. Como os **BCI**-termos são também **BCK**-termos, podemos ainda usar o mesmo algoritmo para decidir se um par tem um habitante principal normal no **BCI**-calculus, calculando-o se for o caso.

Analogamente ao algoritmo de Ben-Yelles para habitantes normais, os algoritmos que apresentamos nas secções 4 e 5 para contar e enumerar os habitantes normais principais de um tipo τ baseiam-se nos factos seguintes. Primeiro, podemos concentrar a procura num tipo especial de habitantes de τ que chamamos longos. Segundo, existem inteiros $0 < \mathbf{d}(\tau) < \mathbf{D}(\tau)$ tais que a cardinalidade do conjunto de habitantes principais normais de τ depende directamente do número de habitantes principais normais longos de τ com profundidade em $[0; \mathbf{d}(\tau)[$ e do número dos que têm profundidade em $[\mathbf{d}(\tau); \mathbf{D}(\tau)[$, onde a profundidade representa uma medida sobre a estrutura de um λ-termo em forma normal-β. Finalmente, Ben-Yelles definiu em [6] um algoritmo de procura (outros são apresentados em [27], [28] e [26]), que para cada tipo τ e cada inteiro não negativo d calcula todos os habitantes normais longos de τ com profundidade menor que ou igual a d, que são em número finito. Assim, basta usar qualquer algoritmo para verificar tipos principais (por exemplo o descrito em [16]) para calcular os habitantes principais normais longos de τ com profundidade menor que ou igual a d. Consequentemente, podemos resolver o problema de contar habitantes principais normais de um tipo τ, calculando os habitantes principais normais longos de τ com profundidade menor que $\mathbf{D}(\tau)$.

A complexidade, por vezes escondida na regra da unificação, de qualquer algoritmo (geral) para calcular pares ou tipos principais dificulta a obtenção de resultados nesta área. Acreditamos por isso que é um bom princípio o de considerar adaptações ou restrições adequadas para cada propósito. Assim baseamos a definição e (provas de) correcção dos limites encontrados para os algoritmos nas secções 4 e 5 numa caracterização simples de deduções principais, obtida para o caso particular dos habitantes normais longos a partir do sistema $\mathbf{TA}_{\mathbf{pln}}$, que desenhamos propositadamente para este fim. Da mesma forma, só conseguimos definir o algoritmo para o **BCK**- e o **BCI**-calculus, na secção 3, depois de considerar uma particularização, para o **BCK**-calculus, do algoritmo generalizado para calcular pares principais de λ-termos.

2 Preliminares

Nesta secção introduzimos os conceitos básicos do λ-calculus tipado, necessários para a leitura deste trabalho. Exposições mais completas podem ser encontradas em [5] e [16]. Começamos por apresentar o λ-calculus puro, especificando o seu conjunto de λ-termos.

Definição 2.1 *Seja $\mathcal{V}$ um conjunto numerável de variáveis. O conjunto Λ de λ-termos é o menor conjunto satisfazendo as condições seguintes:*

- $\mathcal{V} \subseteq \Lambda$;
- $M \in \Lambda,\ x \in \mathcal{V} \quad \Rightarrow \quad (\lambda x M) \in \Lambda$ *(abstracção);*
- $M, N \in \Lambda \quad \Rightarrow \quad (MN) \in \Lambda$ *(aplicação).*

Para não sobrecarregar a notação, utilizam-se as abreviações $\lambda x_1 \ldots x_n.M$ e $MN_1 \ldots N_n$ para $(\lambda x_1(\lambda x_2(\ldots(\lambda x_n M)\ldots)))$ e $(\ldots((MN_1)N_2)\ldots N_n)$ respectivamente.

Definição 2.2 *O conjunto $FV(M)$ das variáveis livres de um termo $M \in \Lambda$ é indutivamente definido por:*

- $FV(x) = \{x\}$;
- $FV(\lambda x.M) = FV(M) \setminus \{x\}$;
- $FV(MN) = FV(M) \cup FV(N)$.

Um termo diz-se fechado se e só se $FV(M) = \emptyset$. Uma ocorrência que não é livre de uma variável x num termo chama-se uma ocorrência ligada de x.

O resultado da substituição de todas as ocorrências livres de uma variável x por um λ-termo N em M denota-se por $M[N/x]$. Como habitualmente, essa substituição só é permitida, e nesse caso diz-se que x é substituível por N em M, se x não ocorrer livre em nenhum sub-termo de M da forma $\lambda y.P$, tal que $y \in FV(N)$.

Daqui em diante, consideremos idênticos os λ-termos que só diferem no nome das variáveis ligadas, o que se designa por conversão-α. Assim é possível substituir uma variável x por qualquer termo N em M, mudando, quando necessário, os nomes de algumas variáveis ligadas em M. Dizemos que um termo M tem conflitos de variáveis ligadas se M contém uma abstração $\underline{\lambda x}$ e uma ocorrência de x que não está no alcance desta abstracção. Note-se que para cada λ-termo M existe um λ-termo $N =_\alpha M$ sem conflitos de variáveis ligadas. Neste trabalho vamos considerar termos sem conflitos de variáveis ligadas.

Definição 2.3 *Um λ-termo da forma $(\lambda x.P)Q$ chama-se um β-redex cujo contractum é $P[Q/x]$. Um termo M reduz-se num passo de redução-β para N, e escreve-se $M \rightarrow_{1\beta} N$, se N pode ser obtido a partir de M por substituição de um β-redex em M pelo seu contractum. A relação $\rightarrow_\beta$ é o fecho reflexivo e transitivo de $\rightarrow_{1\beta}$ e $=_\beta$ é o fecho reflexivo, simétrico e transitivo de $\rightarrow_{1\beta}$.*

Diz-se que um λ-termo $M \in \Lambda$ está em forma normal-β, ou simplesmente em forma normal, se e só se não contém qualquer β-redex.

2.1 Sub-sistemas

Restringindo o conjunto de λ-termos define-se diversos sub-sistemas do λ-calculus puro, dos quais apresentamos três, nomeadamente o $\lambda\mathcal{I}$-calculus, o **BCK**-calculus e o **BCI**-calculus. Estes três sistemas podem ser obtidos a partir do λ-calculus puro, impondo condições simples sobre o número de ocorrências das variáveis nos termos e relacionam-se directamente com três sistemas de cálculo proposicional conhecidos, o que torna o seu estudo particularmente interessante.

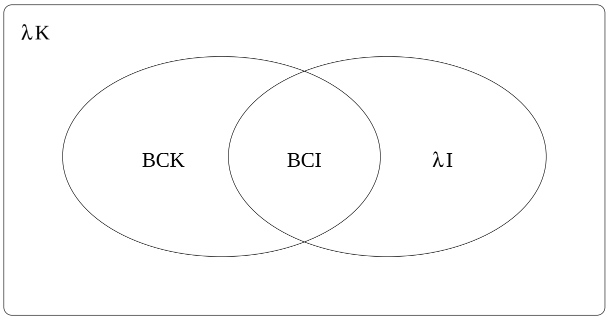

Figura 1:

No $\lambda\mathcal{I}$-calculus cada sub-termo da forma $\lambda x.N$ de um termo M tem pelo menos uma ocorrência da variável x em N, excluindo assim termos como $\lambda xy.x$.

Definição 2.4 ($\lambda\mathcal{I}$-calculus) *O conjunto de $\lambda\mathcal{I}$-termos $\Lambda_{\mathcal{I}}$ é definido por:*

- $\mathcal{V} \subseteq \Lambda_{\mathcal{I}}$;
- $M \in \Lambda_{\mathcal{I}},\ x \in FV(M) \quad \Rightarrow \quad (\lambda x M) \in \Lambda_{\mathcal{I}}$;
- $M, N \in \Lambda_{\mathcal{I}} \quad \Rightarrow \quad (MN) \in \Lambda_{\mathcal{I}}$.

No **BCK**-calculus, nenhuma variável x ocorre mais do que uma vez livre em nenhum sub-termo de um termo M.

Definição 2.5 (BCK-calculus) *O conjunto de* **BCK**-*termos $\Lambda_{\mathcal{A}}$ é definido por:*

- $\mathcal{V} \subseteq \Lambda_{\mathcal{A}}$;
- $M \in \Lambda_{\mathcal{A}},\ x \in \mathcal{V} \quad \Rightarrow \quad (\lambda x M) \in \Lambda_{\mathcal{A}}$;
- $M, N \in \Lambda_{\mathcal{A}},\ FV(M) \cap FV(N) = \emptyset \quad \Rightarrow \quad (MN) \in \Lambda_{\mathcal{A}}$.

No **BCI**-calculus, cada variável ocorre no máximo uma vez livre em todo o sub-termo de um termo M, e para cada sub-termo da forma $\lambda x.N$ de M, a variável x tem exactamente uma ocorrência livre em N.

Definição 2.6 (BCI-calculus) *O conjunto de* **BCI**-*termos $\Lambda_{\mathcal{L}}$ é definido por:*

- $\mathcal{V} \subseteq \Lambda_{\mathcal{L}}$;
- $M \in \Lambda_{\mathcal{L}},\ x \in FV(M) \quad \Rightarrow \quad (\lambda x M) \in \Lambda_{\mathcal{L}}$;
- $M, N \in \Lambda_{\mathcal{L}},\ FV(M) \cap FV(N) = \emptyset \quad \Rightarrow \quad (MN) \in \Lambda_{\mathcal{L}}$.

As restantes noções usadas para o λ-calculus puro definem-se de forma análoga nos três sub-sistemas. Frequentemente usaremos também a notação $\Lambda_{\mathcal{K}}$ ou $\lambda\mathcal{K}$-calculus em vez de Λ e λ-calculus puro. A relação entre os quatro (sub-)sistemas está ilustrada na figura 1.

2.2 Inferência de tipos

Nesta sub-secção descrevemos um sistema de inferência de tipos para λ-termos em forma normal-β no λ-calculus simplesmente tipado à la Curry, começando por descrever o seu conjunto de tipos $\mathcal{T}$. Uma versão generalizada do sistema, que se aplica a λ-termos quaisquer pode ser encontrada em [16].

Definição 2.7 *O conjunto de tipos $\mathcal{T}$ é indutivamente definido a partir de um conjunto numerável $\mathcal{V}ar$ de variáveis para tipos e da conectiva $\rightarrow$:*

- $\mathcal{V}ar \subseteq \mathcal{T}$;
- $\alpha, \beta \in \mathcal{T} \quad \Rightarrow \quad (\alpha \to \beta) \in \mathcal{T}$.

Enquanto que as variáveis para tipos se designam por $a, b, c, \ldots$, utilizamos as letras $\alpha, \beta, \gamma, \ldots$ quando nos referimos a tipos arbitrários. Para a conectiva $\to$ consideramos associatividade à direita abreviando $(\alpha_1 \to (\alpha_2 \to \ldots \to (\alpha_{n-1} \to \alpha_n) \ldots))$ por $\alpha_1 \to \alpha_2 \to \ldots \to \alpha_n$.

Definição 2.8 *Uma declaração é uma expressão da forma $M : \alpha$ onde M é um λ-termo e $\alpha \in \mathcal{T}$; o termo M diz-se o sujeito da declaração. Um conjunto $\mathcal{B}$ finito, possivelmente vazio, de declarações que têm variáveis distintas como sujeitos chamamos uma base. Se $\mathcal{B} = \{x_1 : \rho_1, \ldots, x_m : \rho_m\}$ definimos $Subjects(\mathcal{B}) = \{x_1, \ldots, x_m\}$. Uma* **TA***-fórmula é qualquer expressão da forma $\mathcal{B} \vdash M : \alpha$.*

Definição 2.9 *Dizemos que um tipo $\alpha \in \mathcal{T}$ é inferível para um termo $M \in \Lambda$ em forma normal-β a partir de uma base $\mathcal{B}$ e escrevemos $\mathcal{B} \vdash M : \alpha$ se $\mathcal{B} \vdash M : \alpha$ pode ser obtido, aplicando um número finito de vezes as regras seguintes:*

$$\text{(axioma)} \quad \frac{}{\mathcal{B} \vdash x : \alpha} \quad (\text{se } x : \alpha \in \mathcal{B})$$

$$\text{(app)} \quad \frac{\mathcal{B} \vdash M_1 : \alpha_1 \quad \ldots \quad \mathcal{B} \vdash M_n : \alpha_n}{\mathcal{B} \vdash x M_1 \ldots M_n : \beta} \quad (\text{se } x : \alpha_1 \to \ldots \to \alpha_n \to \beta \in \mathcal{B},\ n \geq 1)$$

$$\text{(abs)} \quad \frac{\mathcal{B}, x : \alpha \vdash M : \beta}{\mathcal{B} \vdash \lambda x.M : \alpha \to \beta}$$

Uma **TA***-dedução Δ da fórmula $\mathcal{B} \vdash M : \tau$ é uma árvore de* **TA***-fórmulas, com raíz $\mathcal{B} \vdash M : \tau$, com axiomas nas folhas e tal que as fórmulas nos nós foram obtidas das anteriores por uma das regras (app) ou (abs).*

Um termo M diz-se tipável, se existir uma base $\mathcal{B}$ e um tipo α tal que $\mathcal{B} \vdash M : \alpha$. Inversamente dizemos que um par $(\mathcal{B}, \alpha)$ é um par admissível, se existir um termo M tal que $\mathcal{B} \vdash M : \alpha$. Neste caso chamamos a M também um habitante de $(\mathcal{B}, \alpha)$. Quando $\mathcal{B} = \emptyset$, dizemos também que α é um tipo admissível e que M é um habitante de α.

Proposição 2.10 *Seja M é um termo em forma normal-β, $\mathcal{B}$ uma base e τ tal que $\mathcal{B} \vdash M : \tau$. Então existe uma e uma só* **TA***-dedução Δ de $\mathcal{B} \vdash M : \tau$.* •

2.3 Tipos principais

O conjunto de todos os tipos inferíveis para um λ-termo tipável M pode ser representado por um único tipo, o tipo principal de M, a partir do qual todos os outros se obtêm por substituição, cf. [14] e [9].

Definição 2.11 *Considerem-se dois tipos $\alpha, \beta \in \mathcal{T}$. Uma substituição $\star$ diz-se um unificador de α e β se $\alpha^\star = \beta^\star$. Um unificador $\star$ de α e β é um unificador mais geral de α e β se para qualquer unificador $\star_1$ (de α e β) existir uma substituição $\star_2$ tal que $\star_1 = \star_2 \circ \star$, i.e. $\star_1(\sigma) = \star_2(\star(\sigma))$, para todo o $\tau \in \mathcal{T}$. Se existirem duas substituições $\star_1$ e $\star_2$ tais que $\alpha^{\star_1} = \beta$ e $\beta^{\star_2} = \alpha$, então dizemos que os tipos α e β são variantes um do outro. Analogamente dizemos ainda que um par $(\mathcal{B}_1, \alpha)$ é uma variante de um outro par $(\mathcal{B}_2, \beta)$ se $(\mathcal{B}_1^{\star_1}, \alpha^{\star_1}) = (\mathcal{B}_2, \beta)$ e $(\mathcal{B}_2^{\star_2}, \beta^{\star_2}) = (\mathcal{B}_1, \alpha)$ para duas substituições $\star_1$ e $\star_2$.*

Definição 2.12 *Considere-se um λ-termo M, uma base $\mathcal{B} = \{x_1 : \alpha_1, \ldots, x_n : \alpha_n\}$ e um tipo $\alpha \in \mathcal{T}$. O par $(\mathcal{B}, \alpha)$ é um par principal de M, se*

1. $\mathcal{B} \vdash M : \alpha$*;*

2. $\mathcal{B}' \vdash M : \alpha' \quad \Rightarrow \quad \exists\star : (\mathcal{B}^\star \subseteq \mathcal{B}' \quad e \quad \alpha^\star = \alpha')$*.*

Se M *for um termo fechado dizemos que* $\alpha \in \mathcal{T}$ *é um tipo principal de* M*, se*

1. $\vdash M : \alpha$*;*

2. $\vdash M : \alpha' \quad \Rightarrow \quad \exists\star : \alpha^\star = \alpha'$*.*

Seja M um λ-termo com um par principal $(\mathcal{B}, \alpha)$. Então verifica-se facilmente que um outro par é também um par principal de M se e só se for uma variante de $(\mathcal{B}, \alpha)$. Foi provado independentemente por Curry em [9], Hindley em [14] e Milner em [22], que qualquer λ-termo tipável M tem um par principal (resp. tipo principal). Mais ainda, que existe um algoritmo que calcula um par principal (resp. tipo principal) para qualquer termo tipável e que falha quando aplicado a um termo que não admite tipo. Por outro lado, é ainda decidível se um dado par é ou não um par principal para um termo M.

3 Habitantes principais em BCK e BCI

Nesta secção apresentamos um algoritmo que decide se um dado par $(\mathcal{B}, \alpha)$ é ou não um par principal para algum **BCK**-termo M em forma normal-β. Se este for o caso, então o algoritmo constrói ainda um termo M' com $M' =_\alpha M$. Concluímos da correcção do algoritmo e da unicidade do seu resultado, que todo o par $(\mathcal{B}, \tau)$ (e em particular todo o tipo τ) tem quando muito um habitante principal normal no **BCK**-calculus e que este é calculado pelo algoritmo. Como os **BCI**-termos são também **BCK**-termos, podemos ainda usar o mesmo algoritmo para decidir se um par tem um habitante principal normal no **BCI**-calculus, e calculá-lo se for o caso.

Começamos por apresentar o algoritmo $\mathcal{PP}$ para calcular um par principal para um **BCK**-termo M em forma normal-β.

Definição 3.1 *Para um* **BCK***-termo* M *em forma normal-*β*, definimos* $\mathcal{PP}(M)$ *por*

- *se* M *é uma variável* x*, então* $\mathcal{PP}(M) = (\{x : a\}, a)$ *onde* a *é uma nova variável para tipos;*
- *para* M *da forma* $xM_1 \ldots M_n$ *seja* $\mathcal{PP}(M_i) = (\mathcal{B}_i, \alpha_i)$ *para* $i = 1, \ldots, n$*, onde* $(\mathcal{B}_1, \alpha_1), \ldots, (\mathcal{B}_n, \alpha_n)$ *não têm variáveis (para tipos) em comum. Então* $\mathcal{PP}(M) = (\{x : \alpha_1 \to \ldots \to \alpha_n \to a\} \cup \mathcal{B}_1 \cup \ldots \cup \mathcal{B}_n, a)$ *onde* a *é nova;*
- *para* M *da forma* $\lambda x.N$ *seja* $\mathcal{PP}(N) = (\mathcal{B}', \alpha)$*. Se* $x : \beta$ *ocorre em* $\mathcal{B}'$*, então* $\mathcal{PP}(M) = (\mathcal{B}' \setminus \{x : \beta\}, \beta \to \alpha)$*. Caso contrário, i.e. se* x *não ocorre em* $\mathcal{B}'$*, então* $\mathcal{PP}(M) = (\mathcal{B}', a \to \alpha)$ *onde* a *é uma nova variável para tipos.*

O algoritmo anterior é uma particularização, para o **BCK**-calculus, do algoritmo generalizado para calcular pares principais de λ-termos. A simplicidade do algoritmo para este caso permite-nos, no entanto, definir um algoritmo "inverso" para calcular, caso exista, um habitante principal normal de um par $(\mathcal{B}, \alpha)$ no **BCK**-calculus.

Definição 3.2 *Para cada par* $(\mathcal{B}, \alpha)$ *o algoritmo* $\mathcal{L}(\mathcal{B}, \alpha)$ *ou falha ou então retorna um termo* M *como se segue:*

- *se* α *é uma variável* a*, então o algoritmo falha se não existir uma única declaração* $x : \beta$ *em* $\mathcal{B}$ *tal que* $\beta = \alpha_1 \to \ldots \to \alpha_n \to a$ *com* $n \geq 0$*. Neste caso,* $\mathcal{L}(\mathcal{B}, \alpha)$ *sucede com* $xM_1 \ldots M_n$ *se e só se existir uma partição de* $\mathcal{B} \setminus \{x : \beta\}$ *em* n *bases disjuntas* $\mathcal{B}_1, \ldots, \mathcal{B}_n$*, tais que* $(\mathcal{B}_1, \alpha_1), \ldots, (\mathcal{B}_n, \alpha_n), (\emptyset, a)$ *não têm variáveis para tipos em comum, e tais que, para i=1,...,n,* $\mathcal{L}(\mathcal{B}_i, \alpha_i)$ *sucede com* M_i*.*

- *para α da forma $\alpha_1 \to \alpha_2$, seja y uma variável nova (para termos). Então o algoritmo sucede se $\mathcal{L}(\mathcal{B} \cup \{y : \alpha_1\}, \alpha_2)$ sucede com N, ou ainda se α_1 é uma variável sem ocorrências em $\mathcal{B}$ e em α_2 e $\mathcal{L}(\mathcal{B}, \alpha_2)$ sucede com N. Em ambos os casos o algoritmo sucede com $\lambda y.N$.*

O lema seguinte será necessário para provar a unicidade do resultado do algoritmo anterior na prova da proposição 3.4.

Lema 3.3 *Se $(\mathcal{B}, \alpha)$ é um par principal para algum* **BCK***-termo M em forma normal-β, então não é possível escrever $V = \mathcal{B} \cup \{\alpha\}$ como $V = V_1 \cup V_2$, onde $V_1, V_2 \neq \emptyset$ e tais que V_1 e V_2 não têm variáveis para tipos em comum.*

Prova Por indução estrutural em M. Para $M = x$, tem-se $V = \{x : a, a\}$ e o resultado é verdadeiro.

Se $M = xM_1 \dots M_n$ com par principal $(\{x : \alpha_1 \to \dots \to \alpha_n \to a\} \cup \mathcal{B}_1 \cup \dots \cup \mathcal{B}_n, a)$ e tal que $(\mathcal{B}_i, \alpha_i)$ é um par principal para M_i, para $1 \leq i \leq n$, então $V = \{x : \alpha_1 \to \dots \to \alpha_n \to a\} \cup \mathcal{B}_1 \cup \dots \cup \mathcal{B}_n \cup \{a\}$. Se $x : \alpha_1 \to \dots \to \alpha_n \to a \in V_1$, então $a \in V_1$ e $\mathcal{B}_1, \dots, \mathcal{B}_n \subseteq V_1$ resulta da hipótese de indução, uma vez que $\alpha_1, \dots, \alpha_n$ ocorre em V_1.

Considere $M = \lambda x.N$ com par principal $(\mathcal{B}, \beta \to \alpha)$, onde $(\mathcal{B} \cup \{x : \beta\}, \alpha)$ é um par principal para N. Pela hipótese de indução não é possível escrever $\mathcal{B} \cup \{x : \beta, \alpha\}$ como $V_1 \cup V_2$ e consequentemente o mesmo verifica-se para $\mathcal{B} \cup \{\beta \to \alpha\}$.

Finalmente, seja $M = \lambda x.N$ com par principal $(\mathcal{B}, a \to \alpha)$, onde $(\mathcal{B}, \alpha)$ é um par principal para N. Resulta da hipótese de indução que não é possível escrever $\mathcal{B} \cup \{\alpha\}$ como $V_1 \cup V_2$, donde concluímos o mesmo para $\mathcal{B} \cup \{a \to \alpha\}$. •

Proposição 3.4

1. *Se $(\mathcal{B}, \alpha)$ é um par principal para um* **BCK***-termo M em forma normal-β, então $\mathcal{L}(\mathcal{B}, \alpha)$ sucede com $M' =_\alpha M$.*
2. *Se $\mathcal{L}(\mathcal{B}, \alpha)$ sucede com M, então $(\mathcal{B}, \alpha)$ é um par principal para M.*

Prova

1. Por indução estrutural em M. Se M é uma variável x, então $(\mathcal{B}, \alpha) = (\{x : a\}, a)$ para alguma variável para tipos a e $\mathcal{L}(\mathcal{B}, \alpha)$ sucede com x.

Se M é $xM_1 \dots M_n$ $(n \geq 1)$, então $(\mathcal{B}, \alpha)$ é da forma $(\{x : \alpha_1 \to \dots \to \alpha_n \to a\} \cup \mathcal{B}_1 \cup \dots \cup \mathcal{B}_n, a)$ tal que $(\mathcal{B}_1, \alpha_1), \dots, (\mathcal{B}_n, \alpha_n)$ são respectivamente pares principais para $M_1, \dots, M_n$, não têm variáveis para tipos em comum e a é uma variável nova. Pela hipótese de indução, $\mathcal{L}(\mathcal{B}_i, \alpha_i)$ sucede com $M'_i =_\alpha M_i$, para cada $i = 1, \dots, n$, e consequentemente $\mathcal{L}(\mathcal{B}, \alpha)$ sucede com $xM'_1 \dots M'_n =_\alpha xM_1 \dots M_n$. Mais ainda, resulta do lema 3.3 e do facto de $(\mathcal{B}_i, \alpha_i)$ ser um par principal para M_i, para $1 \leq i \leq n$, que não é possível dividir $\mathcal{B}_i \cup \{\alpha_i\}$ em dois conjuntos não vazios sem variáveis em comum. Consequentemente, a partição de $\mathcal{B} \setminus \{x : \alpha_1 \to \dots \to \alpha_n \to a\}$ em n bases disjuntas $\mathcal{B}'_1, \dots, \mathcal{B}'_n$, tais que $(\mathcal{B}'_1, \alpha_1), \dots, (\mathcal{B}'_n, \alpha_n), (\emptyset, a)$ não têm variáveis para tipos em comum, é única e terá que ser $\mathcal{B}'_1 = \mathcal{B}_1, \dots, \mathcal{B}'_n = \mathcal{B}_n$.

Consideremos M da forma $\lambda x.N$, com $\alpha = \alpha_1 \to \alpha_2$ e tal que $(\mathcal{B} \cup \{x : \alpha_1\}, \alpha_2)$ é um par principal para N. Então, $\mathcal{L}(\mathcal{B}, \alpha)$ sucede com $\lambda y.N' =_\alpha \lambda x.N$, uma vez que pela hipótese de indução $\mathcal{L}(\mathcal{B} \cup \{y : \alpha_1\}, \alpha_2)$ sucede com $N' =_\alpha N[y/x]$[1].

Finalmente, seja M da forma $\lambda x.N$, com $\alpha = a \to \alpha_2$, onde a é uma variável para tipos nova, x não ocorre em $\mathcal{B}$ e $(\mathcal{B}, \alpha_2)$ é um par principal para N. Então, $\mathcal{L}(\mathcal{B}, \alpha_2)$ sucede com $N' =_\alpha N$ e consequentemente $\mathcal{L}(\mathcal{B}, \alpha)$ sucede com $\lambda y.N$, onde y é uma variável nova. Note-se que nem x nem y ocorrem livres em N, donde $\lambda y.N' =_\alpha \lambda x.N$.

[1] Note-se que, se y é uma variável nova, então $(\mathcal{B}'[y/x], \alpha)$ é um par principal para $N[y/x]$ se e só se $(\mathcal{B}', \alpha)$ é um par principal para N.

2. Por indução estrutural em M. Começamos por notar que $\mathcal{L}(\mathcal{B},\alpha)$ sucede com x se e só se $\alpha = a$ e $\mathcal{B} = \{x : a\}$, para alguma variável para tipos a.

Suponhamos que $\mathcal{L}(\mathcal{B}, a)$ sucede com um termo da forma $xM_1 \ldots M_n$, onde $n \geq 1$. Então existe $x : \alpha_1 \to \ldots \to \alpha_n \to a \in \mathcal{B}$ e existe uma partição de $\mathcal{B} \setminus \{x : \alpha_1 \to \ldots \to \alpha_n \to a\}$ em n bases disjuntas $\mathcal{B}_1, \ldots, \mathcal{B}_n$, tais que $(\mathcal{B}_1, \alpha_1), \ldots, (\mathcal{B}_n, \alpha_n)$ e $(\emptyset, a)$ não têm variáveis para tipos em comum, e para $i = 1, \ldots, n$, $\mathcal{L}(\mathcal{B}_i, \alpha_i)$ sucede com M_i. Da hipótese de indução concluímos que, para $i = 1, \ldots, n$, $(\mathcal{B}_i, \alpha_i)$ é um par principal para M_i, donde $\mathcal{L}(\mathcal{B}, a)$ é um par principal para $xM_1 \ldots M_n$.

Finalmente, seja M da forma $\lambda y.N$. Se $\mathcal{L}(\mathcal{B}, \alpha_1 \to \alpha_2)$ sucedeu por $\mathcal{L}(\mathcal{B} \cup \{y : \alpha_1\}, \alpha_2)$ ter sucedido com N, então resulta da hipótese de indução que $(\mathcal{B} \cup \{y : \alpha_1\}, \alpha_2)$ é um par principal para N, e consequentemente $(\mathcal{B}, \alpha_1 \to \alpha_2)$ é um par principal para $\lambda y.N$. Caso contrário, se α_1 é uma variável para tipos sem outras ocorrências em $\mathcal{B}$ e α_2, se y é nova e se $\mathcal{L}(\mathcal{B}, \alpha_2)$ sucedeu com N, então y não ocorre em $\mathcal{B}$ e concluímos que $(\mathcal{B}, \alpha_1 \to \alpha_2)$ é um par principal para $\lambda y.N$. $\bullet$

Corolário 3.5 *Sejam M e N dois* **BCK***-termos em forma normal-β. Se M e N têm os mesmos pares principais, então $M =_\alpha N$.*

Corolário 3.6 *Aplicado a um par $(\mathcal{B}, \alpha)$ o algoritmo $\mathcal{L}$ decide se existe (e nesse caso calcula) um habitante principal normal no* **BCK***-calculus. Para verificar se existe um habitante principal normal em* **BCI***, basta verificar se $\mathcal{L}(\mathcal{B}, \alpha)$ é ou não um* **BCI***-termo.*

4 Habitantes principais em λK

Definição 4.1 *Seja M um termo em forma normal-β e $(\mathcal{B}, \tau)$ um par tal que $\mathcal{B} \vdash M : \tau$. O termo M diz-se um habitante normal longo de $(\mathcal{B}', \tau)$ (resp. τ) se para toda a fórmula da forma $\mathcal{B}' \vdash xM_1 \ldots M_n : \alpha$, $n \geq 0$, que ocorre na dedução (única) de $\mathcal{B} \vdash M : \tau$ (resp. $\vdash M : \tau$), α é uma variável para tipos (ou átomo). Um termo M diz-se um habitante normal de um tipo τ sse M está em forma normal-β e se $\vdash M : \tau$. Denotamos o conjunto dos habitantes normais de um tipo τ por $\mathsf{Nhabs}(\tau)$ e o conjunto dos habitantes normais e principais de τ por $\mathsf{Nprinc}(\tau)$. O conjunto dos habitantes normais longos de τ é denotado por $\mathsf{Long}(\tau)$ e o conjunto dos habitantes normais longos e principais por $\mathsf{Lprinc}(\tau)$.*

Assim, $\mathsf{Lprinc}(\tau) = \mathsf{Long}(\tau) \cap \mathsf{Nprinc}(\tau) \subseteq \mathsf{Long}(\tau) \subseteq \mathsf{Nhabs}(\tau)$.

Os conjuntos correspondentes de λI-termos são respectivamente denotados por $\mathsf{Nhabs}^{\mathsf{I}}(\tau)$, $\mathsf{Nprinc}^{\mathsf{I}}(\tau)$, $\mathsf{Long}^{\mathsf{I}}(\tau)$ e $\mathsf{Lprinc}^{\mathsf{I}}(\tau)$.

Definição 4.2 *Um η-redex é um λ-termo da forma $\lambda x.Mx$ tal que $x \notin FV(M)$. O contractum de $\lambda x.Mx$ é o termo M. As relações $\to_{1\eta}$, $\to_\eta$ e $=_\eta$ definem-se então analogamente às relações $\to_{1\beta}$, $\to_\beta$ e $=_\beta$, cf. definição 2.3. Por um $\beta\eta$-redex entende-se um β-redex ou um η-redex com definições óbvias das relações $\to_{1\beta\eta}$, $\to_{\beta\eta}$ e $=_{\beta\eta}$.*

Ao conjunto (finito) de todos os termos obtidos a partir de um λ-termo M por redução-η chamamos a família-η de M e denotamo-lo por $\{M\}_\eta$. Mostra-se (cf. [16]) que as famílias-η dos habitantes normais longos de um tipo τ formam uma partição de $\mathsf{Nhabs}(\tau)$ tal que cada subconjunto contém um e um só termo longo. Deste e dos dois resultados seguintes (em [16]) concluímos que contar $\mathsf{Nprinc}(\tau)$ corresponde essencialmente a contar $\mathsf{Lprinc}(\tau)$.

Lema 4.3 (Ben-Yelles, 1979) *Todo o habitante normal de um tipo τ pode ser expandido por expansões-η a um e um só (módulo-$=_\alpha$) habitante normal e longo de τ; i.e. $\{M, N \in Long(\tau)$ e $M =_\eta N\} \implies M =_\alpha N$.*

Lema 4.4 (Hindley, 1997) *Seja M^+ o único elemento de $\mathsf{Long}(\tau)$ para qual é possível η-expandir M. Se $M \in \mathsf{Nprinc}(\tau)$, então $M^+ \in \mathsf{Nprinc}(\tau)$.*

Consequentemente, se $M \in \mathsf{Nprinc}(\tau)$, então $M^+ \in \mathsf{Lprinc}(\tau)$. Concluímos que $\mathsf{Nprinc}(\tau) = \emptyset$ sse $\mathsf{Lprinc}(\tau) = \emptyset$ e que $\mathsf{Nprinc}(\tau)$ é infinito sse $\mathsf{Lprinc}(\tau)$ o for. Mais ainda,

$$\mathsf{Nprinc}(\tau) \subseteq \bigcup_{M \in \mathsf{Lprinc}(\tau)} \{M\}_\eta .$$

Assim, o algoritmo de contagem vai ser baseado na contagem de habitantes normais principais longos e seguindo o algoritmo de Ben-Yelles, a procura será feita em profundidade crescente de termos.

Note-se ainda que sempre que M é um λI-termo e $N =_\eta M$, então N é também um λI-termo. Logo, podemos também basear a contagem de $\mathsf{Nhabs}^{\mathsf{I}}(\tau)$ na contagem de $\mathsf{Long}^{\mathsf{I}}(\tau)$, e a contagem de $\mathsf{Nprinc}^{\mathsf{I}}(\tau)$ na contagem de $\mathsf{Lprinc}^{\mathsf{I}}(\tau)$.

Definição 4.5 *Definimos a profundidade de um λ-termo M em forma normal-β, denotada por $Depth(M)$, pelo seguinte.*

i. $Depth(y) = Depth(\lambda x_1 \dots x_m . y) = 0$;

ii. $Depth(\lambda x_1 \dots x_m . y M_1 \dots M_n) = 1 + max_{1 \le j \le n} Depth(M_j)$, $\quad$ *se* $n > 0$.

Em [6] Ben-Yelles definiu um algoritmo, chamado algoritmo de procura, que para cada tipo τ não atómico (note-se que os tipos atómicos não têm habitantes), produz uma sequência $\mathcal{A}(\tau, 0), \mathcal{A}(\tau, 1), \mathcal{A}(\tau, 2), \dots$ de conjuntos finitos de expressões, a que chamou esquemas-nf, tal que cada elemento de $\mathcal{A}(\tau, d+1)$, que é um λ-termo, é uma forma normal-β de profundidade d. Mais precisamente, se $\mathsf{Long}(\tau, d)$ denota o conjunto de habitantes normais longos de τ de profundidade não superior a d, então temos o seguinte.

Teorema 4.6 (Algoritmo de procura para $\mathsf{Long}(\tau)$, Ben-Yelles, 1979)
O algoritmo de procura aceita como entrada qualquer tipo não atómico τ e produz uma sequência finita ou infinita de sequências de conjuntos $\mathcal{A}(\tau, d)$ ($d = 0, 1, 2, \dots$) tal que para todo o $d \ge 0$,

i. *cada elemento de $\mathcal{A}(\tau, d)$ é um esquema-nf longo com tipo τ, e é*

  (a) *um esquema-nf de profundidade d, ou*

  (b) *um λ-termo de profundidade $d - 1$;*

ii. *$\mathcal{A}(\tau, d)$ é finito;*

iii. $\mathsf{Long}(\tau, d) \subseteq \mathcal{A}(\tau, 0) \cup \dots \cup \mathcal{A}(\tau, d+1)$;

iv. *se chamarmos ao conjunto de todos os λ-termos em $\mathcal{A}(\tau, d)$ "$\mathcal{A}_{termos}(\tau, d)$", então*

$$\mathsf{Long}(\tau) = \bigcup_{d \ge 0} \mathcal{A}_{termos}(\tau, d).$$

Assim, podemos basear o algoritmo de contagem para $\mathsf{Nprinc}(\tau)$ no facto de $\mathsf{Lprinc}(\tau)$ ser infinito sse tiver um elemento cuja profundidade está compreendida entre dois inteiros $\mathbf{d_P}(\tau)$ e $\mathbf{D_P}(\tau)$, que são definidos a partir de τ. Mais ainda, se $\mathsf{Lprinc}(\tau)$ não tiver nenhum elemento com profundidade em $[\mathbf{d_P}(\tau); \mathbf{D_P}(\tau)[$, então $\mathsf{Lprinc}(\tau)$ é finito ou vazio conforme o número de habitantes principais longos de τ com profundidade menor que $\mathbf{d_P}(\tau)$ é finito ou zero.

Definição 4.7 *O número total de ocorrências de átomos num tipo τ é denotado por $|\tau|$ e definido por*

$$|a| = 1, \qquad |\rho \to \sigma| = |\rho| + |\sigma|.$$

O número de átomos distintos que ocorrem em τ será denotado por $||\tau||$. Para cada tipo τ definimos ainda

$$\mathbf{d_P}(\tau) = |\tau| \qquad e \qquad \mathbf{D_P}(\tau) = |\tau|^4.$$

Na sub-secção 4.2 iremos provar o seguinte:

Teorema 4.8 *Para cada tipo τ, tem-se*

i. $\mathsf{Lprinc}(\tau) = \emptyset$ *sse nenhum elemento de* $\mathsf{Lprinc}(\tau)$ *tem profundidade* $< \mathbf{D_p}(\tau)$*;*

ii. $\mathsf{Lprinc}(\tau)$ *é infinito sse tiver um elemento* M *com* $\mathbf{d_p}(\tau) \leq depth(M) < \mathbf{D_p}(\tau)$*;*

iii. $\mathsf{Lprinc}(\tau)$ *é finito sse todos os seus elemento com profundidade* $< \mathbf{D_p}(\tau)$ *tiverem profundidade* $< \mathbf{d_p}(\tau)$*.*

Assim, podemos definir o algoritmo seguinte para contar e enumerar $\mathsf{Nprinc}(\tau)$.

Algoritmo de Contagem para $\mathsf{Nprinc}(\tau)$ **4.9** *Se τ é um átomo, então $\mathsf{Nprinc}(\tau)$ é vazio. Caso contrário, aplicar o algoritmo de procura a τ e calcular $\mathcal{A}_{termos}(\tau, d)$ para $d = 0, \ldots, \mathbf{D_p}(\tau)$. Determinar o conjunto $\mathcal{A}_p$ de todos os λ-termos em $\mathcal{A}_{termos}(\tau, 0) \cup \ldots \cup \mathcal{A}_{termos}(\tau, \mathbf{D_p}(\tau))$ que são habitantes principais de τ.*
Caso I *Se $\mathcal{A}_p = \emptyset$, então $\mathsf{Nprinc}(\tau) = \emptyset$.*
Caso II *Se $\mathcal{A}_p$ tiver um elemento com profundidade $\geq \mathbf{d_p}(\tau)$, então $\mathsf{Nprinc}(\tau)$ é infinito. Aplicar o algoritmo de procura e enumerar $\mathcal{A}_{termos}(\tau, d)$ para $d = 0, 1, 2, \ldots$, escrevendo para cada um destes conjuntos os seus membros que são habitantes principais de τ juntamente com os elementos das suas famílias-η que são habitantes principais de τ.*
Caso III *Se todos os elementos de $\mathcal{A}_p$ tiverem profundidade $< \mathbf{d_p}(\tau)$, então $\mathsf{Nprinc}(\tau)$ é finito. Escrever todos os elementos de $\mathcal{A}_p$ juntamente com os elementos das suas famílias-η que são habitantes principais de τ.*

4.1 Deduções principais para habitantes normais longos

A seguir introduzimos os sistema $\mathbf{TA_{pln}}$ com o qual é possível obter uma caracterização simples de deduções principais para habitantes normais longos. Esta caracterização permite-nos então provar os lemas 4.26 e 4.27, que por sua vez têm a correcção dos limites estabelecidos para $\mathsf{Lprinc}(\tau)$ (teorema 4.8) como corolário.

Para determinar se um par $(\mathcal{B}, \tau)$, com habitante normal longo M, é ou não principal (para M), tem que se verificar dois pontos. Primeiro, a estrutura do par $(\mathcal{B}, \tau)$ só deve conter o estritamente necessário para tipar M. Este objectivo será alcançado marcando, a posteriori, todas as setas cuja existência se justifica por aplicação de regras na dedução de $\mathcal{B} \vdash M : \tau$. Depois, tem de ser garantido que o uso de um mesmo átomo em lugares diferentes em $(\mathcal{B}, \tau)$ é realmente necessário. O principal objectivo do sistema $\mathbf{TA_{pln}}$ é o de registar as ocorrências de átomos em $(\mathcal{B}, \tau)$ para as quais isso é o caso. Assim, podemos resolver este segundo ponto, transformando a $\mathbf{TA}$-dedução de $\mathcal{B} \vdash M : \tau$ numa dedução-$\mathbf{TA_{pln}}$ da fórmula $\mathcal{B}_\mathtt{i} \vdash^{pln} M : \tau_\mathtt{i}$. Aqui o par $(\mathcal{B}_\mathtt{i}, \tau_\mathtt{i})$ tem a mesma estrutura que $(\mathcal{B}, \tau)$ mas usa átomos distintos para ocorrências distintas de átomos em $(\mathcal{B}, \tau)$. No sistema $\mathbf{TA_{pln}}$ a necessidade da unificação de ocorrências distintas de átomos em $(\mathcal{B}, \tau)$ é explicitada pela regra-(U).

Definição 4.10 *O sistema $\mathbf{TA_{pln}}$ é constituído por um conjunto infinito de axiomas e três regras de inferência.*

$$\textit{(axioma)}\quad \frac{}{\mathcal{B} \;\vdash^{pln}\; x : a \;\|\; \emptyset}\quad \textit{(se } x : a \in \mathcal{B}\textit{)}$$

$$\textit{(app)}\quad \frac{\mathcal{B} \vdash^{pln} M_1 : \alpha_1 \;\|\; \phi_1 \quad \ldots \quad \mathcal{B} \vdash^{pln} M_n : \alpha_n \;\|\; \phi_n}{\mathcal{B} \;\vdash^{pln}\; xM_1 \ldots M_n : b \;\|\; \emptyset}$$

$$\textit{(se } x : \alpha_1 \to \ldots \to \alpha_n \to b \in \mathcal{B} \textit{ e } n \geq 1\textit{)}$$

$$\textit{(abs)}\quad \frac{\mathcal{B}, x : \alpha \;\vdash^{pln}\; M : \beta \;\|\; \phi}{\mathcal{B} \;\vdash^{pln}\; \lambda x.M : \alpha \to \beta \;\|\; \emptyset}$$

$$\textit{(U)}\quad \frac{\mathcal{B} \;\vdash^{pln}\; M : a \;\|\; \emptyset}{\mathcal{B} \;\vdash^{pln}\; M : b \;\|\; (a, b)}$$

Uma dedução-$\mathbf{TA_{pln}}$ Δ *é uma árvore constituída por fórmulas-*$\mathbf{TA_{pln}}$*, com axiomas nas folhas e fórmulas obtidas por uma regra de inferência a partir das anteriores nos nós interiores. A raíz de uma dedução* Δ *diz-se a sua conclusão; se esta for* $\mathcal{B} \vdash^{pln} M : \tau \;\|\; \phi$*, dizemos que* Δ *é uma dedução de* $\mathcal{B} \vdash^{pln} M : \tau$.

Definição 4.11 *O* correspondente indexado $(\mathcal{B}_{\mathtt{i}}, \tau_{\mathtt{i}})$ *de um par* $(\mathcal{B}, \tau)$ *obtém-se indexando sucessivamente todas as ocorrências de átomos e setas em* $(\mathcal{B}, \tau)$.

Nota 4.12 *Para qualquer par* $(\mathcal{B}, \tau)$*, tem-se* $||\tau|| \leq |\tau| = |\tau_{\mathtt{i}}| = ||\tau_{\mathtt{i}}||$.

O processo de verificação se um par é ou não um par principal para um termo, usando o sistema $\mathbf{TA_{pln}}$, segue as linhas do exemplo seguinte.

Exemplo 4.13 *O termo* $\lambda xy.x$ *é um habitante normal longo de* $(\emptyset, a \to a \to a)$*. Para verificar se é ou não principal, tranformamos a* $\mathbf{TA}$*-dedução correspondente*

$$\dfrac{\dfrac{x : a, y : a \;\vdash\; x : a}{x : a \;\vdash\; \lambda y.x : a \to a}}{\vdash\; \lambda xy.x : a \to a \to a}$$

numa $\mathbf{TA_{pln}}$*-dedução*

$$\dfrac{\dfrac{\dfrac{x : a_1, y : a_2 \;\vdash^{pln}\; x : a_1 \;\|\; \emptyset}{x : a_1, y : a_2 \;\vdash^{pln}\; x : a_3 \;\|\; (a_1, a_3)}}{x : a_1 \;\vdash^{pln}\; \lambda y.x : a_2 \to_2 a_3 \;\|\; \emptyset}}{\vdash^{pln}\; \lambda xy.x : a_1 \to_1 a_2 \to_2 a_3 \;\|\; \emptyset}$$

da fórmula $\vdash^{pln} \lambda xy.x : a_1 \to_1 a_2 \to_2 a_3$*. Marcamos as setas* $\to_1$ *e* $\to_2$ *a posteriori nesta dedução, uma vez que a sua existência é justificada pela regra (abs). O uso da mesma variável* a *para a sua primeira e última ocorrência em* $a \to a \to a$ *é justificado pelo par* (a_1, a_3)*, mas não existe justificação para o uso da mesma variável* a *na sua segunda ocorrência.*

No que se segue definimos um algoritmo para transformar a dedução-$\mathbf{TA}$ Δ de uma fórmula $\mathcal{B} \vdash M : \tau$, onde M é um habitante normal longo de $(\mathcal{B}, \tau)$ numa dedução-$\mathbf{TA_{pln}}$ $\Delta_{\mathtt{i}}$ da fórmula $\mathcal{B}_{\mathtt{i}} \vdash^{pln} M : \tau_{\mathtt{i}}$. O par $(\mathcal{B}_{\mathtt{i}}, \tau_{\mathtt{i}})$ é o correspondente indexado de $(\mathcal{B}, \tau)$. Na dedução resultante $\Delta_{\mathtt{i}}$, pode haver ocorrências de setas marcadas com o símbolo $\star$. Numa primeira fase é construída uma versão não marcada de $\Delta_{\mathtt{i}}$ a partir de Δ como se segue:

- se $\mathcal{B} \vdash M : \tau$ é um axioma, i.e. $x : a \in \mathcal{B}$, $M = x$ e $\tau = a$, então $x : a_k \in \mathcal{B}'$ e $\tau' = a_l$. Tomar $\mathcal{B}' \vdash^{pln} x : a_l \;\|\; (a_k, a_l)$ como raíz de Δ' (regra-U) precedida por $\mathcal{B}' \vdash^{pln} x : a_k \;\|\; \emptyset$ (axioma).

- se $\mathcal{B} \vdash M : \tau$ resultar de $\mathcal{B} \vdash M_1 : \alpha_1, \ldots, \mathcal{B} \vdash M_n : \alpha_n$ pela regra (app), então $M = xM_1 \ldots M_n$ e $\tau = b$, e existe uma atribuição de tipo $x : \alpha_1' \to \ldots \to \alpha_n' \to b_k$ em $\mathcal{B}'$ e $\tau' = b_l$. Tomar $\mathcal{B}' \vdash^{pln} xM_1 \ldots M_n : b_l \parallel (b_k, b_l)$ como raíz de Δ' (regra-U), precedida por $\mathcal{B}' \vdash^{pln} xM_1 \ldots M_n : b_k \parallel \emptyset$ (app) e preceder esta fórmula pelas deduções $\Delta_1', \ldots, \Delta_n'$ construídas a partir das deduções-**TA** $\Delta_1, \ldots, \Delta_n$ de $\mathcal{B} \vdash M_1 : \alpha_1, \ldots, \mathcal{B} \vdash M_n : \alpha_n$ e pares $(\mathcal{B}', \alpha_1'), \ldots, (\mathcal{B}', \alpha_n')$.
- se $\mathcal{B} \vdash M : \tau$ resultar da regra (abs), i.e. $M = \lambda x.N$ e $\tau = \alpha \to \beta$, de $\mathcal{B}, x : \alpha \vdash N : \beta$, tomar $\mathcal{B}' \vdash^{pln} \lambda x.N : \alpha' \to \beta' \parallel \emptyset$ como raíz de Δ' regra-(abs) precedida pela dedução Δ_1' construída a partir da dedução Δ_1 de $\mathcal{B}, x : \alpha \vdash N : \beta$ e par $(\mathcal{B}' \cup \{x : \alpha'\}, \beta')$.

Numa segunda fase marcam-se setas da forma seguinte.

- Não se marcam setas nos axiomas.
- Se $\mathcal{B} \vdash^{pln} xM_1 \ldots M_n : b \parallel \emptyset$ resultar de $\mathcal{B} \vdash M_1 : \alpha_1 \parallel \phi_1, \ldots, \mathcal{B} \vdash M_n : \alpha_n \parallel \phi_n$ pela regra (app), então marcar $x : \alpha_1 \to \ldots \to \alpha_n \to b$ em $\mathcal{B}$ da forma seguinte: $x : \alpha_1 \to^\star \ldots \to^\star \alpha_n \to^\star b$ (não são marcadas setas em $\alpha_1, \ldots, \alpha_n$).
- Se $\mathcal{B} \vdash^{pln} \lambda x.M : \alpha \to \beta \parallel \emptyset$ resultar de $\mathcal{B}, x : \alpha \vdash^{pln} M : \beta \parallel \phi$ pela regra (abs), marcar a seguinte seta: $\mathcal{B} \vdash^{pln} \lambda x.M : \alpha \to^\star \beta \parallel \emptyset$.
- Não são marcadas setas em fórmulas obtidas pela regra-U.

Exemplo 4.14 *O par $(\emptyset, (a_1 \to_1 a_2) \to_2 a_3 \to_3 a_4)$ é o correspondente indexado de $(\emptyset, (a \to a) \to a \to a)$. Seja $\Theta = \{x : a \to a, y : a\}$, então $\vdash \lambda xy.x(x(xy)) : (a \to a) \to a \to a$ tem a* **TA***-dedução seguinte.*

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Theta \vdash x : a \to a \quad \Theta \vdash y : a}{\Theta \vdash xy : a} \quad \Theta \vdash x : a \to a}{\Theta \vdash x(xy) : a} \quad \Theta \vdash x : a \to a}{x : a \to a, y : a \vdash x(x(xy)) : a}}{x : a \to a \vdash \lambda y.x(x(xy)) : a \to a}}{\vdash \lambda xy.x(x(xy)) : (a \to a) \to a \to a}$$

Seguindo o algoritmo de transformação obtemos a $\mathbf{TA_{pln}}$*-dedução seguinte para a fórmula* $\vdash^{pln} \lambda xy.x(x(xy)) : (a_1 \to_1 a_2) \to_2 a_3 \to_3 a_4$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{x : a_1 \to_1 a_2, y : a_3 \vdash^{pln} x : a_1 \to_1 a_2 \parallel \emptyset \quad \dfrac{x : a_1 \to_1 a_2, y : a_3 \vdash^{pln} y : a_3 \parallel \emptyset}{x : a_1 \to_1 a_2, y : a_3 \vdash^{pln} y : a_1 \parallel (a_3, a_1)}}{x : a_1 \to_1^\star a_2, y : a_3 \vdash^{pln} xy : a_2 \parallel \emptyset}}{x : a_1 \to_1 a_2, y : a_3 \vdash^{pln} xy : a_1 \parallel (a_2, a_1)} \quad x : a_1 \to_1 a_2, y : a_3 \vdash^{pln} x : a_1 \to_1 a_2 \parallel \emptyset}{x : a_1 \to_1^\star a_2, y : a_3 \vdash^{pln} x(xy) : a_2}}{x : a_1 \to_1 a_2, y : a_3 \vdash^{pln} x(xy) : a_1 \parallel (a_2, a_1)} \quad x : a_1 \to_1 a_2, y : a_3 \vdash^{pln} x : a_1 \to_1 a_2 \parallel \emptyset}{x : a_1 \to_1^\star a_2, y : a_3 \vdash^{pln} x(x(xy)) : a_2 \parallel \emptyset}}{x : a_1 \to_1 a_2, y : a_3 \vdash^{pln} x(x(xy)) : a_4 \parallel (a_2, a_4)}}{x : a_1 \to_1 a_2 \vdash^{pln} \lambda y.x(x(xy)) : a_3 \to_3^\star a_4 \parallel \emptyset}}{\vdash^{pln} \lambda xy.x(x(xy)) : (a_1 \to_1 a_2) \to_2^\star a_3 \to_3 a_4 \parallel \emptyset}$$

Note-se que todas as setas indexadas têm pelo menos uma ocorrência marcada nesta dedução o que justifica a estrutura do tipo. Por outro lado, a existência dos pares (a_3, a_1), (a_2, a_1) e (a_2, a_4) nesta dedução torna explícita a necessidade do uso da mesma variável a para todas as suas ocorrências em $(a \to a) \to a \to a$.

Definição 4.15 *Seja τ um tipo e Φ uma relação binária definida no conjunto de átomos que ocorrem em τ. Definimos $\mathcal{C}_\Phi$ como conjunto de classes de equivalência do fecho reflexivo, simétrico e transitivo de Φ.*

Lema 4.16 *Considere-se um conjunto finito não vazio A e n relações binárias $\Phi_1 \subseteq \ldots \subseteq \Phi_n$ definidas em A com $\mathcal{C}_{\Phi_i} \neq \mathcal{C}_{\Phi_j}$ para $1 \leq i \neq j \leq n$. Então $n \leq \#A$.* •

Proposição 4.17 *Seja M um habitante normal longo de um par $(\mathcal{B}, \tau)$ tal que, $Subjects(\mathcal{B}) = FV(M)$. Considere-se o correspondente indexado $(\mathcal{B}_{\mathtt{i}}, \tau_{\mathtt{i}})$ de $(\mathcal{B}, \tau)$. Seja $\Delta_{(\mathcal{B}_{\mathtt{i}}, \tau_{\mathtt{i}})}$ a dedução-$\mathbf{TA_{pln}}$ de $\mathcal{B}_{\mathtt{i}} \vdash^{pln} M : \tau_{\mathtt{i}}$ construída a partir da dedução-$\mathbf{TA}$ de $\mathcal{B} \vdash M : \tau$ e seja Φ o conjunto de todos os pares $\Delta_{(\mathcal{B}_{\mathtt{i}}, \tau_{\mathtt{i}})}$. Então,*

$$(\mathcal{B}, \tau) \text{ é um par principal para } M$$
$$sse$$

para todo o átomo a em $(\mathcal{B}, \tau)$ existe exactamente uma classe de equivalência em $\mathcal{C}_\Phi$ constituída por todas as ocorrências indexadas de a em $(\mathcal{B}_{\mathtt{i}}, \tau_{\mathtt{i}})$ e todas as setas indexadas têm pelo menos uma ocorrência marcada em $\Delta_{(\mathcal{B}_{\mathtt{i}}, \tau_{\mathtt{i}})}$.

Prova Mostra-se facilmente que sempre que $(a, b) \in \Phi$, então a e b são versões indexadas do mesmo átomo. Suponhamos então que $(\mathcal{B}, \tau)$ não é um par principal para M, i.e. que existe um par (mais geral) $(\mathcal{B}_0, \tau_0)$ tal que $\mathcal{B}_0 \vdash M : \tau_0$, $\mathcal{B}_0^\star = \mathcal{B}$ e $\tau_0^\star = \tau$ para alguma substituição $\star$, mas $(\mathcal{B}, \tau)$ não é variante de $(\mathcal{B}_0, \tau_0)$. Se $(\mathcal{B}_0, \tau_0)$ tiver a mesma estrutura de $(\mathcal{B}, \tau)$, mas existirem ocorrências de variáveis distintas em $(\mathcal{B}_0, \tau_0)$ com o mesmo nome a em $(\mathcal{B}, \tau)$, então conclui-se da observação feita no início da prova, que existem pelo menos duas classes de equivalência em $\mathcal{C}_\Phi$ com ocorrências indexadas de a em $(\mathcal{B}_{\mathtt{i}}, \tau_{\mathtt{i}})$. Se, por outro lado, a ocorrência de uma variável em $(\mathcal{B}_0, \tau_0)$ tiver um tipo composto $\alpha \rightarrow \beta$ em $(\mathcal{B}, \tau)$, então a ocorrência correspondente desta seta não vai ser marcada em $\Delta_{(\mathcal{B}_{\mathtt{i}}, \tau_{\mathtt{i}})}$.

Por outro lado, suponhamos que $\mathcal{B} \vdash M : \tau$ onde M é longo relativamente a $(\mathcal{B}, \tau)$ e que $FV(M) \subseteq Subjects(\mathcal{B})$ e sejam $(\mathcal{B}_{\mathtt{i}}, \tau_{\mathtt{i}})$, $\Delta_{(\mathcal{B}_{\mathtt{i}}, \tau_{\mathtt{i}})}$ e Φ como anteriormente. Seja $\Phi' \supseteq \Phi$ uma qualquer relação binária no conjunto das variáveis em $(\mathcal{B}_{\mathtt{i}}, \tau_{\mathtt{i}})$ tal que existe alguma variável para tipos d tal que existem pelo menos duas classes de equivalência distintas $\{d_{i_1}, \ldots, d_{i_m}\}$ e $\{d_{j_1}, \ldots, d_{j_n}\}$ em $\mathcal{C}_{\Phi'}$ com ocorrências indexadas de d. Mostramos por indução estrutural em M que, se substituirmos em $\Delta_{(\mathcal{B}_{\mathtt{i}}, \tau_{\mathtt{i}})}$ as ocorrências de $d_{j_1}, \ldots, d_{j_n}$ por $c_{j_1}, \ldots, c_{j_n}$, para alguma variável nova c, então obtemos uma nova $\mathbf{TA_{pln}}$-dedução Δ'_c que corresponde à $\mathbf{TA}$-dedução Δ_c de $\mathcal{B}_c \vdash M : \tau_c$, onde $(\mathcal{B}_c, \tau_c)$ foi obtido de $(\mathcal{B}, \tau)$ por substituição das ocorrências de d que correspondem a $d_{j_1}, \ldots, d_{j_n}$ por c. Então, $(\mathcal{B}_c, \tau_c)$ é um par mais geral tal que $\mathcal{B}_c \vdash M : \tau_c$ e consequentemente $(\mathcal{B} \restriction FV(M), \tau)$ não é um par principal para M.

Se $M = x$, então $(\mathcal{B}, \tau) = (\{x : a\}, a)$ e o resultado verifica-se trivialmente, uma vez que $\mathcal{C}_\Phi = \{\{a_1, a_2\}\}$. A prova para $M = \lambda x.N$ é directa. Se $M = xM_1 \ldots M_n$, então $\tau = b$, $x : \alpha_1 \rightarrow \ldots \rightarrow \alpha_n \rightarrow b \in \mathcal{B}$ e $\mathcal{B} \vdash xM_1 \ldots M_n : b$ resulta de $\mathcal{B} \vdash M_1 : \alpha_1, \ldots, \mathcal{B} \vdash M_n : \alpha_n$ pela regra (app). Concluímos da hipótese de indução que $\mathcal{B}_c \vdash M_1 : \alpha_1^c, \ldots, \mathcal{B}_c \vdash M_n : \alpha_n^c$. Se $\Delta'_{c,1}, \ldots, \Delta'_{c,n}$ são as $\mathbf{TA_{pln}}$-deduções correspondentes, então Δ'_c é da forma

$$\dfrac{\dfrac{\Delta'_{c,1} \ldots \Delta'_{c,n}}{\mathcal{B}_c \vdash^{pln} xM_1 \ldots M_n : b_k^c \parallel \emptyset}}{\mathcal{B}_c \vdash^{pln} xM_1 \ldots M_n : b_l^c \parallel (b_k^c, b_l^c).}$$

Logo, $(b_k, b_l) \in \Phi \subseteq \Phi'$ pertencem à mesma classe em $\mathcal{C}_\Phi$ e as ocorrências correspondentes em $(\mathcal{B}^c, \tau^c)$ são ocorrências da mesma variável. Assim, $\mathcal{B}^c \vdash xM_1 \ldots M_n : b^c$ pode ser inferido de $\mathcal{B}_c \vdash M_1 : \alpha_1^c, \ldots, \mathcal{B}_c \vdash M_n : \alpha_n^c$ pela regra (app) e Δ'_c corresponde à $\mathbf{TA}$-dedução Δ_c de $\mathcal{B}_c \vdash xM_1 \ldots M_n : b^c$.

Logo, mostramos que sempre que $(\mathcal{B}, \tau)$ é um par principal para M então para cada átomo a em $(\mathcal{B}, \tau)$ existe uma classe de equivalência em $\mathcal{C}_\Phi$ que contém exactamente todas as ocorrências indexadas de a em $(\mathcal{B}_{\mathtt{i}}, \tau_{\mathtt{i}})$. Falta mostrar que todas as setas indexadas têm uma ocorrência marcada em $\Delta_{(\mathcal{B}_{\mathtt{i}}, \tau_{\mathtt{i}})}$. Procedemos por indução estrutural em M.

Os casos $M = x$ e $M = \lambda x.N$ são triviais. Para $M = xM_1 \ldots M_n$ seja $(\mathcal{B}, b)$ um par principal para $xM_1 \ldots M_n$ obtido de $\mathcal{B} \vdash M_1 : \alpha_1, \ldots, \mathcal{B} \vdash M_n : \alpha_n$ pela regra (app) e

$x : \alpha_1 \to \ldots \to \alpha_n \to b \in \mathcal{B}$. Então existem $(\mathcal{B}_1, \alpha_1^0), \ldots, (\mathcal{B}_n, \alpha_n^0)$ respectivamente pares principais para $M_1, \ldots, M_n$, tais que $\mathcal{B} = \mathcal{B}_1^\star \cup \ldots \cup \mathcal{B}_n^\star \cup \{x : v_1 \to \ldots \to v_n \to v\}^\star$, onde $\star$ é a substituição que resulta da unificação dos tipos atribuídos às variáveis em $\mathcal{B}_1, \ldots, \mathcal{B}_n$ e em $\{x : v_1 \to \ldots \to v_n \to v\}$ ($v_1, \ldots, v_n, v$ são átomos novos) e tal que $\alpha_1 = \star(v_1) = \star(\alpha_1^0), \ldots, \alpha_n = \star(v_n) = \star(\alpha_n^0)$ e $b = \star(v)$. Pela hipótese de indução, todas as ocorrências de setas em $\mathcal{B}_1, \ldots, \mathcal{B}_n, \alpha_1^0, \ldots, \alpha_n^0$ ocorrem marcadas nas $\mathbf{TA_{pln}}$-deduções que correspondem às $\mathbf{TA}$-deduções de $\mathcal{B}_1 \vdash M_1 : \alpha_1, \ldots, \mathcal{B}_n \vdash M_n : \alpha_n$. Assim, as setas que resultam da unificação e que correspondem a setas em $\mathcal{B}_1, \ldots, \mathcal{B}_n, \alpha_1^0, \ldots, \alpha_n^0$ ocorrem também marcadas na $\mathbf{TA_{pln}}$-dedução que corresponde à $\mathbf{TA}$-dedução de $\mathcal{B} \vdash xM_1 \ldots M_n : b$. Finalmente, note-se que as n setas principais em $\alpha_1 \to \ldots \to \alpha_n \to b$ serão também marcadas. •

Exemplo 4.18 *Para o exemplo anterior tem-se* $\mathcal{C}_\Phi = \{\{a_1, a_2, a_3, a_4\}\}$. *Por outro lado, todas as setas indexadas têm uma ocorrência marcada na* $\mathbf{TA_{pln}}$*-dedução. Assim, concluímos que* $(\emptyset, (a \to a) \to a \to a)$ *é um par principal para* $\lambda xy.x(x(xy))$.

4.2 Correcção de $\mathbf{d_p}(\tau)$ e $\mathbf{D_p}(\tau)$

Começamos por enumerar algumas definições e resultados sobre a estrutura de tipos e termos, que serão necessários para provar os teoremas 4.8 e 5.6. Note-se que todo o tipo τ se escreve de maneira única sob a forma $\tau_1 \to \ldots \to \tau_m \to e$, tal que e é um átomo (variável para tipos) e $m \geq 0$.

Definição 4.19 *Os sub-tipos significantes (ou s-subtipos) de um tipo* $\tau = \tau_1 \to \ldots \to \tau_m \to e$, *onde* e *é um átomo e* $m \geq 0$, *definem-se recursivamente pelo seguinte.*

- τ *é um s-subtipo de* τ;
- *todos os s-subtipos de um de* $\tau_1, \ldots, \tau_m, e$ *é também um s-subtipo de* τ.

Ocorrências de s-subtipos de τ *dizem-se também s-componentes de* τ *e distinguem-se sublinhando os seus nomes. Uma s-componente de um tipo* τ *define-se como positiva ou negativa pelo seguinte.*

- $\underline{\tau}$ *é uma s-componente positiva de* τ;
- *se* $\tau = \tau_1 \to \ldots \to \tau_m \to e$, *então* $\underline{\tau_1}, \ldots, \underline{\tau_m}$ *são s-componentes negativas de* τ *e* $\underline{e}$ *é uma s-componente positiva de* τ;
- *se* $\tau = \tau_1 \to \ldots \to \tau_m \to e$ *e se* $\underline{\rho}$ *é uma s-componente de um de* $\tau_1, \ldots, \tau_m$, *então* $\underline{\rho}$ *é uma s-componente positiva ou negativa de* τ *conforme é uma s-componente negativa ou positiva de* $\tau_1, \tau_2, \ldots$ *ou de* τ_m.

Definição 4.20 *Se* $\underline{\rho}$ *é uma s-componente composta (i.e. não atómica) de um tipo* τ *tal que* $\rho = \rho_1 \to \ldots \to \rho_n \to a$ $(n \geq 1)$, *então chamamos as s-componentes* $\underline{\rho_1}, \ldots, \underline{\rho_n}$ *as premissas de* $\underline{\rho}$ *e* $\underline{a}$ *a conclusão de* $\underline{\rho}$.

Uma s-componente de τ *diz-se uma sub-premissa (resp. sub-conclusão) de* τ *se é uma premissa ou (resp. conclusão) de uma outra s-componente de* τ.

Definição 4.21 *Se* τ *é composto, então* $\mathsf{NSS}(\tau)$ *é o conjunto das sequências finitas* $< \sigma_1, \ldots, \sigma_n >$ $(n \geq 1)$ *tais que* τ *contém uma s-componente positiva composta da forma* $\sigma_1 \to \ldots \to \sigma_n \to a$ *para algum átomo* a. *Cada elemento de* $\mathsf{NSS}(\tau)$ *diz-se uma sequência negativa de sub-premissas.*

Todo o λ-termo X em forma normal-β escreve-se de forma única como

$$X = \lambda x_1 \ldots x_m.vY_1 \ldots Y_n, \qquad (m + n \geq 1).$$

A cabeça e os argumentos de X são respectivamente $\underline{v}$ e $\underline{Y_1}, \ldots, \underline{Y_n}$.

Definição 4.22 *Um sub-argumento de um λ-termo X em forma normal-β é uma componente que é um argumento de X ou de uma componente própria de X. Se $\underline{Z}$ é um sub-argumento de um λ-termo X, então a cadeia de argumentos de $\underline{X}$ até $\underline{Z}$ é a sequência $< \underline{Z_0}, \underline{Z_1}, \ldots, \underline{Z_k} >$, $(k \geq 1)$, tal que $\underline{Z_0} = \underline{X}$ e $\underline{Z_i}$ é um argumento de $\underline{Z_{i-1}}$ para $i = 1, \ldots, k$, e $\underline{Z_k} = \underline{Z}$. Chamamos-lhe não extensível sse $\underline{Z}$ é um átomo ou um átomo abstraído. O seu comprimento é k (não $k+1$).*

Definição 4.23 *Seja Δ uma $\mathbf{TA_{pln}}$-dedução de $\mathcal{B} \vdash^{pln} M : \tau$, seja $\underline{Z}$ um sub-argumento de M, com*

$$Z = \lambda x_1 \ldots x_m . y Z_1 \ldots Z_n \qquad (m, n \geq 0)$$

e seja $\mathcal{B}_Z \vdash^{pln} Z : \alpha_1 \rightarrow \ldots \alpha_m \rightarrow a \parallel \emptyset$ o nó em Δ que corresponde a $\underline{Z}$. A sequência de tipos das abstrações iniciais $\mathsf{IAT}(\underline{Z})$ é definida por

$$\mathsf{IAT}(\underline{Z}) = < \alpha_1, \ldots, \alpha_m >$$

e tem comprimento m. A sequência de abstrações iniciais $\mathsf{IA}(Z)$ é a sequência

$$\mathsf{IA}(Z) = < x_1, \ldots, x_m > .$$

Lema 4.24 *Seja Δ a dedução de $\vdash M : \tau$, onde M é um habitante normal longo de τ e seja Δ' a $\mathbf{TA_{pln}}$-dedução correspondente de $\vdash^{pln} M : \tau_{\mathtt{i}}$. Seja $\underline{Z}$ um sub-argumento de M, e seja $\mathcal{B}_Z \vdash^{pln} Z : \sigma \parallel \emptyset$, com $\sigma = \sigma_1 \rightarrow \ldots \rightarrow \sigma_k \rightarrow s$, o nó em Δ' que corresponde a $\underline{Z}$. Então,*

i. se σ é um átomo, então $\mathsf{IAT}(\underline{Z}) = \emptyset$;

ii. se σ é um tipo composto, então $\mathsf{IAT}(\underline{Z}) \in \mathsf{NSS}(\tau_{\mathtt{i}})$.

Prova A parte *i.* é trivial. Para a parte *ii.*, mostramos por indução sobre a profundidade de $\underline{Z}$ em M, i.e. o comprimento da cadeia de argumentos de M a $\underline{Z}$, que sempre que $x : \alpha \in \mathcal{B}_Z$, então α ocorre como s-componente negativa em $\tau_{\mathtt{i}}$ e σ (composto) ocorre como s-componente positiva em τ. De facto, suponhamos que $\underline{Z}$ tem profundidade 1 em M, i.e. $M = \lambda x_1 \ldots x_m . v Y_1 \ldots Y_n$ com $m, n \geq 1$ e $\underline{Z} = Y_i$ para algum $1 \leq i \leq n$. Então, $\tau_{\mathtt{i}} = \alpha_1 \rightarrow \ldots \rightarrow \alpha_m \rightarrow a$ e o nó em Δ' que corresponde a $\underline{Z}$ é $x_1 : \alpha_1, \ldots, x_m : \alpha_m \vdash^{pln} Z : \sigma \parallel \emptyset$ (como σ é composto, este nó resulta da regra (abs)). Por definição, $\alpha_1, \ldots, \alpha_m$ são s-componentes negativas de $\tau_{\mathtt{i}}$. Por outro lado, tem-se $v = x_j$ para algum $1 \leq j \leq m$ e α_j é da forma $\beta_1 \rightarrow \ldots \rightarrow \beta_n \rightarrow b$ com $\beta_i = \sigma$. Logo, σ ocorre como s-componente negativa de uma s-componente negativa de $\tau_{\mathtt{i}}$ e ocorre então como s-componente positiva em $\tau_{\mathtt{i}}$.

O passo de indução é essencialmente uma repetição do raciocínio anterior.

Assim, se σ é composto, então σ ocorre como s-componente positiva em $\tau_{\mathtt{i}}$ e por definição de $\mathsf{NSS}(\tau_{\mathtt{i}})$ resulta $\mathsf{IAT}(\underline{Z}) = < \sigma_1, \ldots, \sigma_k > \in \mathsf{NSS}(\tau_{\mathtt{i}})$. •

Foi mostrado em [16] que sempre que τ é um tipo composto, então $\#(\mathsf{NSS}(\tau)) \leq |\tau| - 1$. Assim, se $(\emptyset, \tau_{\mathtt{i}})$ é a correspondente indexada de $(\emptyset, \tau)$, então $\#(\mathsf{NSS}(\tau_{\mathtt{i}})) \leq |\tau_{\mathtt{i}}| - 1 = |\tau| - 1$.

Lema 4.25 *Se τ é composto e se $(\emptyset, \tau_{\mathtt{i}})$ é o correspondente indexado de $(\emptyset, \tau)$, então* $\#(\mathsf{NSS}(\tau_{\mathtt{i}})) \leq |\tau| - 1$. •

Os dois resultados seguintes mostram a correcção dos limites $\mathbf{d_p}(\tau)$ e $\mathbf{D_p}(\tau)$.

Lema 4.26 *Se $\mathsf{Lprinc}(\tau)$ tiver um elemento M com profundidade $\geq \mathbf{D_p}(\tau)$, então*

i. existe $M^\star \in \mathsf{Lprinc}(\tau)$ com $Depth(M) - |\tau|^3 \leq Depth(M^\star) < Depth(M)$;

ii. existe $N \in \mathsf{Lprinc}(\tau)$ com $\mathbf{D_p}(\tau) - |\tau|^3 \leq Depth(N) < \mathbf{D_p}(\tau)$.

Prova Para a parte $i.$ consideremos um λ-termo $M \in \mathsf{Lprinc}(\tau)$ com profundidade $d \geq \mathbf{D_p}(\tau)$ e e sem conflitos de variáveis ligadas. Seja $\Delta_{(\emptyset,\tau_{\mathtt{i}})}$ a $\mathbf{TA_{pln}}$-dedução de $\vdash^{pln} M : \tau_{\mathtt{i}}$ construída a partir da $\mathbf{TA}$-dedução de $\vdash M : \tau$ e seja Φ o conjunto dos pares binários em $\Delta_{(\emptyset,\tau_{\mathtt{i}})}$. Resulta da proposição 4.17 que para cada átomo a em τ existe uma classe de equivalência em $\mathcal{C}_\Phi$ que contém exactamente as ocorrências indexadas de a em $\tau_{\mathtt{i}}$ e que todas as setas (indexadas) em τ_i têm uma ocorrência marcada em $\Delta_{(\emptyset,\tau_{\mathtt{i}})}$. No que se segue construímos um termo de profundidade $< Depth(M)$ que é um habitante principal normal de $(\emptyset, \tau)$. Foi mostrado em [16] que $d = Depth(M)$ é o máximo dos comprimentos de todas as cadeias de argumentos em M. Assim, se M tiver cadeias de argumentos não estensíveis de comprimento d, é necessário encurtar o comprimento de todas estas cadeias para reduzir a profundidade de M. Seja então, $< N_0, \ldots, N_d >$ uma cadeia de comprimento d com

$$N_i = \lambda x_{i,1} \ldots x_{i,m_i} . y_i P_{i,1} \ldots P_{i,n_i} \qquad (m_i, n_i \geq 0).$$

Seja

$$\mathcal{B}_i \vdash^{pln} N_i : \rho_{i,1} \to \ldots \to \rho_{i,m_i} \to a_i \parallel \emptyset$$

o nó em $\Delta_{(\emptyset,\tau_{\mathtt{i}})}$ que corresponde a $\underline{N_i}$, para $i = 0, \ldots, d$. Logo, $\mathsf{IAT}(\underline{N_i}) =< \rho_{i,1}, \ldots, \rho_{i,m_i} >$. Para $i = 0, \ldots, d$ seja $\underline{B_i}$ o corpo de $\underline{N_i}$, i.e. $B_i = y_i P_{i,1} \ldots P_{i,n_i}$ e seja $\mathcal{B}_{B_i} \vdash^{pln} B_i : a_{b_i} \parallel \emptyset$ o nó em $\Delta_{(\emptyset,\tau_{\mathtt{i}})}$ que corresponde a $\underline{B_i}$. Mais ainda, sejam Φ_i e $\mathcal{I}_i$ respectivamente o conjunto de pares binários e de índices de setas marcadas na sub-árvore de $\Delta_{(\emptyset,\tau_{\mathtt{i}})}$ com raíz $\mathcal{B}_{B_i} \vdash^{pln} B_i : a_{b_i} \parallel \emptyset$. Então, $\mathcal{I}_i \subseteq \mathcal{I}_{i-1}$ e $\Phi_i \subseteq \Phi_{i-1}$ para $i = 1, \ldots, d$.

Definimos uma sequência de inteiros $d_0, d_1, \ldots, d_n$ pelo seguinte: $d_0 = 0$ e d_{j+1} é o menor dos $i > d_j$ tais que $\mathsf{IAT}(\underline{N_i})$ difere de todos os $\mathsf{IAT}(\underline{N_{d_0}}), \ldots, \mathsf{IAT}(\underline{N_{d_j}})$. Obviamente, tem-se $n \leq d$ bem como $0 = d_0 < d_1 < \ldots < d_n \leq d$. Mais ainda, para $0 \leq i \leq d$, $\mathsf{IAT}(\underline{N_i})$ é um de $n+1$ $\mathsf{IAT}(\underline{N_{d_0}}), \ldots, \mathsf{IAT}(\underline{N_{d_n}})$, que são todos distintos e por 4.24 são vazios ou então pertencem a $\mathsf{NSS}(\tau_{\mathtt{i}})$. Resulta do lema 4.25 que

$$n + 1 \leq 1 + \#(\mathsf{NSS}(\tau_{\mathtt{i}})) \leq 1 + |\tau| - 1 = |\tau|.$$

Para $i = 0, \ldots, n$ definimos os conjunto não vazios seguintes, aos quais chamamos IAT-intervalos:

$$\begin{array}{rcll} \mathsf{I}_j & = & \{d_j, d_j + 1, \ldots, d_{j+1} - 1\} & \quad 0 \leq j \leq n-1 \\ \mathsf{I}_n & = & \{d_n, d_n + 1, \ldots, d\}. & \end{array}$$

Se I_j contém dois inteiros p, $p + r$ tais que $r \geq 1$ e B_p e B_{p+r} têm o mesmo tipo (i.e. $a_{b_p} = a_{b_{p+r}}$), $\mathcal{C}_{\Phi_p} = \mathcal{C}_{\Phi_{p+r}}$ e $\mathcal{I}_p = \mathcal{I}_{p+r}$, então chamamos a $< p, p + r >$ uma repetição de conclusões. Chamamos-lhe minimal se não existir nenhuma outra repetição de conclusões $< p', q' >$ com $p \leq p' < q' \leq p + r$. Concluímos que todo o IAT-intervalo I_j sem repetição de conclusões tem $\leq |\tau|^3$ elementos e que $r \leq |\tau|^3$. De facto, existem $|\tau|$ átomos distintos em $\tau_{\mathtt{i}}$, e consequentemente existem no máximo $|\tau - 1|$ classes de equivalência distintas correspondentes a $\Phi_0 \supseteq \Phi_1 \supseteq \ldots \supseteq \Phi_d$ (cf. lema 4.16), bem como no máximo $|\tau|$ conjuntos distintos de índices entre $\mathcal{I}_0 \supseteq \ldots \supseteq \mathcal{I}_d$ (note-se que existem exactamente $|\tau|$ setas indexadas em $\tau_{\mathtt{i}}$). Logo, se nenhum dos $n+1$ IAT-intervalos contém uma repetição de conclusões, então a cadeia teria $\leq |\tau|^4$ elemento. Mas a cadeia tem $d + 1$ elementos e

$$d + 1 = Depth(M) + 1 \geq \mathbf{D_p}(\tau) + 1 > |\tau|^4.$$

Concluímos que pelo menos um dos IAT-intervalos contém uma repetição de conclusões.

Seja então I_j o último intervalo com uma repetição minimal $< p, p + r >$. Seja v uma variável que ocorre livre em B_{p+r} e que não ocorre livre em B_p com atribuição de tipo $v : \alpha \in \mathcal{B}_{B_{p+r}}$. Como v não ocorre livre em B_p, v ocorre em $\mathsf{IA}(N_h)$ para algum h com $p + 1 \leq h \leq p + r$. Resulta da definição de IAT-intervalos, que $\mathsf{IAT}(N_h)$ coincide com um dos $\mathsf{IAT}(N_{d_0}), \ldots, \mathsf{IAT}(N_{d_j})$; digamos $\mathsf{IAT}(N_h) = \mathsf{IAT}(N_{d_q})$ para algum $q \leq j \leq p < p + r$. Então, existe uma declaração $v' : \alpha \in \mathcal{B}_{B_{d_q}} \subseteq \mathcal{B}_{B_p}$.

Seja B'_{p+r} um termo obtido de B_{p+r} por substituição das variáveis livres v em B_{p+r} que não ocorrem livres em B_p por alguma variável v' como acima. Finalmente, seja M' o termo obtido a partir de M por substituição de B_p por B'_{p+r}. Note-se que $\mathcal{B}_{B_p} \subseteq \mathcal{B}_{B_{p+r}}$, i.e. $\mathcal{B}_{B_{p+r}} = \mathcal{B}_{B_p} \cup \mathcal{B}_r$ para algum $\mathcal{B}_r$ com $\mathcal{B}_{B_p} \cap \mathcal{B}_r = \emptyset$ e tal que nenhuma variável em $Subjects(\mathcal{B}_r)$ ocorre em B'_{p+r}. Consideremos a sub-árvore Δ_{p+r} de $\Delta_{(\emptyset,\tau_{\mathtt{i}})}$ com raíz $\mathcal{B}_{B_{p+r}} \vdash^{pln} B_{p+r} : a_{b_{p+r}} \parallel \emptyset$. Seja Δ'_{p+r} obtido de Δ_{p+r} por substituição de cada variável v pela variável correspondente $v' \in Subjects(\mathcal{B}_{B_p})$ e por omissão nas bases das fórmulas de todas as declarações para variáveis em $Subjects(\mathcal{B}_r)$. Então, Δ'_{p+r} tem raíz $\mathcal{B}_{B_p} \vdash^{pln} B'_{p+r} : a_{b_{p+r}} \parallel \emptyset$. Consideremos agora a árvore obtida de $\Delta_{(\emptyset,\tau_{\mathtt{i}})}$ por substituição da sua sub-árvore com raíz $\mathcal{B}_{B_p} \vdash^{pln} B_p : a_{b_p} \parallel \emptyset$ por Δ'_{p+r} e por substituição de B_p em todos os nós abaixo por B'_{p+r}. Mostra-se então que a árvore resultante Δ' é uma $\mathbf{TA_{pln}}$-dedução de $\vdash^{pln} M' : \tau_{\mathtt{i}}$ que corresponde à $\mathbf{TA}$-dedução de $\vdash M' : \tau$ e tal que para o conjunto Φ' de pares em Δ' se tem $\mathcal{C}_{\Phi'} = \mathcal{C}_{\Phi}$ e por outro lado todas as setas com ocorrências marcadas em $\Delta_{(\emptyset,\tau_{\mathtt{i}})}$ têm também ocorrências marcadas em Δ'. Assim, concluímos que M' é um habitante principal normal longo de τ. Por outro lado, removemos $r \leq |\tau|^3$ argumentos de uma cadeia em M. Logo, $d - |\tau|^3 \leq Depth(M') \leq d$. Se $Depth(M') < d$ seja $M^\star = M'$. Caso contrário, continuamos a encurtar cadeias de comprimento d enquanto existirem. Definimos $M^\star$ como o primeiro termo produzido por este processo com profundidade menor que d. Então $d - |\tau|^3 \leq Depth(M^\star) < d$. Para a parte *ii.* basta repetir *i.* e tomar o primeiro resultado com profundidade $< \mathbf{D_p}(\tau)$. •

Lema 4.27 *Se* $\mathsf{Lprinc}(\tau)$ *tiver um elemento* M *com profundidade* $\geq \mathbf{d_p}(\tau)$*, então*

i. existe $M^\star \in \mathsf{Lprinc}(\tau)$ *com* $Depth(M^\star) \geq Depth(M) + 1$*;*

ii. $\mathsf{Lprinc}(\tau)$ *é infinito.*

Prova A parte *ii.* resulta de *i.* por repetição. Para $M^\star$ em *i.* consideremos uma qualquer cadeia de argumentos $< N_0, \ldots, N_d >$ de comprimento $d = Depth(M) \geq |\tau|$ e como na prova do lema anterior seja $\underline{B_i}$ o corpo de $\underline{N_i}$ para $i = 0, \ldots, d$. Seja ainda $\mathcal{B}_{B_i} \vdash B_i : a_{b_i} \parallel \emptyset$ o nó correspondente em $\Delta_{(\emptyset,\tau_{\mathtt{i}})}$. Como $d + 1 > |\tau| = ||\tau_{\mathtt{i}}||$, existem $0 \leq p < p + r \leq d$ tais que $\underline{B_p}$ e $\underline{B_{p+r}}$ têm o mesmo tipo $a = a_{b_p} = a_{b_{p+r}}$. Definimos $M^\star$ como resultado obtido por substituição de $\underline{B_{p+r}}$ em M por uma cópia de $\underline{B_p}$, depois de (para evitar conflitos) mudar os nomes das variáveis $x_1, \ldots, x_m$ nesta cópia para $x'_1, \ldots, x'_m$, onde $\{x_1 : \alpha_1, \ldots, x_m : \alpha_m\} = \mathcal{B}_{B_{p+r}} \setminus \mathcal{B}_{B_p}$ bem como todas as outras variáveis ligadas v para v'. Então, $M^\star$ tem uma cadeia de argumentos de comprimento $d + r > d$. Para verificar que τ é um tipo principal de $M^\star$, consideremos a árvore Δ'_{p+r} obtida a partir da sub-árvore de $\Delta_{(\emptyset,\tau_{\mathtt{i}})}$ com raíz $\mathcal{B}_{B_p} \vdash B_p : a \parallel \emptyset$ substituindo primeiro em todos os nós as ocorrências de $x_1, \ldots, x_m$ respectivamente por $x'_1, \ldots, x'_m$ bem como todas as outras variáveis ligadas v por v'. A seguir juntamos $\{x_1 : \alpha_1, \ldots, x_m : \alpha_m\}$ às bases em todos os nós. Note-se que Δ'_{p+r} tem raíz $\mathcal{B}_{B_{p+r}} \vdash B'_p : a \parallel \emptyset$. Aqui B'_p é o termo obtido a partir de B_p por substituição de $x_1, \ldots, x_m$ respectivamente por $x'_1, \ldots, x'_m$ bem como de todas as outras variáveis ligadas v por v'. Finalmente, seja Δ' a árvore obtida de $\Delta_{(\emptyset,\tau_{\mathtt{i}})}$ por substituição da sua sub-árvore com raíz $\mathcal{B}_{B_{p+r}} \vdash B_{p+r} : a \parallel \emptyset$ por Δ'_{p+r}, e por substituição de todos os nós abaixo B_{p+r} por B'_p. É fácil ver que Δ' é a $\mathbf{TA_{pln}}$-dedução que corresponde à $\mathbf{TA}$-dedução de $\vdash M^\star : \tau$ e que todos os pares binários e setas marcadas em $\Delta_{(\emptyset,\tau_{\mathtt{i}})}$ ocorrem também em Δ'. Concluímos que $M^\star \in \mathsf{Lprinc}(\tau)$. •

Sempre que τ é um tipo composto, tem-se $|\tau| \geq 2$, donde

$$\mathbf{d_p}(\tau) = |\tau| < |\tau|^4 - |\tau|^3 = \mathbf{D_p}(\tau) - |\tau|^3 < \mathbf{D_p}(\tau).$$

Logo, obtemos o teorema 4.8 como corolário.

5 Habitantes (principais) em λI

Como já foi dito vamos definir algoritmos análogos ao anterior para contar $\mathsf{Nhabs}^{\mathsf{I}}(\tau)$ e $\mathsf{Nprinc}^{\mathsf{I}}(\tau)$, baseados respectivamente na contagem de $\mathsf{Long}^{\mathsf{I}}(\tau)$ e de $\mathsf{Lprinc}^{\mathsf{I}}(\tau)$.

Para isso, definimos

$$\mathbf{d^I}(\tau) = ||\tau|| \times 2^{|\tau|-1} \qquad \text{e} \qquad \mathbf{D^I}(\tau) = |\tau| \times ||\tau|| \times 2^{|\tau|-1}$$

e ainda

$$\mathbf{d^I_p}(\tau) = |\tau| \times 2^{|\tau|-1} \qquad \text{e} \qquad \mathbf{D^I_p}(\tau) = |\tau|^4 \times 2^{|\tau|-1}.$$

Lema 5.1 *Se* $\mathsf{Long}^{\mathsf{I}}(\tau)$ *tiver um elemento* M *com profundidade* $\geq \mathbf{D^I}(\tau)$*, então*

i. existe $M^\star \in \mathsf{Long}^{\mathsf{I}}(\tau)$ *com* $Depth(M) - ||\tau|| \times 2^{|\tau|-1} \leq Depth(M^\star) < Depth(M)$*;*

ii. existe $N \in \mathsf{Long}^{\mathsf{I}}(\tau)$ *com* $\mathbf{D^I}(\tau) - ||\tau|| \times 2^{|\tau|-1} \leq Depth(N) < \mathbf{D^I}(\tau)$.

Prova Para a parte *i.* consideremos um λ-termo $M \in \mathsf{Long}^{\mathsf{I}}(\tau)$ com profundidade $d \geq \mathbf{D^I}(\tau)$ e sem conflitos de variáveis ligadas. Seja Δ a **TA**-dedução de $\vdash M : \tau$. Seja ainda $< N_0, \ldots, N_d >$ uma cadeia de argumentos em M de comprimento d com $N_i = \lambda x_{i,1} \ldots x_{i,m_i}.y_i P_{i,1} \ldots P_{i,n_i}$ para $0 \leq i \leq d$. Seja $\mathcal{B}_i \vdash N_i : \rho_i = \rho_{i,1} \to \ldots \to \rho_{i,m_i} \to a_i$ o nó em Δ que corresponde a $\underline{N_i}$, para $i = 0, \ldots, d$. Seja $\underline{B_i}$ o corpo de $\underline{N_i}$ e seja $\mathcal{B}_{B_i} \vdash B_i : a_{b_i}$ o nó em Δ que corresponde a $\underline{B_i}$. Consideremos uma sequência de declarações definida da maneira seguinte.

$$\mathcal{VT}_0 = \{x_{0,1} : \rho_{0,1}, \ldots, x_{0,m_0} : \rho_{0,m_0}\}.$$

Para N_i como acima e $N_{i+1} = P_{i,k} = \lambda x_{i+1,1} \ldots x_{i+1,m_{i+1}}.y_{i+1} P_{i+1,1} \ldots P_{i+1,n_{i+1}}$, seja

$$\begin{array}{rl} \mathcal{VT}_{i+1} = & (\mathcal{VT}_i \cup \{x_{i+1,1} : \rho_{i+1,1} \ldots x_{i+1,m_{i+1}} : \rho_{i+1,m_{i+1}}) \setminus \\ & \{z : \sigma \,|\, z \in FV(y_i, P_{i,1}, \ldots, P_{i,k-1}, P_{i,k+1}, \ldots, P_{i,n_i}) \text{ and } z : \sigma \in \mathcal{VT}_i\}. \end{array}$$

(como consideramos termos sem conflitos de variáveis ligadas, existe para cada variável z no máximo um σ nestas condições!). Então, $Subjects(\mathcal{VT}_i) \subseteq FV(B_i)$, para $i = 0, \ldots, d$, resulta de M ser um λI-termo.

Finalmente, para $0 \leq i \leq d$, seja $\mathcal{T}_i$ o multi-conjunto de tipos obtidos de $\mathcal{VT}_i$ apagando as variáveis (para termos), e seja $\mathcal{V}_i$ o conjunto de variáveis obtido de $\mathcal{VT}_i$ apagando os tipos, i.e. $\mathcal{V}_i = Subjects(\mathcal{VT}_i)$. Note-se, que cada ocorrência de um tipo σ em qualquer $\mathcal{T}_i$ corresponde a uma sub-premissa de τ e que ocorrências distintas de um tipo num dos multi-conjuntos $\mathcal{T}_i$ correspondem a sub-premissas distintas de τ. Como foi mostrado em [16] (Lemma 9E6.3) que o número de sub-premissas de τ é $|\tau| - 1$, concluímos que existem no máximo $2^{|\tau|-1}$ multi-conjuntos distintos entre $\mathcal{T}_0, \ldots, \mathcal{T}_d$.

Definimos a sequência de inteiros $d_0, d_1, \ldots, d_n$, bem como a sequência de IAT-intervalos $\mathsf{I}_0, \ldots, \mathsf{I}_n$, exactamente como na prova do lema 4.26. Mas agora chamamos ao par $< p, p+r >$ com $p, p+r \in \mathsf{I}_j$, para $r \geq 1$ e $1 \leq j \leq n$, uma repetição de conclusões sse

- B_p e B_{p+r} tiverem o mesmo tipo (i.e. $a_p = a_{p+r}$)
- e ainda $\mathcal{T}_p = \mathcal{T}_{p+r}$

Chamamos-lhe minimal se não existir nenhuma outra repetição de conclusões $< p', q' >$ com $p \leq p' < q' \leq p + r$.

Usando um raciocínio análogo ao da prova do lema 4.26, é fácil ver que cada IAT-intervalo sem repetição de conclusões tem $\leq ||\tau|| \times 2^{|\tau|-1}$ elementos e ainda que $r \leq ||\tau|| \times 2^{|\tau|-1}$. Concluímos de $d+1 > |\tau| \times ||\tau|| \times 2^{|\tau|-1}$ que pelo menos um dos IAT-intervalos contém uma repetição de conclusões.

Seja então I_j o último intervalo que contém uma repetição de conclusões minimal $< p, p + r >$. Como $\mathcal{T}_p = \mathcal{T}_{p+r}$, existe uma bijecção $g : \mathcal{V}_{p+r} \longrightarrow \mathcal{V}_p$ tal que $x : \rho \in \mathcal{VT}_{p+r}$

e $g(x) : \rho' \in \mathcal{VT}_p$ implica $\rho = \rho'$. Por outro lado, se v é uma variável que ocorre livre em B_{p+r} com declaração $v : \alpha \in \mathcal{B}_{B_{p+r}}$, então $v \in \mathsf{IA}(\underline{N_0}) \cup \ldots \cup \mathsf{IA}(\underline{N_{p+r}})$. Mais ainda, da definição de IAT-intervalos resulta $\alpha \in \mathsf{IAT}(\underline{N_{d_q}})$ para algum $q \leq j \leq p < p + r$. Logo existe alguma declaração $v' : \alpha \in \mathcal{B}_{B_p}$.

Seja B'_{p+r} obtido de B_{p+r} por substituição de todas as variáveis $v \in \mathcal{V}_{p+r}$ por $g(v)$ e de todas as outras variáveis v em B_{p+r} por alguma variável v' como acima (possivelmente a própria se $v : \alpha \in \mathcal{B}_{B_p}$). Finalmente, consideremos M' obtido de M por substituição de B_p por B'_{p+r}. Note-se que $\mathcal{B}_{B_p} \subseteq \mathcal{B}_{B_{p+r}}$, i.e. $\mathcal{B}_{B_{p+r}} = \mathcal{B}_{B_p} \cup \mathcal{B}_r$ para algum $\mathcal{B}_r$ com $\mathcal{B}_{B_p} \cap \mathcal{B}_r = \emptyset$ e tal que nenhuma variável em $Subjects(\mathcal{B}_r)$ occorre em B'_{p+r}. Consideremos a sub-árvore Δ_{p+r} de Δ com raíz $\mathcal{B}_{B_{p+r}} \vdash B_{p+r} : a_{b_{p+r}}$. Seja Δ'_{p+r} obtido de Δ_{p+r} por substituição de todas as variáveis (para termos) $v \in \mathcal{V}_{p+r}$ por $g(v)$ e de toda a outra variável v (que em B_{p+r} foi substituída por $v' \in Subjects(\mathcal{B}_{B_p})$ para obter B'_{p+r}) por v', e apagando nas bases de cada fórmula todas as declarações para variáveis em $Subjects(\mathcal{B}_r)$. Então Δ'_{p+r} tem raíz $\mathcal{B}_{B_p} \vdash B'_{p+r} : a_{b_{p+r}}$. Finalmente, consideremos a árvore obtida de Δ por substituição da sua sub-árvore com raíz $\mathcal{B}_{B_p} \vdash B_p : a_{b_p}$ por Δ'_{p+r} e por substituição de cada B_p nos nós abaixo por B'_{p+r}. A prova de que a árvore resultande Δ' é uma **TA**-dedução de $\vdash M' : \tau$ e de que M' é um habitante normal longo de τ é directa. Por outro lado, suponhamos que existe uma variável $v \in FV(B_p)$, tal que $N_i = \lambda \ldots v \ldots . B_i$ para algum $i \leq p$ e tal que não há ocorrências livres de v em B_i que não sejam em B_p. Então temos $v \in \mathcal{V}_p$ e consequentemente existe exactamente um $\tilde{v} \in \mathcal{V}_{p+r}$ tal que $g(\tilde{v}) = v$. Da definição de $\mathcal{V}_{p+r}$ concluímos que $\tilde{v} \in FV(B_{p+r})$, donde $v = g(\tilde{v}) \in FV(B'_{p+r})$. Logo, M' é um λI-termo. O restante argumento da prova segue exactamente o da prova do lema 4.26. $\bullet$

Lema 5.2 *Se* $\mathsf{Long}^{\mathsf{I}}(\tau)$ *tiver um elemento* M *com profundidade* $\geq \mathbf{d}^{\mathbf{I}}(\tau)$, *então*

i. existe $M^\star \in \mathsf{Long}^{\mathsf{I}}(\tau)$ *com* $Depth(M^\star) \geq Depth(M) + 1$;

ii. $\mathsf{Long}^{\mathsf{I}}(\tau)$ *é infinito.*

Prova Parte *ii.* resulta de *i.* por repetição. Para a parte *i.* suponhamos que M é um λI-habitante normal de τ e seja Δ a **TA**-dedução de $\vdash M : \tau$. Seja $< N_0, \ldots, N_d >$ uma cadeia de argumentos de comprimento $d = Depth(M) \geq |\tau| \times 2^{|\tau|-1}$ em M e como na prova do lema anterior, seja $\underline{B_i}$ o corpo de $\underline{N_i}$ para $i = 0, \ldots, d$. Seja $\mathcal{B}_{B_i} \vdash B_i : a_{b_i}$ o nó correspondente em Δ. Para $i = 0, \ldots, d$ definimos $\mathcal{VT}_i$, $\mathcal{V}_i$ e $\mathcal{T}_i$ como anteriormente. Como $d + 1 > ||\tau|| \times 2^{|\tau|-1}$, existem $0 \leq p < p + r \leq d$ tais que $\underline{B_p}$ e $\underline{B_{p+r}}$ têm o mesmo tipo $a = a_p = a_{p+r}$ e tais que $\mathcal{T}_p = \mathcal{T}_{p+r}$. Isto garante a existência de uma bijecção $f : \mathcal{V}_p \longrightarrow \mathcal{V}_{p+r}$ tal que $x : \rho \in \mathcal{VT}_p$ e $f(x) : \rho' \in \mathcal{VT}_{p+r}$ implica $\rho = \rho'$. Seja B'_p obtido de B_p mudando primeiro os nomes das variáveis $x_1, \ldots, x_m$ nesta cópia para $x'_1, \ldots, x'_m$, onde $\{x_1 : \alpha_1, \ldots, x_m : \alpha_m\} = \mathcal{B}_{B_{p+r}} \setminus \mathcal{B}_{B_p}$ bem como os de todas as variáveis ligadas v para v' (para evitar conflitos de variáveis ligadas), e por substituição de toda a variável $v \in \mathcal{V}_p$ por $f(v)$. Seja $M^\star$ o resultado obtido por substituição de $\underline{B_{p+r}}$ em M por B'_p. Então $M^\star$ tem uma cadeia de argumentos com comprimento $d + r > d$.

Para verificar que $M^\star$ é um λI-habitante de τ, consideremos a árvore Δ'_{p+r} obtida da sub-árvore Δ_p de Δ com raíz $\mathcal{B}_{B_p} \vdash B_p : a$ por substituição das ocorrências de $x_1, \ldots, x_m$ respectivamente por $x'_1, \ldots, x'_m$ e de qualquer outra varíavel ligada v por v'. Depois, substituímos (somente no lado direito das fórmulas) cada variável $v \in \mathcal{V}_p$ por $f(v)$, e finalmente juntamos $\{x_1 : \alpha_1, \ldots, x_m : \alpha_m\}$ às bases em todos os nós. Seja então Δ' a árvore obtida a partir de Δ por substituição da sua sub-árvore com raíz $\mathcal{B}_{B_{p+r}} \vdash B_{p+r}$ por Δ'_{p+r} e por substituição nos nós abaixo de todas as ocorrências de B_{p+r} por B'_p. A prova de que Δ' é uma **TA**-dedução de $M^\star$ e de que $M^\star$ é fechado e longo para $(\emptyset, \tau)$ é directa. Para verificar que $M^\star$ é um λI-termo, seja x uma variável que ocorre em $M^\star$. Se x é v' para alguma variável v que foi substituída por ocorrer ligada em B_p, então existem (pelo menos uma) ocorrência de v' no alcance da abstracção correspondente $\underline{\lambda v'}$ que corresponde exactamente às ocorrências de v no alcance de $\underline{\lambda v}$ em B_p. Caso contrário, x ocorre também em M e as ocorrências da abstracção λx em M e em $M^\star$ correspondem exactamente uma à outra. Consideremos então o sub-termo correspondente $\lambda x.N$ de M. Se B_{p+r} não é sub-termo de N, então todas as ocorrências livres de x em N ainda existem em $M^\star$. Caso contrário, existe uma ocorrência livre de x em N que não é em B_{p+r} e assim a mesma ocorrência vai aparecer em $M^\star$, ou então $x \in \mathcal{V}_{p+r}$. Mas isto significa que existe $v \in \mathcal{V}_p$ tal que $f(v) = x$ e tal que a ocorrência livre de v em B_p foi substituída por x para obter B'_p. Obviamente esta ocorrência de x em B'_p está no alcance de $\underline{\lambda x}$ em $M^\star$. Concluímos que $M^\star \in \mathsf{Long}^{\mathsf{I}}(\tau)$. $\bullet$

Sempre que τ é um tipo composto, tem-se $|\tau| \geq 2$, donde

$$\mathbf{d}^{\mathbf{I}}(\tau) = ||\tau|| \times 2^{|\tau|-1} < |\tau| \times ||\tau|| \times 2^{|\tau|-1} - ||\tau|| \times 2^{|\tau|-1} = \mathbf{D}^{\mathbf{I}}(\tau) - ||\tau|| \times 2^{|\tau|-1} < \mathbf{D}^{\mathbf{I}}(\tau).$$

Obtemos o teorema seguinte como corolário:

Teorema 5.3 *Para cada tipo τ, tem-se*

i. $\mathsf{Long}^{\mathsf{I}}(\tau) = \emptyset$ *sse* $\mathsf{Long}^{\mathsf{I}}(\tau)$ *não tiver nenhum elemento com profundidade* $< \mathbf{D}^{\mathbf{I}}(\tau)$*;*

ii. $\mathsf{Long}^{\mathsf{I}}(\tau)$ *é infinito sse tiver um elemento M com* $\mathbf{d}^{\mathbf{I}}(\tau) \leq depth(M) < \mathbf{D}^{\mathbf{I}}(\tau)$*;*

iii. $\mathsf{Long}^{\mathsf{I}}(\tau)$ *é finito sse todos os seus elementos com profundidade* $< \mathbf{D}^{\mathbf{I}}(\tau)$ *tiverem profundidade* $< \mathbf{d}^{\mathbf{I}}(\tau)$.

Adaptando as provas desta e da secção anterior mostram-se os dois resultados seguintes.

Lema 5.4 *Se* $\mathsf{Lprinc}^{\mathsf{I}}(\tau)$ *tiver um elemento M com profundidade* $\geq \mathbf{D}^{\mathbf{I}}_{\mathbf{p}}(\tau)$*, então*

i. existe $M^\star \in \mathsf{Lprinc}^{\mathsf{I}}(\tau)$ *com* $Depth(M) - |\tau|^3 \times 2^{|\tau|-1} \leq Depth(M^\star) < Depth(M)$*;*

ii. existe $N \in \mathsf{Lprinc}^{\mathsf{I}}(\tau)$ *com* $\mathbf{D}^{\mathbf{I}}_{\mathbf{p}}(\tau) - |\tau|^3 \times 2^{|\tau|-1} \leq Depth(N) < \mathbf{D}^{\mathbf{I}}_{\mathbf{p}}(\tau)$.

Lema 5.5 *Se* $\mathsf{Lprinc}^{\mathsf{I}}(\tau)$ *tiver um elemento M com profundidade* $\geq \mathbf{d}^{\mathbf{I}}_{\mathbf{p}}(\tau)$*, então*

i. existe $M^\star \in \mathsf{Lprinc}^{\mathsf{I}}(\tau)$ *com* $Depth(M^\star) \geq Depth(M) + 1$*;*

ii. $\mathsf{Lprinc}^{\mathsf{I}}(\tau)$ *é infinito.*

O teorema seguinte resulta como corolário, uma vez que para $|\tau| \geq 2$ se tem

$$\mathbf{d}^{\mathbf{I}}_{\mathbf{p}}(\tau) = |\tau| \times 2^{|\tau|-1} < |\tau|^4 \times 2^{|\tau|-1} - |\tau|^3 \times 2^{|\tau|-1} = \mathbf{D}^{\mathbf{I}}_{\mathbf{p}}(\tau) - |\tau|^3 \times 2^{|\tau|-1} < \mathbf{D}^{\mathbf{I}}_{\mathbf{p}}(\tau).$$

Teorema 5.6 *Para cada tipo τ, tem-se*

i. $\mathsf{Lprinc}^{\mathsf{I}}(\tau) = \emptyset$ *sse* $\mathsf{Lprinc}^{\mathsf{I}}(\tau)$ *não tiver nenhum elemento com profundidade* $< \mathbf{D}^{\mathbf{I}}_{\mathbf{p}}(\tau)$*;*

ii. $\mathsf{Lprinc}^{\mathsf{I}}(\tau)$ *é infinito sse tiver um elemento M com* $\mathbf{d}^{\mathbf{I}}_{\mathbf{p}}(\tau) \leq depth(M) < \mathbf{D}^{\mathbf{I}}_{\mathbf{p}}(\tau)$*;*

iii. $\mathsf{Lprinc}^{\mathsf{I}}(\tau)$ *é finito sse todos os seus elementos com profundidade* $< \mathbf{D}^{\mathbf{I}}_{\mathbf{p}}(\tau)$ *tiverem profundidade* $< \mathbf{d}^{\mathbf{I}}_{\mathbf{p}}(\tau)$.

Baseado nos teoremas 5.3 e 5.6 define-se então os algoritmos de contagem para $\mathsf{Nhabs}^{\mathsf{I}}(\tau)$ e para $\mathsf{Nprinc}^{\mathsf{I}}(\tau)$, como apresentado na secção 4.

6 Conclusões e trabalho futuro

Neste trabalho apresentamos resultados a partir dos quais definimos algoritmos para contar e enumerar efectivamente os habitantes principais normais de um tipo τ no λ-calculus simplesmente tipado e nos seus três sub-sistemas principais. Resolvemos ainda o problema da contagem dos habitantes normais em λI.

Na abordagem do problema de contagem para **BCK**- e **BCI**-habitantes principais normais consideramos primeiro a particularização do algoritmo geral, para calcular pares principais de λ-termos, para o caso dos **BCK**-termos em forma normal. A partir deste conseguimos então definir um algoritmo que para cada par decide se este é um par principal para algum **BCK**-termo em forma normal, calculando-o (módulo-α) se este for o caso. Além de resolver o problema da contagem, concluímos ainda da unicidade do resultado deste algoritmo, que a igualdade módulo-α de dois termos normais em **BCK** equivale à igualdade dos seus pares principais (a menos do nome das variáveis). Estamos neste momento a fazer um estudo mais pormenorizado do algoritmo de inferência do tipo principal que definimos para o **BCK**-calculus e o **BCI**-calculus. Note-se que estes dois subsistemas estão, através da correspondência de Curry-Howard, directamente relacionados com dois sistemas de cálculo proposicional minimal, nomeadamente o cálculo-**BCK** e o cálculo-**BCI**. Em [11], [21], [20], [7], [23] e [19] encontram-se alguns dos estudos mais importantes sobre estes dois cálculos. Resultados obtidos por Harrop em [12] e [13] mostram que o problema da completude axiomática para conjuntos finitos de fórmulas, em qualquer um dos dois cálculos, é indecidível. Mostramos, para **BCK** e **BCI** (cf. [2]), que o problema da completude axiomática para um conjunto de fórmulas é equivalente ao problema da completude combinatória, no sistema de λ-calculus correspondente, de um conjunto de λ-termos fechados, ou combinadores, para os quais as fórmulas, interpretadas como tipos, são tipos principais. O problema de completude combinatória para conjuntos de combinadores é um problema em aberto, nos dois sistemas de λ-calculus, mas mostramos em [1] a sua decidibilidade para um subconjunto de termos fechados especiais, os combinadores próprios. Pensamos por isso que uma caracterização dos tipos principais, que esperamos obter a partir do algoritmo apresentado, nos irá por um lado permitir identificar um subconjunto de fórmulas (as que correspondem a tipos principais de combinadores próprios) para os quais o problema de completude axiomática é decidível. Por outro lado, esperamos com esta caracterização e da prova de Harrop obter mais alguns resultados quanto à decidibilidade dos respectivos problemas de completude combinatória.

Outro projecto em curso que dá continuação ao trabalho apresentado, é o de estender o sistema $\mathbf{TA_{pln}}$. Definimos este com o objectivo de obter uma caracterização simples e, o que é mais importante, útil de deduções principais para habitantes normais longos. Neste momento não existe uma caracterização razoável para o caso geral (em [17] encontram-se alguns resultados parciais para o **BCK**-calculus) que esperamos obter a partir de uma extensão adequada de $\mathbf{TA_{pln}}$.

Referências

[1] Broda, S., & Damas, L.. On combinatory complete sets of proper combinators. *Journal of Functional Programming* **7 (6)**, pág. 593-612, 1997.

[2] Broda, S., & Damas, L.. On principal types of combinators. Aceite para publicação em *Journal of Theoretical Computer Science.*

[3] Broda, S., & Damas, L.. Counting a type's principal inhabitants. Proceedings of the 4th International Conference on Typed Lambda Calculi and Applications, *TLCA'99*, *Lecture Notes in Computer Science* 1581, pág. 69-82, Springer Verlag, 1999.

[4] Broda, S., & Damas, L.. Counting a type's (principal) inhabitants. Aceite para publicação em *Fundamenta Informaticae*.

[5] Barendregt, H.. Lambda calculi with types. In Abramsky, Gabbay, and Maibaum, editors, *Background: Computational Structures*, volume 2 of *Handbook of Logic in Computer Science*, pages 117–309. Oxford Science Publications, 1992.

[6] Ben-Yelles, C. B.. *Type-assignment in the lambda-calculus; syntax and semantics.* PhD thesis, Mathematics Dept., University of Wales Swansea, UK, 1979.

[7] Bunder, M.W., & Meyer, R.K.. A result for combinators, BCK-logics and BCK-algebras. *Logique et Analyse*, **109**, 33–40, 1985.

[8] Church, A.. A formulation of the simple theory of types. *Journal of Symbolic Logic*, **5**, 56–68, 1940.

[9] Curry, H. B.. Modified basic functionality in combinatory logic. *Dialectica*, **23**, 83–92, 1969.

[10] Curry, H. B.. Functionality in combinatory logic. *Proc. Nat. Acad. Science USA*, **20**, 584–590, 1934.

[11] Fitch, F. B.. A system of formal logic without an analogue to the Curry W operator. *Journal of Symbolic Logic*, **1**, 92–100, 1936.

[12] Harrop, R.. A relativization procedure for propositional calculi with an application to a generalized form of Post's theorem. *Proc. London Math. Soc.*, **3**, 595–617, 1964.

[13] Harrop, R.. Some generalizations and applications of a relativization procedure for propositional calculus. Pag. 12–41 in: Crossley, J. N., & Dummett, M. A. E. (eds), *Formal Systems and Recursive Functions.* North Holland, 1965.

[14] Hindley, J. R.. The Principal Type-scheme of an Object in Combinatory Logic. *Transactions of the American Mathematical Society*, **146**, 29–60, 1969.

[15] Hindley, J. R.. BCK-combinators and linear λ-terms have types. *Theoretical Computer Science*, **64**, 97–105, 1989.

[16] Hindley, J. R.. *Basic Simple Type Theory.* Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1997.

[17] Hirokawa, S.. Principal types of BCK-lambda-terms. *Theoretical Computer Science*, **107**, 253–276, 1993.

[18] Hirokawa, S.. Infiniteness of proof(α) is polynomial-space complete. *Theoretical Computer Science*, 206:331–339, 1998.

[19] Komori, Y.. BCK algebras and lambda calculus. Pag. 5–11 in: *Proc. 10th Symp. on Semigroups*, 1986.

[20] Meredith, C. A.. Positive Logic and λ-constants. *Studia Logica*, **3**, 267–285, 1978.

[21] Meredith, C. A., & Prior, A. N.. Notes on the axiomatics of the propositional calculus. *Notre Dame J. Formal Logic*, **4**, 171–187, 1963.

[22] Milner, R.. A theory of type polymorphism in programming. *J. Computer and Systems Sciences*, **17**, 348–375, 1978.

[23] Ono, H., & Komori, Y.. Logics without contraction rule. *Journal of Symbolic Logic*, **50**, 169–201, 1985.

[24] Peyton Jones, S. L., Lestes, D.. *Implementing Functional Programming Languages: A Tutorial.* Prentice-Hall International, 1992.

[25] Peyton Jones, S. L. et al.. *The Glasgow Haskell compiler: a technical overview.* In *Proceedings of UK Joint Framework for Information Technology*, Dep. of Computing Science, University of Glasgow, 1992.

[26] Takahashi, M., Akama, Y. and Hirokawa, S.. Normal proofs and their grammar. In M. Hagiya and J. Mitchell, editors, *Theoretical Aspects of Computer Software (TACS'94)*, volume 789 of *Lecture Notes in Computer Science*, pages 465–493. Springer Verlag, 1994.

[27] Zaionc, M.. The set of unifiers in typed λ-calculus as regular expression. In J.-P. Jouannaud, editor, *Rewriting Techniques and Applications*, volume 202 of *Lecture Notes in Computer Science*, pages 430–440. Springer Verlag, 1985.

[28] Zaionc, M.. Mechanical procedure for proof construction via closed terms in typed λ-calculus. *J. Automated Reasoning*, 4:173–190, 1988.

[29] Zaionc, M.. Fixpoint Technique for Counting Terms in Typed Lambda Calculus. Technical Report 95-24, Dep. of Computer Science, State University of New York at Buffalo, 1995.

[30] Zaionc, M.. Counting the Number of Closed Terms in Typed Lambda Calculus. Aceite para publicação em *Fundamenta Informaticae.*