Sabine Babette Broda

Sobre Bases de Combinadores para Sistemas de λ-Calculus

Departamento de Ciência de Computadores

Faculdade de Ciências da Universidade do Porto

1996

Sabine Babette Broda

Sobre Bases de Combinadores para Sistemas de λ-Calculus

Tese submetida para obtenção
do grau de Doutor em Ciência de Computadores

Departamento de Ciência de Computadores
Faculdade de Ciências da Universidade do Porto

1996

Aos meus pais.

Agradecimentos

Em primeiro lugar quero agradecer ao meu orientador, Luís Damas, por todo o apoio que me deu durante o desenvolvimento deste trabalho. Os seus profundos conhecimentos nesta área, as suas valiosas sugestões e a constante disponibilidade para debater os diversos aspectos relacionados com os assuntos que abordei nesta tese, foram fundamentais para a sua execução.

Aos meus colegas do Departamento de Ciência de Computadores agradeço o bom ambiente de trabalho, a amizade e a ajuda que me deram ao longo dos últimos anos. Em particular quero agradecer à Nelma Moreira, ao Armando Campos e Matos, ao Miguel Filgueiras e à Ana Paula Tomás por terem lido partes de trabalhos preliminares, que deram origem a esta dissertação, e ao Fernando Silva por me ter disponibilizado as suas macros de LaTeX para a geração da estrutura da tese. Quero expressar a minha particular gratidão ao Miguel Filgueiras pela paciência com que leu esta dissertação, e pela disponibilidade e a ajuda constante, e também ao Zé Paulo por estar sempre pronto para resolver os meus problemas informáticos.

Agradeço ao Departamento de Matemática Pura pelo apoio e a boa vontade manifestados durante a minha permanência nesse Departamento.

Um agradecimento especial aos meus pais por estarem sempre ao meu lado. Ao Nick pela boa companhia e pelo grande apoio nos últimos tempos.

Índice

Capítulo 1

Introdução

1.1 Nota histórica

O λ-calculus e a sua variante, a lógica combinatória, surgiram no início deste século, respectivamente com os artigos de Alonzo Church, (Church, 1932) e (Church, 1933), e com os artigos de Moses Schönfinkel e Haskell Curry, (Schönfinkel, 1924), (Curry, 1929) e (Curry, 1930). Nesta altura apareceram como partes de teorias gerais de funções e lógicas de ordem superior, cujo intento era o de servirem como fundamento para a matemática. Foi depois de Stephen Kleene e Barkley Rosser, (Kleene & Rosser, 1935), terem mostrado em 1935 a inconsistência destas teorias, que Church abandonou o programa de fundamentos e extraiu a subteoria referente à parte funcional, o λ-calculus, (Church, 1941)[1], cuja consistência decorria de resultados expostos em (Church & Rosser, 1936). Para a lógica combinatória pura, i.e. o subsistema da teoria apresentada por Curry restrito a combinadores, a consistência já tinha sido provada por Curry, (Curry, 1930), que passou a dedicar-se ao desenvolvimento de teorias consistentes, conhecidas por sistemas de lógica combinatória ilativa e descritas em (Curry & Feys, 1958) e (Curry *et al.*, 1972), sem no entanto atingir o objectivo final de fornecer fundamentos adequados para a matemática. Os sistemas *puros*, suficientemente poderosos para representar a classe de funções recursivas (parciais), cf. (Kleene & Rosser, 1935), tinham-se no entanto tornado alvo de investigações intensas e ainda o são até aos dias

[1] O sistema que Church descreve neste artigo é de facto uma subteoria do λ-calculus, nomeadamente o $\lambda\mathcal{I}$-calculus.

de hoje. Exposições de estudos das duas teorias e da sua relação podem ser encontradas em (Barendregt, 1981), (Curry & Feys, 1958), (Hindley & Seldin, 1986) e (Krivine, 1990). Além da sua importância para diversos campos, como a teoria da demonstração e a computação[2], desempenharam um papel importante na obtenção dos primeiros resultados sobre funções recursivas; em particular foi através do λ-calculus que se mostrou originalmente a equivalência das classes de funções recursivas e de funções computáveis por máquinas de Turing.

O λ-calculus é uma teoria que modela funções e o seu comportamento aplicativo, tendo por isso a aplicação como operação básica. Dados uma função f e um argumento a representa-se o resultado da aplicação de f a a por fa. Inversamente existe uma operação de abstracção; se $t(x)$ for uma expressão, contendo possivelmente a variável x, então representa-se por $\lambda x.t(x)$ a função que associa a cada argumento a o valor $t(a)$. A substituição é consequentemente um dos conceitos centrais do λ-calculus, estando no entanto completamente ausente nos sistemas da lógica combinatória. Esta baseia-se no facto de se poder representar qualquer λ-termo fechado a partir de um número finito de termos particulares por aplicações sucessivas, como já tinha sido reparado por Schönfinkel e Curry. A lógica combinatória, que também pode ser vista como variante do λ-calculus, é então uma teoria baseada num conjunto de constantes básicas, os combinadores primitivos, e na operação aplicação, enquanto que a abstracção se define a partir dos combinadores. À propriedade de se poder representar qualquer função algébrica por um λ-termo fechado dá-se o nome de completude combinatória e os λ-termos fechados designam-se também por combinadores. Um conjunto de combinadores a partir do qual se geram todos os λ-termos fechados por aplicações sucessivas chama-se ainda uma base para o λ-calculus.

1.2 Resumo da dissertação

O tema central desta dissertação é o conceito de base de combinadores, que desempenha um papel fundamental no λ-calculus. Assim, obtêm-se, por exemplo, enumerações do conjunto de todos os λ-termos fechados a partir de bases de combinadores, que também encontram aplicação na implementação de linguagens funcionais. Conhece-se um vasto número de bases

[2] Nestes contextos utilizam-se geralmente versões tipadas dos dois sistemas.

para o λ-calculus, ver (Barendregt, 1981), (Abdali, 1976) e (Turner, 1979a), bem como para muitos dos seus subsistemas, ver (Barendregt, 1981), (Hindley, 1989) e (Trigg *et al.*, 1994). Um dos problemas que se põe é o da decidibilidade da completude combinatória, i.e. o de decidir se um conjunto particular de combinadores forma ou não uma base para o λ-calculus ou um dos seus subsistemas. Richard Statman mostrou em 1986, (Statman, 1986), que este problema é em geral indecidível. Contudo não se sabe se o mesmo se verifica para subsistemas mais restritos, e também para bases de combinadores sob forma especial, nomeadamente para os combinadores próprios. Estes desempenham um papel importante, pois qualquer combinador pode ser representado por uma combinação de combinadores próprios. Em (Curry & Feys, 1958) William Craig expõe um estudo sobre a geração de λ-termos a partir de combinadores próprios, donde resulta uma interessante lista de condições necessárias sobre bases (de subsistemas) do λ-calculus. Outras condições, também necessárias, foram apresentadas por Statman, (Statman, 1986).

Depois de introduzir no SEGUNDO CAPÍTULO da dissertação os conceitos e os resultados básicos do λ-calculus e da lógica combinatória, dedicamos o terceiro e o quarto capítulo ao estudo da completude combinatória em três subsistemas naturais do λ-calculus, nomeadamente no $\lambda\mathcal{I}$-calculus, no λ-calculus linear e no λ-calculus afim.

No TERCEIRO CAPÍTULO introduzimos o conceito de domínio aplicativo, o que nos permite obter representações alternativas para termos aplicativos e combinadores próprios (lineares e afins). Utilizando estas representações mostramos que a completude combinatória no λ-calculus linear e no λ-calculus afim é decidível para conjuntos de combinadores próprios. Em ambos os casos apresentamos provas construtivas, no sentido de fornecerem algoritmos para a representação de λ-termos lineares (resp. afins) a partir dos elementos de qualquer base de combinadores próprios. A partir dos resultados obtidos estabelecemos ainda condições necessárias, facilmente verificáveis, para bases de combinadores em geral.

No QUARTO CAPÍTULO fazemos um estudo para o $\lambda\mathcal{I}$-calculus. Adaptamos a prova de indecidibilidade apresentada por Statman para o λ-calculus puro, (Statman, 1986), ao sistema $\lambda\mathcal{I}$, obtendo assim o mesmo resultado para este sistema. A seguir apresentamos uma caracterização simples das extensões de bases lineares com combinadores próprios para bases de $\lambda\mathcal{I}$. Finalmente generalizamos as representações de combinadores próprios lineares e afins, introduzidas no terceiro capítulo, para o $\lambda\mathcal{I}$-calculus, e obtemos uma condição necessária sobre as bases de combinadores próprios de $\lambda\mathcal{I}$.

Existem também sistemas tipados do λ-calculus e da lógica combinatória, essencialmente introduzidos em (Church, 1940) e (Curry, 1934), aos quais dedicamos o QUINTO CAPÍTULO da dissertação. Consideramos aqui as versões simplesmente tipadas à la Curry do λ-calculus puro e dos três subsistemas referidos anteriormente. Estes estão, via isomorfismo de Curry-Howard, fortemente relacionados com quatro cálculos proposicionais minimais. Depois de introduzir brevemente os conceitos e os resultados principais dos sistemas tipados, dos respectivos cálculos proposicionais e da relação existente entre eles, mostramos que em qualquer um dos três subsistemas tipados se verifica que um conjunto de combinadores é uma base se e só se os tipos dos combinadores, quando interpretados como fórmulas, formarem, juntamente com as regras de inferência modus ponens e substituição, um conjunto completo de axiomas para o respectivo sistema do cálculo proposicional minimal. Este resultado não é, por si só, muito profundo e já tinha sido referido para a versão tipada do λ-calculus puro em (Statman, 1986). Contudo permite-nos, para cada um dos sistemas considerados, relacionar a completude combinatória de um conjunto de combinadores não tipados e a completude axiomática do conjunto dos tipos principais destes combinadores, quando interpretados como fórmulas. Note-se que um tipo principal de um λ-termo pode ser visto como representante sintáctico de todos os tipos que lhe podem ser atribuídos. Mostramos então que, em qualquer um dos quatro casos, a completude axiomática implica a completude combinatória e que o recíproco só se verifica no λ-calculus linear e no λ-calculus afim. Obtemos ainda uma caracterização dos λ-termos afins em forma normal a partir dos seus tipos principais, donde decorre um resultado curioso para o λ-calculus linear, onde a equivalência módulo-β corresponde à igualdade, a menos de variantes, entre tipos principais.

Algoritmos para traduzir λ-termos para termos combinatórios, e vice-versa, existem desde (Schönfinkel, 1924) e têm aplicação prática na programação funcional; para uma introdução e referências aconselhamos (Barendregt, 1990) e (Peyton Jones, 1987). Na programação funcional, que se baseia na representação de funções computáveis por λ-termos, a execução de programas corresponde à redução de λ-termos. Através de implementações de linguagens funcionais via combinadores evitam-se então problemas técnicos associados ao conceito de substituição, que está completamente ausente na redução de termos combinatórios. Esta abordagem traz, no entanto, outro tipo de problemas, devido ao facto das representações de λ-termos por termos combinatórios serem geralmente muito complexas. Desde 1924 desenvolveram-se diversas bases de combinadores, juntamente com algoritmos para traduzir λ-termos para termos dos respectivos sistemas combinatórios, que se designam por algoritmos de abstracção, pois permitem definir a operação de abstracção a partir de combinadores primitivos. Em (Mulder, 1985) encontra-se um estudo comparativo da complexidade dos termos produzidos por diversos algoritmos de abstracção. No SEXTO CAPÍTULO apresentamos um sistema de combinadores, que denotamos por $\mathcal{C}\ell(\mathcal{K})$, e um algoritmo de abstracção associado, baseado no algoritmo original de Schönfinkel. Este algoritmo novo tem a propriedade de introduzir, quando muito, um combinador para cada abstracção no λ-termo original e permite, deste modo, evitar um crescimento explosivo na tradução de λ-termos, que contêm um número elevado de abstracções, tornando-se assim a redução de $\mathcal{C}\ell(\mathcal{K})$-termos particularmente eficiente. Provamos a correcção do algoritmo, especificamos a complexidade dos termos produzidos e estabelecemos ainda algumas relações entre o sistema combinatório $\mathcal{C}\ell(\mathcal{K})$ e o λ-calculus puro. Apresentamos também três subsistemas de $\mathcal{C}\ell(\mathcal{K})$ que correspondem respectivamente ao $\lambda\mathcal{I}$-calculus, ao λ-calculus linear e ao λ-calculus afim.

Os resultados para os quais apresentamos demonstrações nesta dissertação são originais, com excepção da proposição 5.5.3, e foram parcialmente expostos em (Broda & Damas, 1995), (Broda & Damas, 1996a) e (Broda & Damas, 1996b).

Capítulo 2

Preliminares

Neste capítulo introduzimos brevemente os conceitos básicos do λ-calculus e da lógica combinatória de modo a fixar a notação e tornar a leitura da dissertação independente de outros textos. Neste sentido incluímos também alguns dos resultados principais das duas teorias, aos quais recorreremos ao longo dos capítulos subsequentes. O leitor interessado pode encontrar as provas respectivas, bem como exposições extensas da matéria abordada em diversos livros da vasta literatura existente nesta área, dos quais aconselhamos (Barendregt, 1981), (Hindley & Seldin, 1986), (Curry & Feys, 1958) e (Krivine, 1990).

2.1 Lambda calculus

O λ-calculus é um cálculo funcional, onde o conceito de função é visto como regra, ou seja como o processo da passagem de um argumento para um valor. Esta perspectiva, contrária à noção de Dirichlet de encarar uma função como grafo, i.e. como conjunto de pares de argumentos e valores respectivos, é importante para o estudo de computabilidade e a teoria da computação em geral, pois sublinha o aspecto computacional associado à noção de função.

Em seguida apresentamos o λ-calculus puro, começando por especificar o seu conjunto de λ-termos.

Definição 2.1.1 *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 geralmente as abreviações

$$\lambda x_1 \ldots x_n.M \qquad \text{e} \qquad MN_1 \ldots N_n$$

respectivamente para

$$(\lambda x_1(\lambda x_2(\ldots(\lambda x_n M)\ldots))) \qquad \text{e} \qquad (\ldots((MN_1)N_2)\ldots N_n).$$

Definição 2.1.2 *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 $FV(M) = \emptyset$. O conjunto de todos os λ-termos fechados designa-se por $\Lambda^0 \subseteq \Lambda$. 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 subtermo 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.

Exemplo 2.1.3 *Tem-se* $(\lambda y.xy)[y/x] = \lambda z.yz$, *já que* x *não é substituível por* y *em* $\lambda y.xy$ *e* $\lambda y.xy =_\alpha \lambda z.xz$.

Definição 2.1.4 *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. Se existir um termo* $N \in \Lambda$ *em forma normal, tal que* $M \rightarrow_\beta N$, *então diz-se que* M *admite forma normal-*β.

Lema 2.1.5 *Todo o* λ*-termo em forma normal é da forma*

$$\lambda x_1 \ldots x_n.yN_1 \ldots N_m$$

com $n, m \geq 0$, $y \in \mathcal{V}$ *e tal que* $N_1, \ldots, N_m$ *estão em forma normal.*

Exemplo 2.1.6

1. *O termo* $(\lambda x.xx)(\lambda x.xx)$ *não admite forma normal-*β *uma vez que*

   $$(\lambda x.xx)(\lambda x.xx) \rightarrow_{1\beta} (\lambda x.xx)(\lambda x.xx) \rightarrow_{1\beta} (\lambda x.xx)(\lambda x.xx) \rightarrow_{1\beta} \ldots$$

2. *O termo* $(\lambda xy.x)(\lambda x.x)((\lambda x.xx)(\lambda x.xx))$ *admite a forma normal* $\lambda x.x$, *mas nem toda a sequência de reduções-*β *leva a essa forma normal.*

Intuitivamente interpreta-se um termo da forma $\lambda x.M$ como função de parâmetro formal x. Assim $(\lambda x.M)N$ representa a aplicação dessa função ao argumento N e $M[N/x]$ é a substituição do parâmetro formal pelo argumento. A forma normal de $M[N/x]$ é vista como o resultado da computação. O resultado seguinte mostra que esta interpretação intuitiva é correcta, i.e. se uma computação termina, então o resultado é único.

Teorema 2.1.7 (Church-Rosser) *Se* $M \twoheadrightarrow_\beta N_1$ *e* $M \twoheadrightarrow_\beta N_2$*, então existe um termo* P *tal que* $N_1 \twoheadrightarrow_\beta P$ *e* $N_2 \twoheadrightarrow_\beta P$*.*

Corolário 2.1.8 *Se* M *admite uma forma normal-*β*, então essa forma normal é única.*

Definição 2.1.9 *O símbolo principal de um redex* $(\lambda x.M)N$ *é o primeiro* λ*. Diz-se que uma ocorrência de um redex* R_1 *num termo* M *é à esquerda de uma ocorrência de um redex* R_2 *em* M*, se o símbolo principal de* R_1 *ocorrer à esquerda do símbolo principal de* R_2*. Escreve-se* $M \rightarrow_{1l\beta} N$ *se* N *resultar de* M *por contracção do seu redex mais à esquerda. A relação* $\twoheadrightarrow_{l\beta}$ *é o fecho reflexivo e transitivo de* $\rightarrow_{1l\beta}$*.*

Uma redução diz-se em ordem normal se a cada passo, o redex que é escolhido para a redução for o mais à esquerda no termo.

Teorema 2.1.10 *Se* M *admite uma forma normal* N*, então existe uma redução em ordem normal de* M *a* N*.*

Definição 2.1.11 *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* $\rightarrow_{1\eta}$*,* $\twoheadrightarrow_\eta$ *e* $=_\eta$ *definem-se então analogamente às relações* $\rightarrow_{1\beta}$*,* $\twoheadrightarrow_\beta$ *e* $=_\beta$*, cf. definição 2.1.4. Por um* $\beta\eta$*-redex entende-se um* β*-redex ou um* η*-redex com definições óbvias das relações* $\rightarrow_{1\beta\eta}$*,* $\twoheadrightarrow_{\beta\eta}$ *e* $=_{\beta\eta}$*.*

2.1.1 Subsistemas

Restringindo o conjunto de λ-termos definem-se diversos subsistemas do λ-calculus puro, dos quais apresentamos três, nomeadamente o $\lambda\mathcal{I}$-calculus, o λ-calculus linear e o λ-calculus afim. Estes três sistemas podem ser obtidos a partir do λ-calculus puro, impondo condições simples sobre o número de ocorrências das variáveis nos termos. Por outro lado, relacionam-se, via isomorfismo de Curry-Howard, directamente com três sistemas de cálculo proposicional conhecidos, como veremos no capítulo 5, o que torna o seu estudo particularmente interessante.

No $\lambda\mathcal{I}$-calculus cada subtermo 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 2.1.12 ($\lambda\mathcal{I}$-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 λ-calculus linear cada variável ocorre no máximo uma vez livre em todo o subtermo de um termo M, e para cada subtermo da forma $\lambda x.N$ de M, a variável x tem exactamente uma ocorrência livre em N.

Definição 2.1.13 (λ-calculus linear) *O conjunto de λ-termos lineares $\Lambda_{\mathcal{L}}$ é definido por:*

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

No λ-calculus afim nenhuma variável x ocorre mais do que uma vez livre em nenhum subtermo de um termo M.

Definição 2.1.14 (λ-calculus afim) *O conjunto de λ-termos afins $\Lambda_{\mathcal{A}}$ é definido por:*

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

As restantes noções definidas para o λ-calculus puro definem-se de forma análoga nos três subsistemas. Os respectivos conjuntos de termos fechados designam-se por $\Lambda^0_{\mathcal{I}}$, $\Lambda^0_{\mathcal{L}}$ e $\Lambda^0_{\mathcal{A}}$.

$$\begin{tikzcd} & \Lambda_{\mathcal{K}} \arrow[dr, dash, "\supsetneq"] & \\ \Lambda_{\mathcal{A}} \arrow[ur, dash, "\subsetneq"] \arrow[dr, dash, "\supsetneq"'] & & \Lambda_{\mathcal{I}} \\ & \Lambda_{\mathcal{L}} \arrow[ur, dash, "\subsetneq"'] & \end{tikzcd}$$

Figura 2.1.1

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 conjuntos $\Lambda_{\mathcal{K}}$, $\Lambda_{\mathcal{I}}$, $\Lambda_{\mathcal{L}}$ e $\Lambda_{\mathcal{A}}$ está ilustrada na figura 2.1.1.

Note-se ainda que os três subsistemas descritos são respectivamente fechados para a redução-β, e que os resultados enunciados para o λ-calculus puro, nomeadamente lema 2.1.5, teorema 2.1.7, corolário 2.1.8 e teorema 2.1.10, continuam a ser válidos em cada um deles.

2.2 Sistemas de lógica combinatória

Os primeiros sistemas de lógica combinatória surgiram no início deste século com os artigos (Schönfinkel, 1924) e (Curry, 1929). Analogamente aos sistemas de λ-calculus modelam funções e o seu comportamento aplicativo, sem no entanto necessitar da operação de abstracção, que é definida a partir de constantes primitivas, os combinadores.

Em seguida apresentamos o sistema de lógica combinatória $\mathcal{C}\ell$ e estabelecemos a sua correspondência com o λ-calculus puro.

Definição 2.2.1 *Considere-se o conjunto numerável de variáveis $\mathcal{V}$ e o conjunto de constantes primitivas $\{\mathsf{S}, \mathsf{K}\}$. O conjunto CL de $\mathcal{C}\ell$-termos define-se indutivamente por:*

- $\mathcal{V} \subseteq \mathsf{CL}$;
- $\mathsf{S}, \mathsf{K} \in \mathsf{CL}$;
- $X, Y \in \mathsf{CL} \quad \Rightarrow \quad XY \in \mathsf{CL}$.

O conjunto $FV(X)$ de variáveis livres de um termo $X \in \mathsf{CL}$ é o conjunto de todas as variáveis que ocorrem em X. Se $FV(X) = \emptyset$, então dizemos também que X é um termo fechado.

Definição 2.2.2 *Um $\mathcal{C}\ell$-termo da forma*

$$\mathsf{S}XYZ \qquad ou \qquad \mathsf{K}XY$$

chama-se um redex em $\mathcal{C}\ell$ com contractum

$$XZ(YZ) \qquad ou \qquad X$$

respectivamente. Equivalentemente diz-se também que aos combinadores S *e* K *estão associadas as regras de redução*

$$\begin{array}{rll} & \mathsf{S}xyz & \geq_{1\omega} \ xz(yz) \\ e & \mathsf{K}xy & \geq_{1\omega} \ x. \end{array}$$

Um $\mathcal{C}\ell$-termo X reduz-se num passo de redução fraca para $Y \in \mathsf{CL}$, o que se denota por $X \geq_{1\omega} Y$, se Y foi obtido a partir de X por substituição de um redex em X pelo seu contractum. A relação $\geq_\omega$ é o fecho reflexivo e transitivo de $\geq_{1\omega}$, enquanto que $=_\omega$ designa o fecho reflexivo, simétrico e transitivo de $\geq_{1\omega}$.

Um $\mathcal{C}\ell$-termo M que não contém nenhum redex como subtermo está em forma normal (fraca). Se existir $N \in \mathsf{CL}$ em forma normal, tal que $M \geq_\omega N$, então diz-se que M admite forma normal.

A partir de aplicações definidas entre os conjuntos Λ e CL, estabelece-se a correspondência entre o λ-calculus puro e o sistema $\mathcal{C}\ell$.

Definição 2.2.3 *A cada $\mathcal{C}\ell$-termo X associa-se o λ-termo $(X)_\lambda \in \Lambda$ definido por indução estrutural em X:*

- $x_\lambda = x$;
- $\mathsf{S}_\lambda = \lambda xyz.xz(yz)$;
- $\mathsf{K}_\lambda = \lambda xy.x$;
- $(XY)_\lambda = X_\lambda Y_\lambda$.

Lema 2.2.4 *Sejam* $X, Y \in \mathsf{CL}$, *então:*

1. $X \geq_\omega Y \quad \Rightarrow \quad X_\lambda \twoheadrightarrow_\beta Y_\lambda$;

2. $X =_\omega Y \quad \Rightarrow \quad X_\lambda =_\beta Y_\lambda$.

Como já referimos anteriormente, define-se a abstracção em $\mathcal{C}\ell$ à custa das constantes primitivas S e K:

Definição 2.2.5 (Abstracção em $\mathcal{C}\ell$)

Sejam $X \in \mathsf{CL}$ *e* $x \in \mathcal{V}$. *O termo* $\lambda^* x.X$ *define-se por:*

- $\lambda^* x.U \quad = \quad \mathsf{K}U \qquad$ *se* $x \notin FV(U)$
- $\lambda^* x.x \quad = \quad \mathsf{SKK}$
- $\lambda^* x.Ux \quad = \quad U \qquad$ *se* $x \notin FV(U)$
- $\lambda^* x.UV \quad = \quad \mathsf{S}(\lambda^* x.U)(\lambda^* x.V)$

Na definição anterior subentende-se, que uma equação só é utilizada no cálculo de $\lambda^* x.X$, se nenhuma das equações precedentes puder ser aplicada.

Lema 2.2.6 *Sejam* $X, Y \in \mathsf{CL}$ *e* $x \in \mathcal{V}$. *Então,*

$$(\lambda^* x.X)Y \geq_\omega X[Y/x].$$

Definição 2.2.7 *A cada* λ*-termo* M *associa-se o* $\mathcal{C}\ell$*-termo* $(M)_H \in \mathsf{CL}$ *definido por indução estrutural em* M.

- $x_H = x$;
- $(MN)_H = M_H N_H$;
- $(\lambda x.M)_H = \lambda^* x.M_H$.

Lema 2.2.8 *Seja* $X \in \mathsf{CL}$ *e* $M \in \Lambda$, *então*

$$(X_\lambda)_H = X \qquad e \qquad (M_H)_\lambda =_{\beta\eta} M.$$

O $\lambda\mathcal{I}$-calculus, o λ-calculus linear e o λ-calculus afim correspondem de forma análoga a sistemas de lógica combinatória, ver (Curry & Feys, 1958), dos quais[1] apresentamos três que designamos respectivamente por $\mathcal{C}\ell_{\mathcal{I}}$, $\mathcal{C}\ell_{\mathcal{L}}$ e $\mathcal{C}\ell_{\mathcal{A}}$. Note-se que estes diferem de $\mathcal{C}\ell$ essencialmente no conjunto dos seus termos e que as definições restantes se mantêm idênticas.

O conjunto $\mathsf{CL}_{\mathcal{I}}$ de $\mathcal{C}\ell_{\mathcal{I}}$-termos define-se de maneira análoga ao conjunto CL, tomando como conjunto de constantes primitivas $\{\mathsf{S},\mathsf{B},\mathsf{C},\mathsf{I}\}$, onde aos combinadores B, C e I estão respectivamente associadas as regras de redução seguintes:

$$\begin{array}{llll} & \mathsf{B}xyz & \geq_{1\omega} & x(yz) \\ & \mathsf{C}xyz & \geq_{1\omega} & xzy \\ \text{e} & \mathsf{I}x & \geq_{1\omega} & x. \end{array}$$

Para $\mathsf{CL}_{\mathcal{L}}$ e $\mathsf{CL}_{\mathcal{A}}$ tomam-se os conjuntos de constantes $\{\mathsf{B},\mathsf{C},\mathsf{I}\}$ e $\{\mathsf{B},\mathsf{C},\mathsf{K}\}$ respectivamente e restringe-se ainda a formação de termos aplicativos na definição 2.2.1 por

- $X, Y \in \mathsf{CL}_{\{\mathcal{L};\mathcal{A}\}},\ FV(X) \cap FV(Y) = \emptyset \quad \Rightarrow \quad XY \in \mathsf{CL}_{\{\mathcal{L};\mathcal{A}\}}.$

2.3 Completude combinatória

No λ-calculus e na lógica combinatória, utiliza-se a designação *combinador* como sinónimo de termo fechado. Assim o termo SKK é um combinador em $\mathcal{C}\ell$, bem como as constantes primitivas S e K, aos quais nos referimos também como combinadores primitivos, para explicitar que não são compostos por outros combinadores.

Definição 2.3.1 *Seja Σ um conjunto de λ-termos. O conjunto das combinações, ou termos aplicativos, sobre Σ designa-se por Σ^+ e define-se como o menor conjunto satisfazendo as regras seguintes:*

- $\Sigma \subseteq \Sigma^+$;
- $X, Y \in \Sigma^+ \quad \Rightarrow \quad XY \in \Sigma^+$.

[1]Para cada caso existe, obviamente, uma infinidade de sistemas possíveis, dependendo do conjunto de constantes primitivas escolhido.

Seja $Y \in \Lambda$ e suponhamos que existe $X \in \Sigma^+$ tal que $Y =_\beta X$. Então dizemos também que Y é definível a partir de Σ.

O λ-calculus puro e o sistema $\mathcal{C}\ell$ gozam ambos da propriedade de *completude combinatória*, que exprime que toda a função algébrica pode ser representada por um elemento do respectivo sistema. Formalmente, essa propriedade traduz-se no λ-calculus puro pelo lema seguinte, havendo uma versão correspondente para $\mathcal{C}\ell$.

Lema 2.3.2 (Completude combinatória) *Para toda a sequência de variáveis $x_1, \ldots, x_n$ e toda a combinação X de $x_1, \ldots, x_n$ existe um termo $M \in \Lambda^0$ tal que*

$$Mx_1 \ldots x_n \twoheadrightarrow_\beta X.$$

Esta propriedade pode ainda ser definida para conjuntos de combinadores:

Definição 2.3.3 *Seja $\Sigma \subseteq \Lambda^0$ um conjunto de combinadores.*

1. *Σ diz-se uma base para um conjunto $\Gamma \subseteq \Lambda^0$, se*

   $$\forall M \in \Gamma \; \exists N \in \Sigma^+ \quad N =_\beta M,$$

   i.e. todo o termo em Γ é definível a partir de Σ.

2. *Σ é uma base para o λ-calculus puro, ou simplesmente uma base, se Σ for uma base para Λ^0.*

3. *O conjunto Σ diz-se combinatoriamente completo (para o λ-calculus puro) se e só se for uma base.*

Com esta definição obtém-se então um resultado correspondente ao lema 2.3.2 para conjuntos de combinadores, cf. (Statman, 1986):

Lema 2.3.4 *Um conjunto $\Sigma \subseteq \Lambda^0$ é combinatoriamente completo, ou uma base, para o λ-calculus puro se e só se para toda a sequência de variáveis $x_1, \ldots, x_n$ e toda a combinação X de $x_1, \ldots, x_n$ existir um termo aplicativo $M \in \Sigma^+$ tal que*

$$Mx_1 \ldots x_n \rightarrow_\beta X.$$

Nos subsistemas descritos anteriormente, i.e. no $\lambda\mathcal{I}$-calculus, no λ-calculus linear e no λ-calculus afim, definem-se estes conceitos de maneira análoga, substituindo na definição 2.3.3 o conjunto Λ^0 pelo conjunto correspondente de termos fechados. Note-se ainda, que as expressões *base* e *combinatoriamente completo* só são utilizadas sem especificação do sistema em causa, quando se trata do λ-calculus puro. Mostram-se ainda versões correspondentes ao lema 2.3.4 para cada um dos subsistemas considerados:

Lema 2.3.5 *Um conjunto $\Sigma \subseteq \Lambda^0_{\mathcal{I}}$ (resp. $\Sigma \subseteq \Lambda^0_{\mathcal{L}}$ ou $\Sigma \subseteq \Lambda^0_{\mathcal{A}}$) é combinatoriamente completo, ou uma base, para o $\lambda\mathcal{I}$-calculus (resp. λ-calculus linear ou λ-calculus afim) se e só se para toda a sequência de variáveis $x_1, \ldots, x_n$ e toda a combinação X de $x_1, \ldots, x_n$, tal que cada x_i, para $1 \leq i \leq n$, ocorre pelo menos (resp. exactamente ou no máximo) uma vez em X, existir um termo aplicativo $M \in \Sigma^+$ tal que*

$$Mx_1 \ldots x_n \rightarrow_\beta X.$$

Exemplo 2.3.6 *No λ-calculus considerem-se os combinadores*

$$\begin{array}{rcl} \mathbf{S} & = & \lambda xyz.xz(yz) \\ \mathbf{K} & = & \lambda xy.x \\ \mathbf{B} & = & \lambda xyz.x(yz) \\ \mathbf{C} & = & \lambda xyz.xzy \\ e \quad \mathbf{I} & = & \lambda x.x. \end{array}$$

Os conjuntos $\{\mathbf{S}, \mathbf{K}\} \subseteq \Lambda^0_{\mathcal{K}}$, $\{\mathbf{S}, \mathbf{B}, \mathbf{C}, \mathbf{I}\} \subseteq \Lambda^0_{\mathcal{I}}$, $\{\mathbf{B}, \mathbf{C}, \mathbf{I}\} \subseteq \Lambda^0_{\mathcal{L}}$, e $\{\mathbf{B}, \mathbf{C}, \mathbf{K}\} \subseteq \Lambda^0_{\mathcal{A}}$ formam respectivamente bases para o λ-calculus puro, o $\lambda\mathcal{I}$-calculus, o λ-calculus linear e o afim.

De facto conhece-se, para cada sistema, um grande número de bases finitas e infinitas, que por sua vez podem ser úteis para provar a completude de outros conjuntos de combinadores:

Lema 2.3.7 *Seja $\Gamma \subseteq \Lambda^0_{\mathcal{X}}$ uma base para um conjunto de λ-termos $\Lambda^0_{\mathcal{X}}$, com $\mathcal{X} \in \{\mathcal{K}, \mathcal{I}, \mathcal{L}, \mathcal{A}\}$. Um conjunto de combinadores $\Sigma \subseteq \Lambda^0_{\mathcal{X}}$ é outra base para $\Lambda^0_{\mathcal{X}}$ se e só se todo o elemento $M \in \Gamma$ for definível a partir de Σ.*

Analisar a completude de um conjunto de combinadores Σ a partir de bases conhecidas é de facto a maneira mais usual e provavelmente a mais simples. Mesmo assim, este processo pode por vezes ser bastante complicado, especialmente quando se quer provar que um combinador particular não é definível a partir do conjunto Σ para concluir da sua incompletude. Estes problemas motivaram o estudo da definibilidade de combinadores a partir de outros, ver (Curry & Feys, 1958) e (Statman, 1986), de que resultou uma lista interessante de condições necessárias sobre bases de combinadores, com êxito particular para combinadores próprios (que apresentaremos na secção seguinte). A indecidibilidade do problema de completude combinatória para conjuntos arbitrários no λ-calculus puro foi demonstrada em 1986 por Statman:

Teorema 2.3.8 (Statman, 1986) *Seja $M \in \Lambda^0$. Então não é decidível se M é definível a partir de um conjunto finito Σ de combinadores em forma normal.*

Deste resultado segue, pela existência de bases constituídas por um só combinador, como por exemplo $\{\lambda x.x\mathbf{KSK}\}$, ver (Barendregt, 1981), o corolário seguinte.

Corolário 2.3.9 *Não é decidível se um conjunto finito Σ de combinadores (em forma normal) é uma base para o λ-calculus puro.*

2.3.1 Bases de combinadores próprios

Definição 2.3.10 *Um combinador próprio é um λ-termo fechado M da forma*

$$\lambda x_1 \ldots x_n.X$$

onde X é uma combinação de $x_1, \ldots, x_n$. Se existir uma variável $x_i \in \{x_1, \ldots, x_n\}$ tal que x_i não ocorre em X, então diz-se que M tem efeito cancelador. Se alguma das variáveis $x_1, \ldots, x_n$ ocorrer mais do que uma vez em X, então M tem efeito duplicador. O combinador próprio M chama-se um selector, se $X = x_i$ para alguma variável $x_i \in \{x_1, \ldots, x_n\}$. Dizemos ainda que M tem efeito de composição, se o termo X contiver parênteses.

As bases de combinadores próprios são, devido à estrutura simples dos seus elementos, mais facilmente caracterizáveis do que as bases de combinadores em geral. Por outro lado, o seu estudo tem interesse acrescido uma vez que qualquer combinador pode ser representado por uma combinação de combinadores próprios. A seguir enumeramos algumas das condições necessárias sobre bases de combinadores próprios apresentadas por Craig em (Curry & Feys, 1958). Mais condições são dadas em (Statman, 1986).

Teorema 2.3.11 (Craig) *Seja M um combinador próprio definível por uma combinação de combinadores próprios $M_1, \ldots, M_n$.*

1. *Se nenhum dos combinadores $M_1, \ldots, M_n$ tiver efeito cancelador, então M também não tem efeito cancelador.*
2. *Se nenhum dos combinadores $M_1, \ldots, M_n$ tiver efeito duplicador, então M também não tem efeito duplicador.*
3. *Se nenhum dos combinadores $M_1, \ldots, M_n$ for um selector, então M também não é um selector.*
4. *Se nenhum dos combinadores $M_1, \ldots, M_n$ tiver efeito de composição, então M também não tem efeito de composição.*

Nos sistemas de lógica combinatória entende-se por combinador próprio os combinadores M aos quais estão associadas reduções da forma

$$Mx_1 \ldots x_n \geq_\omega X$$

tal que X é uma combinação de $x_1, \ldots, x_n$. Analogamente ao λ-calculus dizemos que M tem efeito cancelador, se alguma das variáveis $x_1, \ldots, x_n$ não ocorrer em X, etc.

Exemplo 2.3.12 *O combinador* SKK *é próprio, pois tem-se*

$$\mathsf{SKK}x \geq_{1\omega} \mathsf{K}x(\mathsf{K}x) \geq_{1\omega} x,$$

enquanto que CBK *não é um combinador próprio, uma vez que a única redução possível é*

$$\mathsf{CBK}xy \geq_{1\omega} \mathsf{B}x\mathsf{K}y \geq_{1\omega} x(\mathsf{K}y).$$

Dado um conjunto Σ de combinadores (próprios) num sistema de lógica combinatória define--se o conjunto de combinações, ou termos aplicativos, sobre Σ como no λ-calculus. Dizemos ainda que um combinador próprio X é definível a partir de Σ se existir $Y \in \Sigma^+$ tal que Y define X no sentido de lhe estar associada a mesma regra de redução. Utilizando a noção de redução em ordem normal, mostra-se facilmente o resultado seguinte.

Lema 2.3.13 *Seja* $\Gamma \subseteq \Lambda^0$ *um conjunto de combinadores próprios. Para cada termo* $N \in \Gamma$ *da forma*

$$\lambda x_1 \ldots x_{n_N}.X_N$$

seja T_N *uma constante primitiva com regra de redução*

$$\mathsf{T}_N x_1 \ldots x_{n_N} \geq_{1\omega} X_N$$

num sistema combinatório $\mathcal{C}\ell_\Sigma$ *baseado no conjunto de constantes primitivas*

$$\Sigma = \{\mathsf{T}_N : N \in \Gamma\}.$$

Seja agora M *uma combinação sobre* Γ *e* T *a mesma combinação sobre* Σ*, onde cada* $N \in \Gamma$ *foi substituído pelo combinador correspondente* T_N*. Então tem-se para toda a sequência de variáveis* $x_1, \ldots, x_n$ *e toda a combinação* X *de* $x_1, \ldots, x_n$*,*

$$Mx_1 \ldots x_n \to_\beta X \quad \Leftrightarrow \quad Tx_1 \ldots x_n \geq_\omega X.$$

Assim conclui-se que um conjunto de combinadores próprios $\Gamma \subseteq \Lambda^0_{\mathcal{X}}$, onde $\mathcal{X} \in \{\mathcal{K}, \mathcal{I}, \mathcal{L}, \mathcal{A}\}$, é uma base para $\Lambda^0_{\mathcal{X}}$ se e só se todo o combinador próprio do sistema combinatório $\mathcal{C}\ell_{\mathcal{X}}$ for definível por uma combinação sobre Σ.

Capítulo 3

Bases lineares e bases afins

Neste capítulo estudamos a decidibilidade da completude combinatória no λ-calculus linear e no λ-calculus afim. Começamos por introduzir o conceito de domínio aplicativo, o que nos permite obter representações alternativas, mais úteis para este estudo, de termos aplicativos e combinadores próprios lineares e afins. Utilizamos estas representações para caracterizar as bases finitas de combinadores próprios dos dois sistemas de λ-calculus, mostrando também que os respectivos problemas de decidibilidade são solúveis. Dos resultados obtidos seguem ainda condições necessárias, facilmente verificáveis, para bases em geral, i.e. não necessariamente constituídas por combinadores próprios. Uma vez que consideramos somente os dois sistemas referidos, entendemos até ao fim deste capítulo as palavras *termo* e *combinador* respectivamente como *termo linear ou afim* e *combinador linear ou afim*, a não ser que o especifiquemos de outra forma.

3.1 Completude de conjuntos de combinadores próprios

Seja $\Gamma \subseteq \Lambda^0_{\mathcal{X}}$, com $\mathcal{X} \in \{\mathcal{L}, \mathcal{A}\}$, um conjunto de combinadores próprios. Já vimos no capítulo anterior, lema 2.3.13, como associar a Γ um conjunto de combinadores primitivos Σ num sistema combinatório, tal que Γ é uma base de $\Lambda^0_{\mathcal{X}}$ se e só se todo o combinador próprio de $\mathcal{C}\ell_{\mathcal{X}}$ for definível a partir de Σ. Em virtude desta propriedade suponhamos a partir de agora e até ao início da última secção deste capítulo que Σ designa um conjunto de combinadores primitivos num sistema combinatório e veremos em que condições é que qualquer combinador próprio linear (resp. afim) é definível a partir de Σ.

3.1.1 Domínios aplicativos

Seja $\mathcal{S} = \{0,1\}^*$ o conjunto das palavras sobre o alfabeto $\{0,1\}$, onde $\cdot$ e ϵ representam respectivamente a operação de concatenação e a palavra vazia. Usaremos ainda a notação $\mathcal{S}^+ = \{0,1\}^+$ para o conjunto das palavras não vazias sobre $\{0,1\}$. As letras a e b irão ser usadas para representar os elementos de $\{0,1\}$; quando a é 0 tomaremos $\overline{a}$ como 1 e vice-versa. Os elementos de $\mathcal{S}$ e $\mathcal{S}^+$ serão representadas por $\alpha, \beta, \gamma, \ldots$, etc. Habitualmente iremos omitir o símbolo da operação de concatenação e escrever $\alpha\beta$ em vez de $\alpha \cdot \beta$. A seguir introduzimos uma classe de subconjuntos finitos de $\mathcal{S}$ aos quais chamamos domínios aplicativos. Note-se que existem semelhanças entre estas estruturas e os domínios de árvore (tree-domains) utilizadas para representações de árvores completamente etiquetadas (totally labelled trees), cf. (Guessarian, 1981).

Definição 3.1.1 *Um subconjunto finito $A \subseteq \mathcal{S}$ diz-se um domínio aplicativo se satisfizer as condições seguintes:*

(i) $A \neq \emptyset$;

(ii) $\alpha \in A \ \wedge \ \alpha\beta \in A \quad \Rightarrow \quad \beta = \epsilon$;

(iii) $\alpha a \beta \in A \quad \Rightarrow \quad \exists \gamma \in \mathcal{S} : \ \alpha\overline{a}\gamma \in A$.

Se A é um domínio aplicativo e $\alpha \in \mathcal{S}$ uma palavra, então $A(\alpha)$ denota o conjunto

$$A(\alpha) = \{\beta : \alpha\beta \in A\}.$$

Exemplo 3.1.2 *Os conjuntos $A = \{\epsilon\}$, $B = \{00, 010, 011, 1\}$ e $C = \{1, 01, 001, 000\}$ são domínios aplicativos, bem como $B(01) = \{0,1\}$.*

As propriedades que se seguem irão ser consecutivamente utilizadas nas provas dos resultados deste capítulo.

Lema 3.1.3 *Seja A um domínio aplicativo e $\alpha \in \mathcal{S}$. Então,*

$$A(\alpha) \neq \emptyset \quad \Rightarrow \quad A(\alpha) \text{ é um domínio aplicativo.}$$

Prova Se $\delta \in A(\alpha)$ e $\delta\beta \in A(\alpha)$, então $\alpha\delta \in A$ e $\alpha\delta\beta \in A$, donde $\beta = \epsilon$. Por outro lado, $\delta a\beta \in A(\alpha)$ implica $\alpha\delta a\beta \in A$. Logo existe $\gamma \in \mathcal{S}$ tal que $\alpha\delta\overline{a}\gamma \in A$ e consequentemente $\delta\overline{a}\gamma \in A(\alpha)$. $\bullet$

Lema 3.1.4 *Considere-se um domínio aplicativo A tal que $\alpha = a_1 \ldots a_s \in A$, onde $s \geq 1$ e $a_1, \ldots, a_s \in \{0,1\}$. Então,*

$$\beta \in A \ \wedge \ \beta \neq \alpha \quad \Rightarrow \quad \beta = a_1 \ldots a_{q-1}\overline{a}_q\gamma$$

com $q \in \{1, \ldots, s\}$ e $\gamma \in \mathcal{S}$.

Prova Suponhamos que $\beta \in A$, $\beta \neq \alpha$ e $\beta \neq a_1 \ldots a_{q-1}\overline{a}_q\gamma$ para todo $q \in \{1, \ldots, s\}$ e todo $\gamma \in \mathcal{S}$. Segue-se que $\beta = a_1 \ldots a_{q-1}$ com $q \in \{1, \ldots, s\}$, ou $\beta = a_1 \ldots a_s\gamma$ para algum $\gamma \neq \epsilon$. Ambos os casos contrariam a hipótese de A ser domínio aplicativo pela condição 3.1.1 (ii). $\bullet$

Lema 3.1.5 *Seja A um domínio aplicativo e $\alpha \in \mathcal{S}$. Então,*

$$\exists \beta \in \mathcal{S} : \ \alpha\beta \in A \quad \vee \quad \exists \beta \in A \ \exists \gamma \in \mathcal{S} : \ \alpha = \beta\gamma.$$

Prova O caso de $\alpha = \epsilon$ é trivial. Para $\alpha = a_1 \ldots a_n$, com $n \geq 1$ e $a_1, \ldots, a_n \in \{0,1\}$, suponhamos que

$$\epsilon \notin A \ \wedge \ a_1 \notin A \ \wedge \ a_1a_2 \notin A \ \wedge \ \ldots \ \wedge \ a_1 \ldots a_n \notin A.$$

De $\epsilon \notin A$ e $A \neq \emptyset$ segue a existência de $\gamma_1, \gamma_2 \in \mathcal{S}$ tal que $a_1\gamma_1 \in A$ e $\overline{a}_1\gamma_2 \in A$. Concluímos que $A(a_1) \neq \emptyset$ é um domínio aplicativo tal que

$$\epsilon \notin A(a_1) \ \wedge \ a_2 \notin A(a_1) \ \wedge \ \ldots \ \wedge \ a_2 \ldots a_n \notin A(a_1).$$

Repetindo o mesmo processo para a_2 e $A(a_1)(a_2) = A(a_1a_2)$, para a_3 e $A(a_1a_2a_3)$, etc. resulta que $A(a_1 \ldots a_n)$ é um domínio aplicativo. Logo existe $\beta \in \mathcal{S}$ tal que $\alpha\beta \in A$. •

3.1.2 Representação de termos aplicativos

Existe uma correspondência directa entre os termos aplicativos sobre $\mathcal{V} \cup \mathcal{C}$, onde $\mathcal{C}$ é um conjunto de constantes, e as funções cujos domínios são domínios aplicativos com atribuição de valores em $\mathcal{V} \cup \mathcal{C}$.

Definição 3.1.6 *Dadas ocorrências de dois termos aplicativos X e Y denota-se a posição de Y em X por $p(Y,X) \in \mathcal{S} \cup \{\otimes\}$, definida por*

- $p(Y,X) = \otimes$ *se a ocorrência de Y não é em X;*
- $p(Y,UV) = 0 \cdot p(Y,U)$ *se a ocorrência de Y é em U;*
- $p(Y,UV) = 1 \cdot p(Y,V)$ *se a ocorrência de Y é em V;*
- $p(Y,Y) = \epsilon$.

Repare-se que o conjunto das posições de todas as ocorrências dos subtermos básicos, i.e. ocorrências de variáveis e constantes, de um qualquer termo aplicativo X forma naturalmente um domínio aplicativo que designaremos geralmente por A_X. Por outro lado existe uma e uma só função

$$t_X : A_X \longrightarrow \mathcal{V} \cup \mathcal{C}$$

que satisfaz

$$p(t_X(\alpha), X) = \alpha \quad \forall \alpha \in A_X$$

e conversamente toda a função

$$t : A \longrightarrow \mathcal{V} \cup \mathcal{C}$$

determina um e um só termo aplicativo quando A é um domínio aplicativo. Em virtude desta correspondência biunívoca chamaremos ocasionalmente, e quando não houver perigo de confusão, termos às funções cujos domínios são domínios aplicativos com atribuição de valores em $\mathcal{V} \cup \mathcal{C}$.

Exemplo 3.1.7 *Ao termo* $X = x(\mathsf{B}y)z$ *(ilustrado na figura 3.1) corresponde a função*

$$t_X : \{00, 010, 011, 1\} \longrightarrow \mathcal{V} \cup \mathcal{C}$$

com $t_X(00) = x$*,* $t_X(010) = \mathsf{B}$*,* $t_X(011) = y$ *e* $t_X(1) = z$*.*

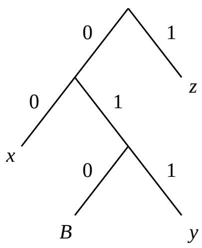

Figura 3.1: $X = x(\mathsf{B}y)z$

Podemos agora definir diversas noções para funções sobre domínios aplicativos que correspondem exactamente às mesmas noções conhecidas para os termos que elas representam, o que pode ser verificado sem qualquer dificuldade.

Definição 3.1.8 *Considerem-se três domínios aplicativos* A_X*,* A_Y *e* A_Z *juntamente com as funções* $t_X : A_X \longrightarrow \mathcal{V} \cup \mathcal{C}$*,* $t_Y : A_Y \longrightarrow \mathcal{V} \cup \mathcal{C}$ *e* $t_Z : A_Z \longrightarrow \mathcal{V} \cup \mathcal{C}$*. Diz-se que* t_Y *é um subtermo de* t_X *na posição* α *se e só se*

$$A_X(\alpha) = A_Y \quad \wedge \quad \forall \beta \in A_Y \quad t_X(\alpha\beta) = t_Y(\beta).$$

A substituição de t_Y *em* t_X *por* t_Z *é então definida por* $t : A \longrightarrow \mathcal{V} \cup \mathcal{C}$*, onde*

$$A = (A_X \setminus A_X(\alpha)) \cup \{\alpha\beta : \beta \in A_Z\}$$

e

$$t(\gamma) = \begin{cases} t_X(\gamma) & \text{para } \gamma \in A_X \setminus A_X(\alpha) \\ t_Z(\gamma') & \text{para } \gamma = \alpha\gamma' \in \{\alpha\beta : \beta \in A_Z\} \ . \end{cases}$$

3.1.3 Redução com combinadores próprios

Em seguida veremos quais os efeitos de um passo de redução, $X \geq_{1\omega} Y$, de um termo X a outro termo Y nas respectivas representações por t_X e t_Y.

Para isso consideremos um combinador próprio M com regra de redução

$$Mx_0 \ldots x_n \geq_{1\omega} Z \tag{3.1}$$

e suponhamos que Y foi obtido de X por aplicação de (3.1).

Por outras palavras, Y resulta de X por substituição de um subtermo da forma

$$MX_0 \ldots X_n$$

por

$$Z' = Z[X_0/x_0, \ldots, X_n/x_n].$$

No que diz respeito a t_X e t_Y concluímos que existe $\alpha \in \mathcal{S}$ tal que

$$t_{MX_0 \ldots X_n} \quad \text{é subtermo de } t_X \text{ na posição} \quad \alpha$$

e que

$$t_{Z'} \quad \text{é subtermo de } t_Y \text{ na posição} \quad \alpha.$$

Segue-se que

$$\left[\begin{array}{lllll} t_{X_n} & \text{é subtermo de} & t_X & \text{na posição} & \alpha 1 \\ t_{X_{n-1}} & \text{é subtermo de} & t_X & \text{na posição} & \alpha 01 \\ & & \vdots & & \\ t_{X_0} & \text{é subtermo de} & t_X & \text{na posição} & \alpha \underbrace{0 \ldots 0}_{n} 1 \\ t_M & \text{é subtermo de} & t_X & \text{na posição} & \alpha \underbrace{0 \ldots 0}_{n+1} \end{array}\right.$$

e tomando

$$\begin{array}{lllll} \alpha_0 & = & p(x_n, Z) & = & p(X_n, Z') \\ \alpha_1 & = & p(x_{n-1}, Z) & = & p(X_{n-1}, Z') \\ & & \vdots & & \\ \alpha_n & = & p(x_0, Z) & = & p(X_0, Z') \end{array} \qquad (3.1.3)^1$$

concluímos que

$$\left[\begin{array}{llll} t_{X_n} & \text{é subtermo de} & t_Y & \text{na posição} \quad \alpha\alpha_0 \\ t_{X_{n-1}} & \text{é subtermo de} & t_Y & \text{na posição} \quad \alpha\alpha_1 \\ & & \vdots & \\ t_{X_0} & \text{é subtermo de} & t_Y & \text{na posição} \quad \alpha\alpha_n \end{array} \right.$$

sempre que $\alpha_i \neq \otimes$.

Resulta então que $t_Y : A_Y \longrightarrow \mathcal{V} \cup \mathcal{C}$ pode ser definido a partir de t_X e de (3.1.3) por

$$A_Y = (A_X \setminus A_X(\alpha)) \cup A_Y(\alpha)$$

onde

$$A_Y(\alpha) = \{\alpha\alpha_i\beta : \; \alpha_i \neq \otimes \; \wedge \; \alpha\underbrace{0\ldots0}_{i}1\beta \in A_X \; \wedge \; 0 \leq i \leq n\}$$

e

$$t_Y(\gamma) = \begin{cases} t_X(\gamma) & \text{para } \gamma \in A_X \setminus A_X(\alpha) \\ t_X(\alpha\underbrace{0\ldots0}_{i}1\beta) & \text{para } \gamma = \alpha\alpha_i\beta \in A_Y(\alpha). \end{cases}$$

Reciprocamente, é possível expandir um termo Y com o combinador M para outro termo X, i.e. $Y \leq_{1\omega} X$ usando (3.1), se e só se

(3.2) $$\exists \alpha \in \mathcal{S} \; \forall i \in \{0, \ldots, n\} \quad \alpha_i \neq \otimes \; \Rightarrow \; A_Y(\alpha\alpha_i) \neq \emptyset.$$

Neste caso Y pode ser expandido para um termo X com M, ou equivalentemente t_Y pode ser expandido para t_X, substituindo em A_Y as palavras

$$\alpha\alpha_i\beta \quad \text{por} \quad \alpha\underbrace{0\ldots0}_{i}1\beta$$

[1]Note-se, que os combinadores considerados neste capítulo não são duplicadores donde faz sentido falar "da" posição de x_i em Z.

para $0 \leq i \leq n$ e sempre que $\alpha_i \neq \otimes$, mas mantendo o valor de t_Y para estes argumentos, i.e. tomando

$$t_X(\alpha \underbrace{0\ldots0}_{i} 1\beta) = t_Y(\alpha\alpha_i\beta).$$

Além disso junta-se a palavra $\alpha \underbrace{0\ldots0}_{n+1}$ ao domínio com

$$t_X(\alpha \underbrace{0\ldots0}_{n+1}) = M,$$

bem como, para cada $\alpha_i = \otimes$, os elementos de um conjunto

$$\{\alpha \underbrace{0\ldots0}_{i} 1\beta : \beta \in B_i\},$$

onde B_i pode ser qualquer domínio aplicativo, cujos valores podem ser arbitrariamente escolhidos em $\mathcal{V} \cup \mathcal{C}$.

Resumindo, a expansão de termos com combinadores próprios pode, de um modo grosseiro, ser visto como substituição de subsequências por sequências da forma $0\ldots01$. Isto leva-nos, juntamente com o facto do combinador M ficar completamente determinado pela sequência $\alpha_0, \ldots, \alpha_n$, a empregar a representação seguinte para combinadores próprios

$$\psi(M) = \begin{cases} \alpha_0 & \Leftarrow_1 \quad 1 \\ & \vdots \\ \alpha_n & \Leftarrow_1 \quad \underbrace{0\ldots0}_{n} 1 \end{cases}$$

com a seguinte generalização para conjuntos de combinadores próprios

$$\psi(\Sigma) = \{\psi(M) \mid M \in \Sigma\}.$$

Note-se ainda que os conjuntos

$$\{\alpha_0, \ldots, \alpha_n\} \setminus \{\otimes\}$$

e

$$\{1, 01, \ldots, \underbrace{0\ldots0}_{n} 1, \underbrace{0\ldots0}_{n+1}\}$$

são ambos domínios aplicativos qualquer que seja o combinador M.

Exemplo 3.1.9 *O conjunto* $\psi(\{\mathsf{B},\mathsf{C},\mathsf{I},\mathsf{K}\})$ *contém as regras*

$$\psi(\mathsf{B}) \;=\; \begin{cases} p(z,x(yz)) & \Leftarrow_1 \;\; p(z,\mathsf{B}xyz) \\ p(y,x(yz)) & \Leftarrow_1 \;\; p(y,\mathsf{B}xyz) \\ p(x,x(yz)) & \Leftarrow_1 \;\; p(x,\mathsf{B}xyz) \end{cases} \;=\; \begin{cases} 11 & \Leftarrow_1 \;\; 1 \\ 10 & \Leftarrow_1 \;\; 01 \\ 0 & \Leftarrow_1 \;\; 001 \end{cases}$$

$$\psi(\mathsf{C}) \;=\; \begin{cases} 01 & \Leftarrow_1 \;\; 1 \\ 1 & \Leftarrow_1 \;\; 01 \\ 00 & \Leftarrow_1 \;\; 001 \end{cases}$$

$$\psi(\mathsf{I}) = \{\epsilon \;\; \Leftarrow_1 \;\; 1 \qquad e \qquad \psi(\mathsf{K}) = \begin{cases} \otimes & \Leftarrow_1 \;\; 1 \\ \epsilon & \Leftarrow_1 \;\; 01 \end{cases}$$

A partir do conjunto $\{\mathsf{B},\mathsf{C},\mathsf{I},\mathsf{K}\}$ *é possível definir o combinador próprio* N *com regra de redução*

$$Nxyz \geq_{1\omega} y(zx)$$

por $\mathsf{B}(\mathsf{CB})(\mathsf{CI})$ *como verificamos pelo quadro seguinte.*

	$y(zx)$	$\leq_{1\omega}$	$y(\mathsf{I}zx)$	$\leq_{1\omega}$	$y(\mathsf{CI}xz)$	$\leq_{1\omega}$	$\mathsf{B}y(\mathsf{CI}x)z$	$\leq_{1\omega}$	$\mathsf{CB}(\mathsf{CI}x)yz$	$\leq_{1\omega}$	$\mathsf{B}(\mathsf{CB})(\mathsf{CI})xyz$
z :	10	$\Leftarrow_1$	101	$\Leftarrow_1$	11	$\Leftarrow_1$	1	$\Leftarrow_1$	1	$\Leftarrow_1$	1
y :	0	$\Leftarrow_1$	0	$\Leftarrow_1$	0	$\Leftarrow_1$	001	$\Leftarrow_1$	01	$\Leftarrow_1$	01
x :	11	$\Leftarrow_1$	11	$\Leftarrow_1$	101	$\Leftarrow_1$	011	$\Leftarrow_1$	0011	$\Leftarrow_1$	001
I :			100	$\Leftarrow_1$	1001	$\Leftarrow_1$	0101	$\Leftarrow_1$	00101	$\Leftarrow_1$	00011
C :					1000	$\Leftarrow_1$	0100	$\Leftarrow_1$	00100	$\Leftarrow_1$	00010
B :							000	$\Leftarrow_1$	0001	$\Leftarrow_1$	000011
C :									0000	$\Leftarrow_1$	000010
B :											00000

Repare-se, por exemplo, que a última coluna deste quadro foi obtida da coluna anterior por expansão com $\psi(\mathsf{B})$ *na posição* 00. *De facto substituíram-se todas as palavras da forma*

$$00 \cdot 11 \cdot \beta \quad por \quad 00 \cdot 1 \cdot \beta$$

as palavras da forma

$$00 \cdot 10 \cdot \beta \quad por \quad 00 \cdot 01 \cdot \beta$$

e as da forma

$$00 \cdot 0 \cdot \beta \quad por \quad 00 \cdot 001 \cdot \beta$$

juntando ainda a palavra $00 \cdot 000$ *com valor* B *ao domínio.*

3.1.4 Independência das regras

No que se segue, iremos estabelecer uma condição necessária e suficiente sobre o conjunto Σ, para que todo o combinador próprio linear (resp. afim) M com regra de redução

$$Mx_1 \ldots x_n \geq_{1\omega} X$$

seja definível a partir de Σ. É claro que, para isso, terá que existir uma sequência da forma

$$X \leq_{1\omega} X_1 \leq_{1\omega} \ldots \leq_{1\omega} Zx_1 \ldots x_n$$

onde cada passo de expansão é feito com um combinador em Σ e onde Z é uma combinação constituída somente por elementos de Σ.

Esta condição equivale ainda à existência, para todo o termo aplicativo X e toda a variável x, de uma sequência da forma

$$X \leq_{1\omega} X_1 \leq_{1\omega} \ldots \leq_{1\omega} Z_0 x$$

com $x \notin FV(Z_0)$, obtida por aplicação dos combinadores em Σ. No caso de se tratar do λ-calculus linear restringe-se esta condição aos pares (X, x) em que x tem exactamente uma ocorrência em X, enquanto que para o λ-calculus afim x deverá ocorrer no máximo uma vez em X.

No que diz respeito a t_X e $\psi(\Sigma)$ isto traduz-se na possibilidade de expansão de t_X com regras $\psi(M) \in \psi(\Sigma)$ a

$$t_Z : A_Z \longrightarrow \mathcal{V} \cup \mathcal{C}$$

tal que

$$t_Z^{-1}(x) = \{1\},$$

i.e. Z é da forma $Z_0 x$ e $x \notin FV(Z_0)$.

Em particular, deverá ser possível re-escrever a posição de x em X para 1 com as linhas das regras de $\psi(\Sigma)$. Para $\alpha = p(x, X)$ denotamos esta noção por

$$\Sigma : \alpha \Leftarrow 1$$

que passamos a definir formalmente.

Definição 3.1.10 *Considere-se um conjunto de combinadores próprios Σ. Uma palavra $\alpha \in \mathcal{S} \cup \{\otimes\}$ re-escreve-se a partir de Σ num passo para $\beta \in \mathcal{S}$, o que denotamos por $\Sigma : \alpha \Leftarrow_1 \beta$, se existir $M \in \Sigma$ tal que,*

se $\alpha \neq \otimes$, então

$$\exists \alpha_1, \alpha_2, \gamma, \delta \in \mathcal{S} : \ \alpha = \alpha_1 \gamma \alpha_2 \ \wedge \ \beta = \alpha_1 \delta \alpha_2 \ \wedge \ \gamma \Leftarrow_1 \delta \in \psi(M)$$

e para $\alpha = \otimes$

$$\otimes \Leftarrow_1 \beta \in \psi(M).$$

Escrevemos ainda

$$\Sigma : \alpha \Leftarrow \beta$$

se existirem $n \geq 0$ e $\alpha_0, \ldots, \alpha_n \in \mathcal{S} \cup \{\otimes\}$ tal que $\alpha = \alpha_0$, $\beta = \alpha_n$ e $\Sigma : \alpha_i \Leftarrow_1 \alpha_{i+1}$ para $i \in \{0, \ldots, n-1\}$.

Ocasionalmente, e quando não houver dúvidas em relação ao conjunto Σ, escrevemos também $\alpha \Leftarrow \beta$ em vez de $\Sigma : \alpha \Leftarrow \beta$. Com esta notação as considerações anteriores tomam a forma seguinte.

Lema 3.1.11 *Considere-se um conjunto de combinadores próprios lineares (resp. afins) Σ, tal que todo o combinador próprio linear (resp. afim) é definível a partir de Σ. Então $\Sigma : \alpha \Leftarrow 1$ para todo $\alpha \in \mathcal{S}$ (resp. $\alpha \in \mathcal{S} \cup \{\otimes\}$).* •

Em seguida mostramos que o resultado converso também é válido, obtendo assim uma caracterização mais simples das bases de combinadores próprios para o λ-calculus linear e para o λ-calculus afim. O resultado deve-se essencialmente à presença de selectores em qualquer base de combinadores próprios, cf. teorema 2.3.11, e resulta então directamente da propriedade seguinte.

Lema 3.1.12 *Considere-se um selector T, um combinador próprio (linear ou afim) M com*

$$\psi(M) = \begin{cases} \alpha_0 & \Leftarrow_1 \; 1 \\ & \vdots \\ \alpha_n & \Leftarrow_1 \; \underbrace{0\ldots0}_{n}1 \end{cases}$$

um termo aplicativo X e $\alpha = p(cv, X)$ para algum $cv \in \mathcal{V} \cup \mathcal{C}$. Se $\alpha = \beta\alpha_i\gamma$ onde $\alpha_i \in \{\alpha_0, \ldots, \alpha_n\}$ (ou $\alpha = \otimes$ e $\otimes \in \{\alpha_1, \ldots, \alpha_n\}$), então é possível expandir o termo $t_X : A_X \longrightarrow \mathcal{V} \cup \mathcal{C}$ um número finito de vezes com $\psi(T)$ para $t_{\tilde{X}} : A_{\tilde{X}} \longrightarrow \mathcal{V} \cup \mathcal{C}$, tal que $p(cv, \tilde{X}) = \alpha$ e $t_{\tilde{X}}$ é expansível com M na posição β (resp. ϵ).

Prova Mostramos o resultado para $\alpha = \beta\alpha_i\gamma$ com $\alpha_i \in \{\alpha_0, \ldots, \alpha_n\}$ uma vez que a prova é análoga para $\alpha = \otimes$ e $\otimes \in \{\alpha_0, \ldots, \alpha_n\}$. Para isso construiremos uma expansão de t_X para $t_{\tilde{X}}$ com T tal que $p(cv, \tilde{X}) = \alpha$ e

$$\forall j \in \{0, \ldots, n\} \;\; \alpha_j \neq \otimes \;\; \Rightarrow \;\; A_{\tilde{X}}(\beta\alpha_j) \neq \emptyset$$

o que, como vimos em (3.2), garante que $t_{\tilde{X}}$ é expansível com M na posição β.

O caso de $n = 0$ é trivial, pelo que consideramos $n \geq 1$.

Começamos por notar que para todo o selector T se tem

$$\psi(T) = \begin{cases} \otimes & \Leftarrow_1 \; 1 \\ & \vdots \\ \epsilon & \Leftarrow_1 \; \underbrace{0\ldots0}_{d\geq 0}1 \\ & \vdots \\ \otimes & \Leftarrow_1 \; \underbrace{0\ldots0}_{N}1 \end{cases}$$

e que T' é um selector, sempre que $\epsilon \Leftarrow_1 \underbrace{0\ldots0}_{d'\geq 0}1 \;\in\; \psi(T')$.

Concluímos deste modo que o resultado é válido para $\alpha_i = \epsilon$, i.e. para M selector, uma vez que se tem então $\alpha_j = \otimes$ para $j \neq i$.

Finalmente suponhamos

$$\alpha_i = a_1 \ldots a_s$$

com $s \geq 1$ e $a_1, \ldots, a_s \in \{0,1\}$ e consideremos $\alpha_j \in \{\alpha_0, \ldots, \alpha_n\}$ tal que $\alpha_j \neq \otimes$, $i \neq j$ e $A_X(\beta\alpha_j) = \emptyset$. Já dissemos antes que o conjunto $\{\alpha_0, \ldots, \alpha_n\} \setminus \{\otimes\}$ é um domínio aplicativo para todo o combinador próprio M. Assim concluímos do lema 3.1.4 que

$$\alpha_j = a_1 \ldots a_{q-1}\overline{a}_q b_1 \ldots b_t$$

onde $q \in \{1, \ldots, s\}$ e $t \geq 0$. Decorre de $A_X(\beta\alpha_j) = \emptyset$ que $\beta a_1 \ldots a_{q-1}\overline{a}_q b_1 \ldots b_t \gamma \notin A_X$ qualquer que seja $\gamma \in \mathcal{S}$. Donde, pelo lema 3.1.5

$$\beta a_1 \ldots a_{q-1}\overline{a}_q b_1 \ldots b_{t'} \in A_X$$

com $0 \leq t' < t$. Assim, expandindo t_X na posição $\beta a_1 \ldots a_{q-1}\overline{a}_q b_1 \ldots b_{t'}$ com $\psi(T)$ para $t_{X'}$ obtemos

$$\begin{array}{lcl}
\beta a_1 \ldots a_{q-1}\overline{a}_q b_1 \ldots b_{t'}1 & \in & A_{X'} \\
\beta a_1 \ldots a_{q-1}\overline{a}_q b_1 \ldots b_{t'}01 & \in & A_{X'} \\
 & \vdots & \\
\beta a_1 \ldots a_{q-1}\overline{a}_q b_1 \ldots b_{t'}\underbrace{0\ldots0}_{N}1 & \in & A_{X'} \\
\beta a_1 \ldots a_{q-1}\overline{a}_q b_1 \ldots b_{t'}\underbrace{0\ldots0}_{N+1} & \in & A_{X'}
\end{array}$$

enquanto que $p(cv, X') = \alpha$. Se nenhum destes elementos novos for da forma $\beta\alpha_j\gamma_j$ com $\gamma_j \in \mathcal{S}$, então é claro que um deles é prefixo de $\beta\alpha_j$ e podemos expandir $t_{X'}$ nesta posição com $\psi(T)$ até obter t_{X_j} tal que $\beta\alpha_j\gamma_j \in A_{X_j}$ com $\gamma_j \in \mathcal{S}$. Note-se que este método pode ser aplicado às palavras $\alpha_j \neq \alpha_i$ em qualquer ordem, sem nunca interferir com palavras introduzidas anteriormente e em particular sem nunca alterar a posição de cv. •

Corolário 3.1.13 *Um conjunto de combinadores próprios lineares (resp. afins) Σ define todo o combinador próprio linear (resp. afim) se e só se $\Sigma : \alpha \Leftarrow 1$ para todo $\alpha \in \mathcal{S}$ (resp. $\alpha \in \mathcal{S} \cup \{\otimes\}$).* •

3.2 Bases de combinadores próprios no λ-calculus linear

Nesta secção pretendemos caracterizar as bases de combinadores próprios do λ-calculus linear através dos conjuntos de regras $\psi(\Sigma)$. Começamos por enumerar uma lista de condições necessárias sobre $\psi(\Sigma)$.

Lema 3.2.1 *Seja Σ um conjunto de combinadores próprios lineares. Se todo o combinador próprio em $\mathcal{C}\ell_{\mathcal{L}}$ for definível a partir de Σ, então*

$$\exists \alpha \in \mathcal{S}^{+} : \ \alpha 1 \Leftarrow_1 1 \ \in \ \psi(\Sigma).$$

Prova Decorre do corolário 3.1.13 que

$$\Sigma : 01 \Leftarrow 1.$$

Por outras palavras, existe uma sequência finita de palavras

$$01 = \beta_0 \Leftarrow_1 \ldots \Leftarrow_1 \beta_n = 1$$

tal que cada β_i foi obtido de β_{i-1}, $1 \leq i \leq n$, por substituição de uma subsequência γ por uma sequência da forma $\underbrace{0\ldots0}_{d\geq 0}1$, onde $\gamma \Leftarrow_1 \underbrace{0\ldots0}_{d}1 \ \in \psi(\Sigma)$. É claro que cada β_i, $0 \leq i \leq n$, é da forma $\beta_i = \alpha_i 1$ com $\alpha_i \in \mathcal{S}$. Assim, e como $\beta_0 = 0 \cdot 1$ e $\beta_n = \epsilon \cdot 1$, concluímos que pelo menos uma das regras aplicadas é da forma $\alpha 1 \Leftarrow_1 1$ com $\alpha \in \mathcal{S}^{+}$. $\bullet$

Definição 3.2.2 *Para $\alpha \in \mathcal{S}$ sejam $\#_0(\alpha)$ e $\#_1(\alpha)$ respectivamente o número de 0's e o número de 1's em α.*

Lema 3.2.3 *Seja Σ um conjunto de combinadores próprios lineares. Se todo o combinador próprio em $\mathcal{C}\ell_{\mathcal{L}}$ for definível a partir de Σ, então*

$$\exists \gamma \in \mathcal{S}^{+} : \quad \#_0(\gamma) > 0 \quad \wedge \quad \gamma \Leftarrow_1 1 \ \in \ \psi(\Sigma).$$

Prova Analogamente à prova do lema anterior, este resultado decorre de $\Sigma : 01 \Leftarrow 1$, já que $\#_0(01) = 1$ e $\#_0(1) = 0$. •

Definição 3.2.4 *Considere-se um conjunto finito Σ de combinadores próprios lineares. Para cada regra $\alpha_i \Leftarrow_1 \underbrace{0\ldots0}_{d_i}1$ em $\psi(\Sigma)$, $1 \leq i \leq r$, seja $p_i = \#_0(\alpha_i) - d_i$. Então definimos a equação diofantina (nos naturais) de Σ por*

$$\sum_1^r p_i x_i = 1.$$

Exemplo 3.2.5 *A equação diofantina de $\Sigma = \{\mathsf{I}, \mathsf{C}, \mathsf{B}\}$ é*

$$x - y - z = 1$$

e tem solução $x = 1$ e $y = z = 0$. Os coeficientes de x e y resultam respectivamente da primeira e da segunda regra em $\psi(\mathsf{C})$ e o coeficiente de z provém da terceira regra em $\psi(\mathsf{B})$.

Lema 3.2.6 *Seja Σ um conjunto finito de combinadores próprios lineares tal que todo o combinador próprio em $\mathcal{C}\ell_{\mathcal{L}}$ é definível a partir de Σ. Então a equação diofantina de Σ*

$$\sum_1^r p_i x_i = 1$$

admite solução $(x_1, \ldots, x_r) \in \mathbb{N}^r$.

Prova Cada coeficiente p_i está directamente associado a uma regra $\alpha_i \Leftarrow_1 \underbrace{0\ldots0}_{d_i}1 \in \psi(\Sigma)$ e representa o número de 0's consumidos numa aplicação desta regra. Para cada $i \in \{1, \ldots, r\}$ seja x_i o número de vezes que a regra correspondente foi aplicada em

$$01 \Leftarrow_1 \ldots \Leftarrow_1 1.$$

Então segue-se de $\#_0(01) - \#_0(1) = 1$ que $(x_1, \ldots, x_r) \in \mathbb{N}^r$ é solução de $\sum_1^r p_i x_i = 1$. •

O resultado seguinte mostra que as condições necessárias anteriores, juntamente com duas condições estabelecidas por Craig (ver teorema 2.3.11), são suficientes para um conjunto finito de combinadores próprios lineares formar uma base para o λ-calculus linear.

Teorema 3.2.7 *Um conjunto finito de combinadores próprios lineares é uma base para o λ-calculus linear se e só se o conjunto associado Σ de combinadores primitivos satisfizer as condições seguintes.*

1. $\mathsf{I} \in \Sigma$;
2. Σ *contém um combinador com efeito de composição;*
3. $\exists \alpha \in \mathcal{S}^+ : \ \alpha 1 \Leftarrow_1 1 \ \in \ \psi(\Sigma)$;
4. $\exists \gamma \in \mathcal{S}^+ : \ \#_0(\gamma) > 0 \ \wedge \ \gamma \Leftarrow_1 1 \ \in \ \psi(\Sigma)$;
5. *a equação diofantina de* Σ

   (3.3) $$\sum_1^r p_i x_i = 1$$

   é solúvel.

Prova Tendo em atenção o teorema 2.3.11 e os lemas 3.2.1, 3.2.3 e 3.2.6, basta mostrar que as condições 1.-5. garantem a completude de Σ, o que, pelo corolário 3.1.13, equivale a mostrar

$$\forall \alpha \in \mathcal{S} \quad \Sigma : \alpha \Leftarrow 1$$

ou ainda

(3.4) $$\forall \alpha \in \mathcal{S}^+ \quad \Sigma : \alpha \Leftarrow 1$$

uma vez que $\epsilon \Leftarrow_1 1 \ \in \psi(\Sigma)$ decorre de $\mathsf{I} \in \Sigma$.

Começamos por provar que

(3.5) $$\exists k_0 \geq 0 : \ k \geq k_0 \quad \Rightarrow \quad \Sigma : \underbrace{0 \ldots 0}_{k} 0 \ \Leftarrow \ \underbrace{0 \ldots 0}_{k} \underbrace{1 \ldots 1}_{\sum_1^r x_i}$$

onde $(x_1, \ldots, x_r)$ é uma solução da equação diofantina (3.3).

Esta propriedade deriva do facto de todo $p_i \in \{p_1, \ldots, p_r\}$ corresponder a uma regra $\gamma_i \Leftarrow_1 \underbrace{0 \ldots 0}_{d_i} 1$ com $\#_0(\gamma_i) = d_i + p_i$. Assim, escolhendo k suficientemente grande e aplicando sempre que necessário a regra $\epsilon \Leftarrow_1 1$, concluímos que

$$\underbrace{0 \ldots 0}_{k} 0 \Leftarrow_1 \ldots \Leftarrow_1 \underbrace{0 \ldots 0}_{k-(d_i+p_i)} 0 \gamma_i \Leftarrow_1 \underbrace{0 \ldots 0}_{k-p_i} 01.$$

Repetindo esse processo x_i vezes obtemos

$$\underbrace{0\ldots0}_{k}0 \Leftarrow_1 \ldots \Leftarrow_1 \underbrace{0\ldots0}_{k-p_ix_i}0\underbrace{1\ldots1}_{x_i}$$

e continuando para $p_j \neq p_i$, $j \in \{1,\ldots,r\}$, tem-se

$$\underbrace{0\ldots0}_{k}0 \Leftarrow_1 \ldots \Leftarrow_1 \underbrace{0\ldots0}_{k-\sum_1^r p_ix_i}\ 0\underbrace{1\ldots1}_{\sum_1^r x_i} = \underbrace{0\ldots0}_{k}\underbrace{1\ldots1}_{\sum_1^r x_i}$$

uma vez que $\sum_1^r p_ix_i = 1$.

Passamos então a provar (3.4), distinguindo dois casos. Primeiro suponhamos que existe $n \geq 2$ tal que $\underbrace{1\ldots1}_{n} \Leftarrow_1 1 \in \psi(\Sigma)$. De $\epsilon \Leftarrow_1 1 \in \psi(\Sigma)$ segue

$$\forall y \geq 0 \quad \underbrace{1\ldots1}_{y} \Leftarrow 1.$$

Concluímos a prova para este caso, mostrando que existe $n_0 \geq 1$ tal que $0 \Leftarrow \underbrace{1\ldots1}_{n_0}$.

Note-se mais uma vez que para todo o combinador M o conjunto das palavras α com $\alpha \Leftarrow_1 0\ldots01 \ \in \psi(M)$ forma um domínio aplicativo. Como $\underbrace{1\ldots1}_{n\geq 2} \Leftarrow_1 1 \in \psi(\Sigma)$, existem

$$\begin{aligned}
\alpha_1 &\in \{0, 01, 011, \ldots\}\\
\alpha_2 &\in \{10, 101, 1011, \ldots\}\\
&\ \vdots\\
\alpha_n &\in \{\underbrace{1\ldots1}_{n-1}0, \underbrace{1\ldots1}_{n-1}01, \underbrace{1\ldots1}_{n-1}011, \ldots\}
\end{aligned}$$

tal que para $1 \leq i \leq n$,

$$\alpha_i \Leftarrow_1 \underbrace{0\ldots0}_{d_i\geq 1}1 \ \in \psi(\Sigma)$$

onde $d_i \neq d_j$ para $i \neq j$. De $n \geq 2$ decorre que $\psi(\Sigma)$ contém uma regra

$$\alpha \Leftarrow_1 \underbrace{0\ldots0}_{d}1$$

com $\alpha \in \{\alpha_1,\ldots,\alpha_n\}$, $\#_0(\alpha) = 1$ e $d \geq 2$. Assim obtém-se

$$0 \Leftarrow \alpha \Leftarrow \underbrace{0\ldots0}_{d-1}01 \Leftarrow \underbrace{0\ldots0}_{d-1}\alpha 1 \Leftarrow \underbrace{0\ldots0}_{d-1}\underbrace{0\ldots0}_{d-1}011 \Leftarrow \ldots \Leftarrow \underbrace{0\ldots0}_{k}01\ldots1$$

onde k é o menor múltiplo comum de $(d-1)$ e $p = \#_0(\gamma)$ maior ou igual a k_0 em (3.5), para γ da condição 4. Donde

$$\underbrace{0\ldots0}_{k}01\ldots1 \Leftarrow \underbrace{0\ldots0}_{k}1\ldots1 \Leftarrow \underbrace{0\ldots0}_{k-p}1\ldots1 \Leftarrow \ldots \Leftarrow 1\ldots1$$

o que conclui esta parte da prova.

Suponhamos agora que não existe nenhuma regra da forma $\underbrace{1\ldots1}_{n} \Leftarrow_1 1$ com $n \geq 2$ em $\psi(\Sigma)$. Daqui resulta $\#_0(\alpha) = s \geq 1$ para α da condição 3. Logo existem $\tilde{n}, d \geq 1$ tal que

$$\underbrace{1\ldots1}_{\tilde{n}} \Leftarrow_1 \underbrace{0\ldots0}_{d}1 \ \in \psi(\Sigma) \tag{3.6}$$

donde

$$0 \Leftarrow 0\underbrace{1\ldots1}_{\tilde{n}} \Leftarrow \underbrace{0\ldots0}_{d}01 \Leftarrow \underbrace{0\ldots0}_{k}01\ldots1 \Leftarrow \underbrace{0\ldots0}_{k}1\ldots1 \Leftarrow 1\ldots1$$

onde k é o menor múltiplo comum de d e s maior ou igual a k_0 em (3.5).

Para completar a prova resta-nos mostrar que

$$\forall y \geq 1 \quad \underbrace{1\ldots1}_{y} \Leftarrow 1.$$

Se existir uma regra como em (3.6) tal que $\tilde{n} \geq 2$, então

$$\underbrace{1\ldots1}_{y} \quad \Leftarrow \quad \underbrace{1\ldots1}_{m(\tilde{n}-1)s}1 \quad \Leftarrow \quad \underbrace{0\ldots0}_{mds}1 \quad \Leftarrow \quad \underbrace{0\ldots0}_{(md-1)s}\alpha 1 \quad \Leftarrow \quad \underbrace{0\ldots0}_{(md-1)s}1 \quad \Leftarrow \quad 1$$

onde $m(\tilde{n}-1)s$ é o menor múltiplo de $(\tilde{n}-1)s$ maior ou igual a $y-1$.

Finalmente suponhamos $\tilde{n} = 1$ para qualquer regra como em (3.6). Pela condição 3. tem-se

$$1 \Leftarrow_1 \underbrace{0\ldots0}_{d\geq1}1 \ \in \psi(\Sigma)$$

e da existência de um combinador com efeito de composição em Σ segue

$$\beta 11 \Leftarrow_1 \underbrace{0\ldots0}_{c\geq0}1 \ \in \psi(\Sigma)$$

pelo que $\#_0(\beta) = e \geq 1$.

Tomemos $f = e - c$. Então,

$$\begin{array}{llllll} \underbrace{1\ldots1}_{y} & \Leftarrow & \underbrace{1\ldots1}_{m_1s+1} & \Leftarrow & \underbrace{0\ldots0}_{m_2sd}\underbrace{1\ldots1}_{m_1s+1} & \Leftarrow & \underbrace{0\ldots0}_{m_2sd-e}\beta11\underbrace{1\ldots1}_{m_1s-1} \\ & \Leftarrow & \underbrace{0\ldots0}_{m_2sd-f}\underbrace{1\ldots1}_{m_1s} & \Leftarrow & \underbrace{0\ldots0}_{(m_2d-m_1f)s}\ 1 & \Leftarrow & 1 \end{array}$$

onde m_1s é o menor múltiplo de s maior ou igual a $y-1$ e onde m_2sd é o menor múltiplo de sd tal que $m_2sd \geq e$ e $m_2d \geq m_1f$. •

Algoritmos para resolver (sistemas de) equações diofantinas sobre o conjunto dos números naturais são conhecidos desde do início do século, ver por exemplo (Stanley, 1983), pelo que concluímos o seguinte:

Corolário 3.2.8 *A completude combinatória para o λ-calculus linear é decidível para conjuntos finitos de combinadores próprios.* •

Exemplo 3.2.9 *Considere-se o conjunto de combinadores próprios lineares*

$$\Gamma = \{\lambda x.x, \lambda xyzu.y(zu)x\}$$

ao qual associamos $\Sigma = \{\mathsf{I}, \mathsf{A}\}$, *tal que*

$$\begin{array}{rll} \mathsf{I}x & \geq_{1\omega} & x \\ e \qquad \mathsf{A}xyzu & \geq_{1\omega} & y(zu)x. \end{array}$$

Então $\psi(\Sigma)$ *contém as regras seguintes:*

$$\psi(\mathsf{I}) = \{\epsilon \Leftarrow_1 1 \qquad e \qquad \psi(\mathsf{A}) = \left\{\begin{array}{lll} 011 & \Leftarrow_1 & 1 \\ 010 & \Leftarrow_1 & 01 \\ 00 & \Leftarrow_1 & 001 \\ 1 & \Leftarrow_1 & 0001 \end{array}\right.$$

e a equação diofantina de Σ *é*

$$x + y - 3z = 1$$

com solução $x = 1$ *e* $y = z = 0$.

Verifica-se facilmente que Σ satisfaz as 5 condições do teorema 3.2.7 e consequentemente conclui-se que Γ forma uma base para o λ-calculus linear.

De facto, para definir o combinador linear

$$\mathsf{C}_* xy \geq_{1\omega} yx$$

em Σ basta seguir a demonstração do teorema 3.2.7 para obter

y :	0	$\Leftarrow_1$	01	$\Leftarrow_1$	011	$\Leftarrow_1$	1	$\Leftarrow_1$	1	$\Leftarrow_1$	1	$\Leftarrow_1$	1	$\Leftarrow_1$	1
x :	1	$\Leftarrow_1$	1	$\Leftarrow_1$	1	$\Leftarrow_1$	0001	$\Leftarrow_1$	00011	$\Leftarrow_1$	001	$\Leftarrow_1$	0011	$\Leftarrow_1$	01
I :			00	$\Leftarrow_1$	00	$\Leftarrow_1$	001	$\Leftarrow_1$	001	$\Leftarrow_1$	000001	$\Leftarrow_1$	000001	$\Leftarrow_1$	0001001
I :					010	$\Leftarrow_1$	01	$\Leftarrow_1$	01	$\Leftarrow_1$	01	$\Leftarrow_1$	01	$\Leftarrow_1$	00001
A :							0000	$\Leftarrow_1$	0000	$\Leftarrow_1$	00001	$\Leftarrow_1$	00001	$\Leftarrow_1$	000101
I :									00010	$\Leftarrow_1$	0001	$\Leftarrow_1$	0001	$\Leftarrow_1$	00011
A :											000000	$\Leftarrow_1$	000000	$\Leftarrow_1$	0001000
I :													0010	$\Leftarrow_1$	001
A :															00000

$$\begin{aligned} \textit{i.e.} \qquad yx &\leq_{1\omega} \mathsf{I}yx \leq_{1\omega} \mathsf{I}(\mathsf{I}y)x \leq_{1\omega} \mathsf{A}x\mathsf{II}y \leq_{1\omega} \mathsf{A}(\mathsf{I}x)\mathsf{II}y \\ &\leq_{1\omega} \mathsf{AIAI}x\mathsf{I}y \leq_{1\omega} \mathsf{AIAI}(\mathsf{I}x)\mathsf{I}y \leq_{1\omega} \mathsf{AI}(\mathsf{AIAI})\mathsf{I}xy \end{aligned}$$

e consequentemente C_* pode ser definido por $\mathsf{AI}(\mathsf{AIAI})\mathsf{I}$.

3.3 Bases de combinadores próprios no λ-calculus afim

Nesta secção obtemos um resultado similar ao teorema 3.2.7 para o λ-calculus afim. Analogamente à secção anterior começamos por estabelecer algumas propriedades dos conjuntos Σ de combinadores primitivos associados às bases de combinadores próprios deste sistema.

Lema 3.3.1 *Seja Σ um conjunto de combinadores próprios afins tal que todo o combinador próprio em $\mathcal{C}\ell_{\mathcal{A}}$ é definível a partir de Σ. Se nenhum selector $T \in \Sigma$ tiver regra de redução da forma $Tx_1 \ldots x_n \geq_{1\omega} x_n$, então existe $\alpha \in \mathcal{S}$ tal que*

$$0\alpha \Leftarrow_1 1 \ \in \ \psi(\Sigma).$$

Prova Se Σ não contém nenhum selector com regra de redução $Tx_1 \ldots x_n \geq_{1\omega} x_n$, então $\epsilon \Leftarrow_1 1 \ \notin \psi(\Sigma)$. Pelo corolário 3.1.13 tem-se

$$01 \Leftarrow_1 \ldots \Leftarrow_1 1$$

e é claro que pelo menos uma das regras aplicadas nesta sequência é da forma $0\alpha \Leftarrow_1 1$ com $\alpha \in \mathcal{S}$. $\bullet$

Verifica-se facilmente que os resultados 3.2.1, 3.2.3 e 3.2.6 são generalizáveis para bases do λ-calculus afim, mas como a propriedade estabelecida no lema 3.2.6 não é suficientemente forte para garantir a completude de um conjunto de combinadores próprios afins, definimos as equações diofantinas de um conjunto finito Σ de combinadores próprios afins de modo a representar a variação do número de 0's bem como a variação do número de 1's durante a aplicação das regras em $\psi(\Sigma)$. Supomos a partir de agora que $\Sigma = \Sigma_0 \cup \Sigma_T$, onde $\Sigma_T = \{T_1, \ldots, T_t\}$ é o conjunto de todos os selectores em Σ e $\Sigma_0 \cap \Sigma_T = \emptyset$.

Definição 3.3.2

1. *Para toda a regra em $\psi(\Sigma)$ da forma*

$$\underbrace{0\ldots0}_{u_i\geq 0}1 \Leftarrow_1 \underbrace{0\ldots0}_{v_i\geq 0}1$$

seja $t_i = u_i - v_i$, com $1 \leq i \leq m$. Então definimos a equação $eq(\Sigma)$ por

$$\sum_1^m t_i z_i = 1.$$

2. Considerem-se as regras

$$\alpha_i 1 \Leftarrow_1 \beta_i \ \in \psi(\Sigma_0)$$

com $\#_1(\alpha_i) \geq 1$ *e* $1 \leq i \leq l$. *Então sejam*

$$p_i^0 = \#_0(\alpha_i) - \#_0(\beta_i) \quad e \quad p_i^1 = \#_1(\alpha_i).$$

Para as restantes regras

$$\alpha_i \Leftarrow_1 \beta_i \ \in \psi(\Sigma_0)$$

com $\alpha_i \neq \otimes$ *e* $1 \leq i \leq r$ *tomem-se*

$$q_i^0 = \#_0(\alpha_i) - \#_0(\beta_i) \quad e \quad q_i^1 = \#_1(\alpha_i) - 1.$$

Suponha-se ainda que as regras

$$\epsilon \Leftarrow_1 \underbrace{0\ldots0}_{d_i \geq 0}1 \ \in \psi(\Sigma)$$

com $1 \leq i \leq t$ *correspondem aos selectores* $T_1, \ldots, T_t$. *Então definimos o sistema de equações diofantinas* $Eq(\Sigma)$ *por*

$$\begin{cases} \sum_1^l p_i^0 x_i + \sum_1^r q_i^0 y_i - \sum_1^t d_i n_i & = 1 \\ \sum_1^l p_i^1 x_i + \sum_1^r q_i^1 y_i - \sum_1^t n_i & = 0. \end{cases}$$

Lema 3.3.3 *Seja* Σ *um conjunto de combinadores próprios afins. Se todo o combinador próprio em* $\mathcal{C}\ell_{\mathcal{A}}$ *for definível a partir de* Σ, *então*

- $eq(\Sigma)$ *tem solução* $(z_1, \ldots, z_m) \in \mathbb{N}^m$

ou

- $Eq(\Sigma)$ *tem solução* $(x_1, \ldots, x_l, y_1, \ldots, y_r, n_1, \ldots, n_t) \in \mathbb{N}^{l+r+t}$ *tal que* $x_i > 0$ *para algum* $i \in \{1, \ldots, l\}$.

Prova Consideremos mais uma vez a sequência

$$\beta_0 = 01 \Leftarrow_1 \ldots \Leftarrow_1 1 = \beta_n \tag{3.7}$$

e relembremos a correspondência entre os coeficientes de $eq(\Sigma)$ e $Eq(\Sigma)$ e as regras em $\psi(\Sigma)$. Mais precisamente, os coeficientes de $eq(\Sigma)$ bem como os da primeira equação de $Eq(\Sigma)$ representam respectivamente o número de 0's consumidos numa aplicação da regra correspondente, enquanto que os coeficientes da segunda equação de $Eq(\Sigma)$ representam o número de 1's consumidos. Se em (3.7) só foram aplicadas regras da forma

$$0 \ldots 01 \Leftarrow_1 0 \ldots 01,$$

basta tomar z_i igual ao número de aplicações da regra correspondente, para $1 \leq i \leq m$. Assim $\sum_1^m t_i z_i = 1$ decorre de $\#_0(01) - \#_0(1) = 1$.

Se no entanto nem todas as regras aplicadas em (3.7) tiverem a forma $0 \ldots 01 \Leftarrow_1 0 \ldots 01$, então é claro que existe $i \in \{1, \ldots, n-1\}$ tal que $\#_1(\beta_i) > 1$ e consequentemente uma das regras aplicadas é da forma $\alpha 1 \Leftarrow_1 \beta$ com $\#_1(\alpha) \geq 1$. Escolhendo os valores de x_i para $1 \leq i \leq l$, de y_i para $1 \leq i \leq r$ e de n_i para $1 \leq i \leq t$ igual ao número de aplicações da regra correspondente em (3.7) tem-se $x_i > 0$ para algum $i \in \{1, \ldots l\}$ e

$$\begin{cases} \sum_1^l p_i^0 x_i + \sum_1^r q_i^0 y_i - \sum_1^t d_i n_i & = 1 \\ \sum_1^l p_i^1 x_i + \sum_1^r q_i^1 y_i - \sum_1^t n_i & = 0. \end{cases}$$

uma vez que $\#_0(01) - \#_0(1) = 1$ e $\#_1(01) - \#_1(1) = 0$. $\bullet$

Exemplo 3.3.4 *Para* $\Sigma = \{\mathsf{B}, \mathsf{C}, \mathsf{K}\}$ *temos*

$$eq(\Sigma): \quad z_1 - z_2 = 1$$

e

$$Eq(\Sigma): \quad \begin{cases} - y_1 + y_2 - y_3 - n = 1 \\ x - y_1 - y_4 - n = 0 \end{cases}$$

respectivamente com solução $z_1 = 1$, $z_2 = 0$ *e* $x = y_2 = y_4 = 1$, $y_1 = y_3 = n = 0$.

Analogamente à secção anterior obtemos uma caracterização das bases do λ-calculus afim a partir das propriedades estabelecidas no teorema 2.3.11 e nos lemas 3.2.1, 3.2.3, 3.3.1 e 3.3.3.

Teorema 3.3.5 *Um conjunto finito de combinadores próprios afins é uma base para o λ-calculus afim se e só se o conjunto associado Σ de combinadores primitivos satisfizer as condições seguintes.*

1. *Σ contém um selector;*
2. *Σ contém um combinador com efeito de composição;*
3. *Σ contém um combinador com efeito cancelador;*
4. *se nenhum selector T em Σ tiver regra de redução $Tx_1 \dots x_n \geq_{1\omega} x_n$, então existe $\alpha \in \mathcal{S}$ tal que $0\alpha \Leftarrow_1 1 \ \in \ \psi(\Sigma)$;*
5. *existe $\beta \in \mathcal{S}^+$ tal que $\beta 1 \Leftarrow_1 1 \ \in \ \psi(\Sigma)$;*
6. *existe $\gamma \in \mathcal{S}^+$ tal que $\#_0(\gamma) > 0$ e $\gamma \Leftarrow_1 1 \ \in \psi(\Sigma)$;*
7. *a equação $eq(\Sigma)$ tem solução $(z_1, \dots, z_m) \in \mathbb{N}^m$ ou $Eq(\Sigma)$ tem solução $(x_1, \dots, x_l, y_1, \dots, y_r, n_1, \dots, n_t) \in \mathbb{N}^{l+r+t}$ onde $x_i > 0$ para algum $i \in \{1, \dots, l\}$.*

Prova Já vimos no teorema 2.3.11 e nos lemas 3.2.1, 3.2.3, 3.3.1 e 3.3.3, que as propriedades 1.-7. decorrem da completude de Σ. Pelo corolário 3.1.13 basta então provar

$$\forall \alpha \in \mathcal{S} \cup \{\otimes\} \quad \Sigma : \alpha \Leftarrow 1.$$

Como existem combinadores com efeito cancelador em Σ, há pelo menos uma regra da forma $\otimes \Leftarrow_1 0 \dots 01$ em Σ e consequentemente a demonstração de

$$\forall \alpha \in \mathcal{S} \quad \Sigma : \alpha \Leftarrow 1 \tag{3.8}$$

é suficiente para garantir a completude de Σ. Por outro lado, já vimos que (3.8) é satisfeito quando Σ satisfaz as cinco condições do teorema 3.2.7, das quais a última só é usada para provar a existência de $k_0 \geq 0$ tal que $k \geq k_0$ implica

$$\underbrace{0\ldots0}_{k+1} \Leftarrow \underbrace{0\ldots0}_{k}\underbrace{1\ldots1}_{\sum x_i}.$$

Uma vez que as condições 2.-4. do teorema 3.2.7 estão entre as propriedades 1.-7., basta-nos mostrar que

(i) existe $k_0 \geq 0$ tal que $k \geq k_0$ implica $\underbrace{0\ldots0}_{k+1}1 \Leftarrow \underbrace{0\ldots0}_{k}1$

e

(ii) o combinador I é definível a partir de Σ.

Começamos pela prova de **(i)** que se torna trivial, se a equação diofantina $eq(\Sigma)$ tiver solução $(z_1, \ldots, z_m) \in \mathbb{N}^m$, uma vez que se pode aplicar à sequência $\underbrace{0\ldots0}_{k+1}1$ sucessivamente z_i vezes cada regra $\underbrace{0\ldots0}_{u_i}1 \Leftarrow_1 \underbrace{0\ldots0}_{v_i}1$, para $1 \leq i \leq m$, desde que o número de 0's a esquerda seja suficientemente grande. De $\sum_1^m t_i z_i = 1$ resulta que a sequência obtida é da forma $\underbrace{0\ldots0}_{k}1$.

Suponhamos agora que $(x_1, \ldots, x_l, y_1, \ldots, y_r, n_1, \ldots, n_t) \in \mathbb{N}^{l+r+t}$ é uma solução de $Eq(\Sigma)$, tal que pelo menos um dos factores $x \in \{x_1, \ldots, x_l\}$, que corresponde a uma regra da forma $\alpha 1 \Leftarrow_1 \underbrace{0\ldots0}_{b\geq 0}1$ com $\#_1(\alpha) \geq 1$, é positivo. Consideremos o multiconjunto Γ constituído por

$$x_i \ \text{ ocorrências da regra correspondente } \ \alpha_i 1 \Leftarrow_1 \beta_i$$

para $1 \leq i \leq l$, por

$$y_i \ \text{ ocorrências da regra correspondente } \ \alpha_i \Leftarrow_1 \beta_i$$

para $1 \leq i \leq r$, e por

$$n_i \ \text{ ocorrências de } \ \epsilon \Leftarrow_1 \underbrace{0\ldots0}_{d_i}1$$

para $1 \leq i \leq t$.

O resultado **(i)** verifica-se, se for possível aplicar todas as regras em Γ sucessivamente a uma sequência da forma $0\ldots01$. Então resulta de $Eq(\Sigma)$ que a sequência obtida tem menos um 0 do que a sequência inicial e que tem exactamente um 1 que está necessariamente na última posição da sequência. Para mostrar que tal é possível consideremos uma cópia Γ^+ de Γ onde algumas ocorrências dos símbolos ϵ, 0 e 1 na parte esquerda das regras estão marcadas com os sinais $+$ e $-$. Mais precisamente marcamos os 1's, que não estão na última posição da sequência em que ocorrem com o sinal $+$ e marcamos os 0's em última posição bem como todos os ϵ's com o sinal $-$, por exemplo, para

$$\Gamma = \{\epsilon \Leftarrow_1 0\ldots01 \ ,\ \epsilon \Leftarrow_1 0\ldots01 \ ,\ 01100 \Leftarrow_1 0\ldots01 \ ,\ 101 \Leftarrow_1 0\ldots01\}$$

toma-se

$$\Gamma^+ = \{\epsilon^- \Leftarrow_1 0\ldots01 \ ,\ \epsilon^- \Leftarrow_1 0\ldots01 \ ,\ 01^+1^+00^- \Leftarrow_1 0\ldots01 \ ,\ 1^+01 \Leftarrow_1 0\ldots01\}.$$

Veremos primeiro que Γ^+ contém exactamente tantos símbolos positivos como negativos, mostrando que a diferença entre o número de símbolos positivos e o número de símbolos negativos ocorrendo numa regra $\alpha \Leftarrow_1 \beta$, que denotamos por $dif(\alpha \Leftarrow_1 \beta)$, é precisamente o coeficiente desta regra na segunda equação de $Eq(\Sigma)$. Assim o resultado decorre de

$$\sum_1^l p_i^1 x_i + \sum_1^r q_i^1 y_i - \sum_1^t n_i = 0.$$

De facto, para $\#_1(\alpha_i) \geq 1$ temos

$$dif(\alpha_i 1 \Leftarrow_1 \beta_i) = \#_1(\alpha_i) = p_i^1$$

para $\#_1(\alpha_i) = 0$ temos

$$dif(\alpha_i 1 \Leftarrow_1 \beta_i) = \#_1(\alpha_i) = \#_1(\alpha_i 1) - 1 = q_i^1$$

temos

$$dif(\alpha_i 0^- \Leftarrow_1 \beta_i) = \#_1(\alpha_i) - 1 = q_i^1$$

bem como

$$dif(\epsilon^- \Leftarrow_1 \underbrace{0\ldots0}_{d_i}1) = 1.$$

Em vez de mostrar que podemos aplicar todas as regras $\alpha \Leftarrow_1 \beta$ em Γ^+ sucessivamente a uma sequência da forma $0\ldots01$ no sentido habitual, i.e. substituindo uma subsequência da forma α por β, vamos mostrar que o podemos fazer no sentido inverso, i.e. substituindo β por α. Como $(x_1, \ldots, x_l, y_1, \ldots, y_r, n_1, \ldots, n_t)$ é solução de $Eq(\Sigma)$, é claro que a sequência inicial γ_2 tem menos um 0 que a sequência final γ_1, e que se tem $\gamma_1 \Leftarrow \gamma_2$. Pressupondo que γ_2 é da forma $0\ldots01$ segue ainda, pelo modo de aplicação das regras, que o mesmo se verifica para γ_1.

Consideremos então $\gamma_2 = 0\ldots01$, supondo ainda que o número de 0's à esquerda é suficientemente grande para todos os passos que se seguem. Primeiro aplicamos as regras não marcadas em Γ^+, i.e. as da forma

$$\underbrace{0\ldots0}_{a}1 \Leftarrow_1 \underbrace{0\ldots0}_{b}1$$

substituindo

$$0\ldots0\underbrace{0\ldots0}_{b}1 \quad \text{por} \quad 0\ldots0\underbrace{0\ldots0}_{a}1.$$

A seguir aplicamos a regra

$$\alpha 1 \Leftarrow \underbrace{0\ldots0}_{b\geq 0}1$$

com $\#_1(\alpha) \geq 1$, substituindo

$$0\ldots0\underbrace{0\ldots0}_{b}1 \quad \text{por} \quad 0\ldots0\alpha 1.$$

De $\#_1(\alpha) \geq 1$ decorre que a sequência obtida é da forma $0\ldots01^+\beta_1 1$. Pressupondo, que em cada passo a regra utilizada é retirada de Γ^+, note-se que a partir deste momento e até ao fim do processo, Γ^+ contém mais símbolos negativos do que positivos enquanto que o número total de símbolos positivos, i.e. em Γ^+ e $0\ldots01^+\beta_1 1$, é sempre igual ao número de símbolos negativos em Γ^+. Repetimos o passo anterior para as restantes regras da forma

$$\beta_2 1 \Leftarrow_1 0\ldots01$$

substituindo

$$0\ldots01^+\beta_1 1 \quad \text{por} \quad 0\ldots0\beta_2 1^+\beta_1 1.$$

A seguir aplicamos as regras da forma

$$\delta_2 0^- \Leftarrow_1 0\ldots01$$

com $\#_1(\delta_2) \geq 1$, transformando uma sequência da forma

$$0\ldots01^+\delta_1 1 \quad \text{em} \quad 0\ldots0\delta_2 0\delta_1 1.$$

Reparamos também, que cada um destes passos anula um símbolo positivo com um negativo. Finalmente usamos as regras sem símbolos positivos, i.e. os da forma

$$0\ldots00^- \Leftarrow_1 0\ldots01$$

ou

$$\epsilon^- \Leftarrow_1 0\ldots01$$

transformando uma sequência da forma

$$0\ldots01^+\gamma1 \quad \text{em} \quad 0\ldots0\gamma1.$$

Pelas considerações feitas anteriormente, este processo só pára quando $\Gamma^+ = \emptyset$ e produz assim uma sequência da forma $0\ldots01$, o que conclui a prova de **(i)**.

Para provar que o combinador I é definível a partir de Σ, i.e. $\epsilon \Leftarrow 1$, começamos por mostrar duas consequências imediatas das condições 1.-7. de que precisaremos mais tarde. Note-se, que nos basta considerar o caso em que nenhum selector em Σ tem regra de redução $Tx_1\ldots x_n \geq_{1\omega} x_n$ uma vez que neste caso $\epsilon \Leftarrow_1 1 \;\in \psi(\Sigma)$ e de facto I é definível por $\underbrace{TT\ldots T}_{n}$.

C1: $\exists m \geq 1$ tal que $1 \Leftarrow \underbrace{0\ldots0}_{m}1$.

Seja $T \in \Sigma_T$, então existe $d \geq 1$ tal que

$$\epsilon \Leftarrow_1 \underbrace{0\ldots0}_{d}1 \;\in \psi(\Sigma)$$

uma vez que a regra de redução de T não é da forma $Tx_1\ldots x_n \geq_{1\omega} x_n$. De $0\alpha \Leftarrow_1 1 \;\in \psi(\Sigma)$ decorre a existência de

$$\underbrace{1\ldots1}_{p\geq1} \Leftarrow_1 \underbrace{0\ldots0}_{q\geq1}1 \;\in \psi(\Sigma)$$

donde resulta

$$1 \Leftarrow \underbrace{0\ldots0}_{d}11 \Leftarrow \ldots \Leftarrow \underbrace{0\ldots0}_{d(p-1)}\underbrace{1\ldots1}_{p} \Leftarrow \underbrace{0\ldots0}_{d(p-1)+q\geq1}1.$$

C2: $\forall \alpha \in \mathcal{S}\ \exists p \geq 0$ tal que $1 \Leftarrow \underbrace{0\ldots0}_{p}\alpha 1$.

Esta propriedade decorre de $\epsilon \Leftarrow \underbrace{0\ldots0}_{d\geq 1}1$ e $1 \Leftarrow \underbrace{0\ldots0}_{m\geq 1}1$, para d e m como em **C1**.

Voltemos então à prova de **(ii)**, supondo primeiro que existe uma regra da forma

$$0\alpha 1 \Leftarrow_1 1 \quad \in \quad \psi(\Sigma)$$

para algum $\alpha \in \mathcal{S}$.

Se $\#_1(\alpha) = 0$, então $0\alpha 1 = \underbrace{0\ldots0}_{q\geq 1}1$ e consideremos o menor múltiplo vq de q maior ou igual a k_0 em **(i)**. Então obtemos

$$\epsilon \quad \Leftarrow_1 \quad \underbrace{0\ldots0}_{d\geq 1}1 \quad \overset{\mathbf{C1}}{\Leftarrow} \quad \underbrace{0\ldots0}_{t\geq vq}1 \quad \overset{\mathbf{(i)}}{\Leftarrow} \quad \underbrace{0\ldots0}_{vq}1 \quad \Leftarrow_1 \quad \underbrace{0\ldots0}_{(v-1)q}1 \quad \Leftarrow_1 \quad \ldots \quad \Leftarrow_1 \quad 1.$$

Se $\#_1(\alpha) \geq 1$, então $0\alpha 1 = \underbrace{0\ldots0}_{q\geq 1}1\beta 1$ para algum $\beta \in \mathcal{S}$. Seja vq o menor múltiplo de q maior ou igual a k_0. Concluímos,

$$\begin{array}{llll}
\epsilon \ \Leftarrow_1 & \underbrace{0\ldots0}_{d\geq 1}1 & \overset{\mathbf{C2}}{\Leftarrow} & 0\ldots01\underbrace{\beta 1\beta 1\ldots\beta 1}_{v} \\
\overset{\mathbf{C1}}{\Leftarrow} & \underbrace{0\ldots0}_{t\geq vq}1\underbrace{\beta 1\beta 1\ldots\beta 1}_{v} & \overset{\mathbf{(i)}}{\Leftarrow} & \underbrace{0\ldots0}_{vq}1\underbrace{\beta 1\beta 1\ldots\beta 1}_{v} \\
\Leftarrow_1 & \underbrace{0\ldots0}_{(v-1)q}1\underbrace{\beta 1\beta 1\ldots\beta 1}_{v-1} & \Leftarrow_1 & \ldots \Leftarrow_1 1.
\end{array}$$

Se, no entanto, nenhuma regra em $\psi(\Sigma)$ for da forma $0\alpha 1 \Leftarrow_1 1$, então resulta das condições 4. e 5. a existência de duas regras

$$1\beta 1 \Leftarrow_1 1 \quad \text{e} \quad (0\alpha 0 \Leftarrow_1 1 \ \text{ ou } \ 0 \Leftarrow_1 1)$$

com $\alpha, \beta \in \mathcal{S}$.

Se $\#_1(\alpha) \geq 1$, então $0\alpha 0 = \underbrace{0\ldots0}_{q\geq 1}1\gamma 0$ e consideremos vq o menor múltiplo de q maior ou igual a k_0. Então,

$$\begin{array}{llll}
\epsilon \ \Leftarrow_1 & \underbrace{0\ldots0}_{d\geq 1}1 & \overset{\mathbf{C2}}{\Leftarrow} & 0\ldots01\underbrace{\gamma 0\ldots\gamma 0}_{v}\beta 1 \\
\overset{\mathbf{C1}}{\Leftarrow} & \underbrace{0\ldots0}_{t\geq vq}1\underbrace{\gamma 0\ldots\gamma 0}_{v}\beta 1 & \overset{\mathbf{(i)}}{\Leftarrow} & \underbrace{0\ldots0}_{vq}1\underbrace{\gamma 0\ldots\gamma 0}_{v}\beta 1 \\
\Leftarrow_1 & \underbrace{0\ldots0}_{(v-1)q}1\underbrace{\gamma 0\ldots\gamma 0}_{v-1}\beta 1 & \Leftarrow_1 & \ldots \quad \Leftarrow_1 1\beta 1 \Leftarrow_1 1.
\end{array}$$

Finalmente suponhamos $1\beta 1 \Leftarrow_1 1$ e $\underbrace{0\ldots 0}_{n\geq 1} \Leftarrow_1 1$ pertencentes a $\psi(\Sigma)$. Tomando

$$q = (\#_1(\beta)+1)\cdot n + \#_0(\beta)$$

e aplicando $\#_1(\beta)+1$ vezes a regra $\underbrace{0\ldots 0}_{n\geq 1} \Leftarrow_1 1$ nas posições certas obtemos

$$\underbrace{0\ldots 0}_{q}1 \Leftarrow_1 1\underbrace{0\ldots 0}_{q-n}1 \Leftarrow_1 \ldots \Leftarrow_1 1\beta 1 \Leftarrow_1 1.$$

Uma vez que $q \geq 1$, procedemos como no primeiro caso, o que conclui a prova. •

Corolário 3.3.6 *A completude combinatória para o λ-calculus afim é decidível para conjuntos finitos de combinadores próprios.*

Exemplo 3.3.7 *Considere-se o conjunto de combinadores próprios afins*

$$\Gamma = \{\lambda xyz.x, \lambda xyzuv.y(zv)x\}$$

ao qual fica associado o conjunto $\Sigma = \{\mathsf{K}_2, \mathsf{A}\}$, tal que K_2 e A são definidos por

$$\mathsf{K}_2 xyz \geq_{1\omega} x \quad e \quad \mathsf{A}xyzuv \geq_{1\omega} y(zv)x.$$

Então $\psi(\Sigma)$ é constituído por

$$\psi(\mathsf{K}_2) = \begin{cases} \otimes & \Leftarrow_1 & 1 \\ \otimes & \Leftarrow_1 & 01 \\ \epsilon & \Leftarrow_1 & 001 \end{cases} \quad e \quad \psi(\mathsf{A}) = \begin{cases} 011 & \Leftarrow_1 & 1 \\ \otimes & \Leftarrow_1 & 01 \\ 010 & \Leftarrow_1 & 001 \\ 00 & \Leftarrow_1 & 0001 \\ 1 & \Leftarrow_1 & 00001 \end{cases}$$

e Σ satisfaz as condições 1.-6. do teorema 3.3.5. Mesmo assim Γ não forma uma base para o λ-calculus afim, uma vez que nem

$$eq(\Sigma): \quad -4z = 1$$

nem

$$Eq(\Sigma): \begin{cases} x - y_1 - 4y_2 - 2n = 1 \\ x - y_1 - n = 0 \end{cases}$$

admite solução sobre os números naturais.

3.4 Uma condição necessária para o caso geral

Como já referimos anteriormente, é possível representar qualquer combinador M por uma combinação de combinadores próprios, havendo uma infinidade de escolhas diferentes para o conjunto Γ_M de combinadores próprios utilizados na representação de M e também para cada conjunto Γ_M existe uma infinidade de combinações sobre Γ_M que definem o combinador M. Por essa razão não é possível generalizar o resultado da decidibilidade de completude combinatória para conjuntos de combinadores próprios, no λ-calculus linear e no λ-calculus afim, directamente para bases de combinadores em geral. Obtemos no entanto a seguinte condição necessária, que em alguns casos permite inferir a incompletude de um conjunto de combinadores.

Lema 3.4.1 *Seja $\{M_1, \ldots, M_n\}$ uma base de combinadores. Se Γ_{M_i} forem conjuntos de combinadores próprios, tal que cada termo M_i, para $1 \leq i \leq n$, é definível a partir Γ_{M_i}, então o conjunto $\Gamma_{M_1} \cup \ldots \cup \Gamma_{M_n}$ forma uma base de combinadores próprios.* •

Exemplo 3.4.2 *Mostrou-se em (Meredith & Prior, 1963) que os combinadores*

$$\lambda abcd.ad(b(\lambda x.x)c) \quad e \quad \lambda abcd.bd(a(\lambda x.x)c)$$

formam ambos bases unárias para o λ-calculus linear.

Consideremos agora o combinador

$$M = \lambda abcd.ba(d(\lambda x.x)c)$$

que pode ser representado por

$$(\lambda xabcd.ba(dxc))(\lambda x.x)$$

sendo portanto definível a partir do conjunto de combinadores próprios

$$\Gamma_M = \{\lambda x.x, \lambda xabcd.ba(dxc)\}.$$

Associamos a Γ_M o conjunto $\Sigma = \{\mathsf{I}, \mathsf{A}\}$ tal que

$$\mathsf{I}x \geq_{1\omega} x \quad e \quad \mathsf{A}xyzuv \geq_{1\omega} zy(vxu)$$

donde $\psi(\Sigma)$ é constituído pelas regras

$$\psi(\mathsf{I}) = \{\epsilon \ \Leftarrow_1 \ 1 \qquad e \qquad \psi(\mathsf{A}) = \begin{cases} 100 & \Leftarrow_1 & 1 \\ 11 & \Leftarrow_1 & 01 \\ 00 & \Leftarrow_1 & 001 \\ 01 & \Leftarrow_1 & 0001 \\ 101 & \Leftarrow_1 & 00001\ . \end{cases}$$

A partir do teorema 3.2.7 concluímos que este conjunto de combinadores próprios não é completo e consequentemente $\{\lambda abcd.ba(d(\lambda x.x)c)\}$ não forma uma base para o λ-calculus linear.

Capítulo 4

Completude combinatória em $\lambda\mathcal{I}$

Neste capítulo determinamos algumas propriedades das bases do $\lambda\mathcal{I}$-calculus. Foi mostrado em 1986 por Richard Statman que o problema da completude combinatória no λ-calculus puro é indecidível para conjuntos finitos de combinadores em forma normal e consequentemente também para combinadores em geral, (Statman, 1986). Começamos por mostrar que o mesmo resultado ainda é válido quando restringido ao sistema $\lambda\mathcal{I}$, adaptando a prova apresentada por Statman para $\lambda\mathcal{K}$. Depois, na secção 4.2, obtemos uma caracterização das bases do $\lambda\mathcal{I}$-calculus, que são extensões com combinadores próprios de bases do λ-calculus linear, o subsistema natural de $\lambda\mathcal{I}$. Finalmente, na secção 4.3, generalizamos as representações utilizadas no capítulo anterior para combinadores próprios lineares e afins para o $\lambda\mathcal{I}$-calculus e apresentamos uma condição necessária sobre as bases de combinadores próprios neste sistema. Para não sobrecarregar a notação entendemos neste capítulo as palavras *combinador* e *base* respectivamente como *$\lambda\mathcal{I}$-combinador* e *$\lambda\mathcal{I}$-base*.

4.1 Indecidibilidade no caso geral

Na prova da indecidibilidade da completude combinatória para conjuntos finitos de combinadores (em forma normal) no sistema $\lambda\mathcal{I}$, utilizaremos noções e resultados clássicos sobre funções recursivas parciais e a sua definibilidade em $\lambda\mathcal{I}$, que podem ser encontrados no Apêndice A. Necessitamos também do lema seguinte que se encontra provado em (Barendregt, 1981).

Lema 4.1.1 *Um termo $M \in \Lambda_{\mathcal{I}}$ admite forma normal se e só se todo o subtermo de M admitir forma normal.*

A demonstração do resultado que se segue é essencialmente uma adaptação para o $\lambda\mathcal{I}$-calculus da prova do mesmo resultado para $\lambda\mathcal{K}$, cf. (Statman, 1986).

Teorema 4.1.2 *O problema de determinar se um conjunto finito de combinadores em forma normal é uma base para o $\lambda\mathcal{I}$-calculus é indecidível.*

Prova Mostra-se, que o combinador

$$\mathbf{X}_{\mathcal{I}} = \lambda x.x\mathbf{K}^*\mathbf{J}\mathbf{I}$$

onde $\mathbf{K}^*$, $\mathbf{J}$ e $\mathbf{I}$ são respectivamente definidos por

$$\begin{array}{lcl} \mathbf{K}^* & = & \lambda xy.y\mathbf{IIII}x \\ \mathbf{J} & = & \lambda abcd.ab(adc) \\ \mathbf{I} & = & \lambda x.x \end{array}$$

forma uma base unária para o $\lambda\mathcal{I}$-calculus, ver [(Barendregt, 1981), Prop. 9.3.8].

Daqui decorre que um conjunto finito de combinadores Γ é combinatoriamente completo para o $\lambda\mathcal{I}$-calculus se e só se $\mathbf{X}_{\mathcal{I}}$ for definível a partir de Γ; problema que mostraremos ser indecidível.

Dado uma função recursiva parcial φ cujo número de Gödel é e, escrevemos φ como φ_e. Sabe-se, ver Apêndice A, que a função

$$\begin{array}{rccc} F: & \mathbb{N} & \to & \mathbb{N} \\ & e & \rightsquigarrow & \varphi_e(e) \end{array}$$

é uma função recursiva parcial, para qual não é decidível se $F(e)\downarrow$, i.e. se $F(e)$ está definido.

Como F é uma função recursiva parcial, segue ainda que F é definível em $\lambda\mathcal{I}$, i.e. existe um termo $\mathbf{F} \in \Lambda^0_{\mathcal{I}}$ tal que

$$\begin{array}{ll} \mathbf{F}\llcorner e \lrcorner = \llcorner m \lrcorner & \text{se } F(e) = m, \\ \mathbf{F}\llcorner e \lrcorner \text{ não tem forma normal} & \text{se } F(e)\uparrow, \end{array}$$

onde $\llcorner e \lrcorner$ representa o e-ésimo numeral de Church (ver Apêndice A).

Para $e \in \mathbb{N}$ seja

$$\mathbf{N}_e = \mathbf{F}\llcorner e \lrcorner \mathbf{I}\mathbf{I}\mathbf{X}_{\mathcal{I}}.$$

Veremos que o termo $\mathbf{N}_e$ tem forma normal se e só se $F(e)$ estiver definido e que neste caso $\mathbf{N}_e \twoheadrightarrow_\beta \mathbf{X}_{\mathcal{I}}$.

É claro que o subtermo $\mathbf{F}\llcorner e \lrcorner$ tem forma normal se e só se $F(e)\downarrow$. Por outro lado, caso exista, esta forma normal é um numeral de Church, c. Como para qualquer numeral de Church se verifica $c\mathbf{I}\mathbf{I} \twoheadrightarrow_\beta \mathbf{I}$, resulta do lema 4.1.1 que $\mathbf{F}\llcorner e \lrcorner \mathbf{I}\mathbf{I}\mathbf{X}_{\mathcal{I}}$ tem forma normal se e só $F(e)\downarrow$ e que a forma normal é então $\mathbf{X}_{\mathcal{I}}$.

Consideremos agora $\tilde{\mathbf{N}}_e$ obtido a partir de $\mathbf{N}_e$ por substituição de cada redex $(\lambda y.U)V$ por $x(\lambda y.U)V$ e definamos o termo

$$\mathbf{P}_e = \lambda x.xx\mathbf{I}\tilde{\mathbf{N}}_e.$$

Concluímos a prova mostrando que o conjunto de combinadores em forma normal

$$\Gamma_e = \{\mathbf{P}_e, \mathbf{I}\}$$

define o termo $\mathbf{X}_{\mathcal{I}}$ se e só se $\mathbf{N}_e$ admitir forma normal.

De facto, verifica-se

$$\mathbf{P}_e\mathbf{I} \twoheadrightarrow_\beta \mathbf{I}\mathbf{I}\mathbf{I}\mathbf{N}_e \twoheadrightarrow_\beta \mathbf{N}_e$$

e se $\mathbf{N}_e$ admitir forma normal, então

$$\mathbf{N}_e = \mathbf{F}\llcorner e \lrcorner \mathbf{I}\mathbf{I}\mathbf{X}_{\mathcal{I}} \twoheadrightarrow_\beta \mathbf{I}\mathbf{X}_{\mathcal{I}} \twoheadrightarrow_\beta \mathbf{X}_{\mathcal{I}}.$$

Caso contrário, decorre do lema 4.1.1 que $\mathbf{P}_e\mathbf{I}$ não é subtermo de nenhum termo M tal que $M \twoheadrightarrow_\beta \mathbf{X}_{\mathcal{I}}$.

Por outro lado tem-se

$$\begin{aligned}
\mathbf{P}_e\mathbf{P}_e &\twoheadrightarrow_\beta \mathbf{P}_e\mathbf{P}_e\mathbf{I}(\tilde{\mathbf{N}}_e[\mathbf{P}_e/x]) \\
&\twoheadrightarrow_\beta \mathbf{P}_e\mathbf{P}_e\mathbf{I}(\tilde{\mathbf{N}}_e[\mathbf{P}_e/x])\mathbf{I}(\tilde{\mathbf{N}}_e[\mathbf{P}_e/x]) \\
&\twoheadrightarrow_\beta \ldots
\end{aligned}$$

e consequentemente $\mathbf{P}_e\mathbf{P}_e$ também não admite forma normal pela proposição 2.1.10. Como $\mathbf{X}_\mathcal{I}$ é diferente de $\mathbf{I}$ e de $\mathbf{P}_e$ e não é equivalente a nenhum termo envolvendo $\mathbf{P}_e\mathbf{I}$ ou $\mathbf{P}_e\mathbf{P}_e$ como subtermos, o conjunto Γ_e não pode gerar $\mathbf{X}_\mathcal{I}$.

Concluímos que o conjunto de combinadores em forma normal Γ_e é uma base para o $\lambda\mathcal{I}$-calculus se e só se a função F estiver definida em e, i.e. $F(e)\downarrow$. •

4.2 Extensões de bases lineares com combinadores próprios

Em 1986, Statman mostrou o resultado seguinte, que caracteriza as extensões de bases afins com combinadores próprios que formam bases do λ-calculus puro, cf. (Statman, 1986).

Teorema 4.2.1 (Statman) *Se P for um combinador próprio com efeito duplicador, então o conjunto $\{\mathbf{B}, \mathbf{I}, \mathbf{C}_*, \mathbf{K}, P\}$ é uma base para o λ-calculus puro.*

É claro que daqui resulta uma condição necessária e suficiente para que uma extensão de uma base afim com combinadores próprios seja combinatoriamente completa para o λ-calculus puro.

Corolário 4.2.2 *Considere-se uma base Σ para o λ-calculus afim e seja Γ um conjunto de combinadores próprios. Então $\Sigma \cup \Gamma$ forma uma base para o λ-calculus puro se e só se Γ contém um combinador com efeito duplicador.* •

Nesta secção provamos um resultado similar para extensões de bases lineares com combinadores próprios a bases para o $\lambda\mathcal{I}$-calculus. Do teorema 2.3.11 decorre que qualquer base de combinadores próprios para $\lambda\mathcal{I}$ contém necessariamente o combinador $\mathbf{I}$ bem como combinadores com efeito duplicador. Por outro lado nenhum $\lambda\mathcal{I}$-termo tem efeito cancelador e assim é fácil ver, que qualquer base de $\lambda\mathcal{I}$ contém um combinador com um subtermo da forma $\lambda x.N$, com exactamente duas ocorrências livres de x em N. Aos combinadores (próprios) com esta propriedade passamos a chamar combinadores (próprios) com efeito duplicador simples.

Lema 4.2.3 *Qualquer base para o $\lambda\mathcal{I}$-calculus contém um combinador com efeito duplicador simples.* •

O resultado seguinte será utilizado na prova do teorema 4.2.5.

Lema 4.2.4 *Para $m, n, k \geq 0$ seja p o menor inteiro positivo tal que $p \geq n - m - 1$ e considere-se o combinador linear*

$$\mathbf{C}^* = \lambda x_1 \ldots x_{m+k+p+3}.(x_{m+2} \ldots x_{m+k+p+2} x_2 \ldots x_{m+1} x_{m+k+p+3} x_1).$$

Então existem $a, b \geq 0$ tal que para todo o termo X e toda a variável $z \in \mathcal{V}$ se tem

$$\mathbf{C}^* z \underbrace{\mathbf{I} \ldots \mathbf{I}}_{m} (\mathbf{C}^* z \underbrace{\mathbf{I} \ldots \mathbf{I}}_{n}) \underbrace{\mathbf{I} \ldots \mathbf{I}}_{k} X \underbrace{\mathbf{I} \ldots \mathbf{I}}_{p} \rightarrow_\beta X \underbrace{\mathbf{I} \ldots \mathbf{I}}_{a} z \underbrace{\mathbf{I} \ldots \mathbf{I}}_{b} z.$$

Prova Começamos por notar que

$$m + k + p + 3 \geq n + k + 2$$

resulta de $p \geq n - m - 1$ e que para o combinador $\mathbf{C}^*$ se verifica

$$\text{(4.1)} \qquad \mathbf{C}^* x_1 \ldots x_{m+k+p+3} \rightarrow_\beta x_{m+2} \ldots x_{m+k+p+2} x_2 \ldots x_{m+1} x_{m+k+p+3} x_1.$$

Consequentemente temos

$$\begin{aligned}
\mathbf{C}^* z \underbrace{\mathbf{I} \ldots \mathbf{I}}_{m} (\mathbf{C}^* z \underbrace{\mathbf{I} \ldots \mathbf{I}}_{n}) \underbrace{\mathbf{I} \ldots \mathbf{I}}_{k} X \underbrace{\mathbf{I} \ldots \mathbf{I}}_{p} &\rightarrow_\beta \mathbf{C}^* z \underbrace{\mathbf{I} \ldots \mathbf{I}}_{n} \underbrace{\mathbf{I} \ldots \mathbf{I}}_{k} X \underbrace{\mathbf{I} \ldots \mathbf{I}}_{p-1} \underbrace{\mathbf{I} \ldots \mathbf{I}}_{m} \mathbf{I} z \\
&\rightarrow_\beta \mathbf{I} \ldots \mathbf{I} X \underbrace{\mathbf{I} \ldots \mathbf{I}}_{a} z \underbrace{\mathbf{I} \ldots \mathbf{I}}_{b} z \\
&\rightarrow_\beta X \underbrace{\mathbf{I} \ldots \mathbf{I}}_{a} z \underbrace{\mathbf{I} \ldots \mathbf{I}}_{b} z.
\end{aligned}$$

O primeiro passo de redução resulta da aplicação directa de (4.1). Por outro lado, verifica-se o seguinte:

(i) O combinador $\mathbf{C}^*$ tem aridade $m + k + p + 3$;

(ii) A primeira ocorrência de z em $\mathbf{C}^* z \underbrace{\mathbf{I}\ldots\mathbf{I}}_{n}\underbrace{\mathbf{I}\ldots\mathbf{I}}_{k} X \underbrace{\mathbf{I}\ldots\mathbf{I}}_{p-1}\underbrace{\mathbf{I}\ldots\mathbf{I}}_{m}\mathbf{I}z$ é o primeiro termo depois de $\mathbf{C}^*$ e corresponde assim a x_1 em (4.1);

(iii) O termo X corresponde a x_{n+k+2} em (4.1) e é argumento de $\mathbf{C}^*$ por $m+k+p+3 \geq n+k+2$;

(iv) A segunda ocorrência de z é o $(n+k+p+m+3)$-ésimo termo depois de $\mathbf{C}^*$, e não é argumento de $\mathbf{C}^*$ a não ser que se tenha $n = 0$. Neste caso, esta ocorrência de z corresponde ao último argumento de $\mathbf{C}^*$.

Pelas considerações anteriores conclui-se que depois do segundo passo de redução, o termo X está à esquerda das duas ocorrências de z, das quais uma está em último lugar e que todos os outros subtermos são ocorrências do combinador $\mathbf{I}$. Os restantes passos resultam da redução com $\mathbf{I}$. $\bullet$

Teorema 4.2.5 *Considere-se uma base Σ para o λ-calculus linear e seja Γ um conjunto de combinadores próprios. Então $\Sigma \cup \Gamma$ forma uma base para o $\lambda\mathcal{I}$-calculus se e só se Γ contém um combinador com efeito duplicador simples.*

Prova A existência de um combinador com efeito duplicador simples em Γ segue directamente do lema 4.2.3 e pelo facto de Σ não conter nenhum combinador com efeito duplicador.

Consideremos agora um combinador $\mathbf{D} \in \Gamma$ com efeito duplicador simples ao qual está associado a regra de redução

$$\mathbf{D}x_1 \ldots z \ldots x_d \;\to_\beta\; Z$$

tal que a variável z ocorre exactamente duas vezes em Z. De $\mathbf{I}x \to_\beta x$ decorre a existência de inteiros $m, n, k \geq 0$ tais que

$$\mathbf{D}\underbrace{\mathbf{I}\ldots\mathbf{I}z\mathbf{I}\ldots\mathbf{I}}_{d} \;\to_\beta\; z\underbrace{\mathbf{I}\ldots\mathbf{I}}_{m}(z\underbrace{\mathbf{I}\ldots\mathbf{I}}_{n})\underbrace{\mathbf{I}\ldots\mathbf{I}}_{k} \tag{4.2}$$

para os quais consideramos o combinador próprio linear $\mathbf{C}^*$ e os inteiros $a, b \geq 0$ definidos como no lema 4.2.4. Note-se que $\mathbf{C}^*$ é definível a partir de Σ por esta ser uma base linear.

Consideremos então um termo aplicativo X sobre $\{x_1, \ldots, x_N\}$ tal que cada uma das variáveis x_i, para $1 \leq i \leq N$, ocorre pelo menos uma vez em X. Veremos como definir uma combinação M sobre $\Sigma \cup \Gamma$ tal que

$$Mx_1 \ldots x_N \to_\beta X.$$

Já vimos anteriormente que a definibilidade de M em $\Sigma \cup \Gamma$ equivale à existência de uma sequência da forma

$$X = X_0 \leftarrow_\beta X_1 x_N \leftarrow_\beta \ldots \leftarrow_\beta X_N x_1 \ldots x_N$$

tal que para $1 \leq j \leq N$ o termo X_j é uma combinação sobre $\{x_1, \ldots, x_{N-j}\} \cup \Sigma \cup \Gamma$.

Se x_N ocorre exactamente uma vez em X, é possível expandir X para $X_1 x_N$ por Σ ser uma base linear. Caso contrário, seja $t \geq 2$ o número de ocorrências de x_N em X. Vamos mostrar que X é expansível para um termo da forma $X^1 x_N$ com $t-2$ ocorrências de x_N em X^1. Para $t > 2$ repetimos este processo para $X^1 x_N$, i.e. expandindo $X^1 x_N$ para $X^2 x_N$ com $t-3$ ocorrências de x_N em X^2 até obter

$$X \leftarrow_\beta X^1 x_N \leftarrow_\beta X^2 x_N \leftarrow_\beta \ldots \leftarrow_\beta X^{t-1} x_N = X_1 x_N$$

tal que X_1 é uma combinação sobre $\{x_1, \ldots, x_{N-1}\} \cup \Sigma \cup \Gamma$.

Suponhamos que x_N tem $t \geq 2$ ocorrências em X. Como Σ é uma base para o λ-calculus linear é possível expandir o termo X com combinadores em Σ obtendo-se

$$\begin{aligned} X &\leftarrow_\beta \underbrace{\mathbf{I} \ldots \mathbf{I}}_{a+b} X \\ &\leftarrow_\beta X' \underbrace{\mathbf{I} \ldots \mathbf{I}}_{a} x_N \underbrace{\mathbf{I} \ldots \mathbf{I}}_{b} x_N. \end{aligned}$$

Assim, resulta de (4.2) e do lema 4.2.4

$$\begin{aligned} X' \underbrace{\mathbf{I} \ldots \mathbf{I}}_{a} x_N \underbrace{\mathbf{I} \ldots \mathbf{I}}_{b} x_N &\leftarrow_\beta \mathbf{C}^* x_N \underbrace{\mathbf{I} \ldots \mathbf{I}}_{m} (\mathbf{C}^* x_N \underbrace{\mathbf{I} \ldots \mathbf{I}}_{n}) \underbrace{\mathbf{I} \ldots \mathbf{I}}_{k} X' \underbrace{\mathbf{I} \ldots \mathbf{I}}_{p} \\ &\leftarrow_\beta \mathbf{D}\mathbf{I} \ldots \mathbf{I}(\mathbf{C}^* x_N)\mathbf{I} \ldots \mathbf{I} X' \underbrace{\mathbf{I} \ldots \mathbf{I}}_{p} \\ &\leftarrow_\beta X^1 x_N. \end{aligned}$$

Depois de se obter $X \leftarrow_\beta X_1 x_N$ basta repetir o mesmo para $x_{N-1}, \ldots, x_1$. •

4.3 Bases de combinadores próprios para o $\lambda\mathcal{I}$-calculus

Nesta secção estabelecemos mais uma condição necessária para um conjunto de combinadores próprios formar uma base para o $\lambda\mathcal{I}$-calculus. Começamos por generalizar a representação, utilizada no capítulo anterior para combinadores próprios lineares e afins, para combinadores próprios com efeito duplicador[1]. Consideramos então um combinador próprio M num sistema de lógica combinatória com regra de redução

$$Mx_0 \ldots x_n \geq_{1\omega} Z$$

tal que cada uma das variáveis $x_0, \ldots, x_n$ ocorre pelo menos uma vez em Z.

Suponhamos que, para $0 \leq i \leq n$, cada variável x_i tem n_i ocorrências em Z, respectivamente com posições

$$\alpha^1_{n-i}, \ldots, \alpha^{n_i}_{n-i}.$$

Então representamos o combinador M por

$$\psi(M) \quad = \quad \left\{ \begin{array}{l} \left[\begin{array}{c} \alpha^1_0 \\ \vdots \\ \alpha^{n_n}_0 \end{array} \right. \quad \Leftarrow_1 \quad 1 \\ \qquad\qquad \vdots \\ \left[\begin{array}{c} \alpha^1_n \\ \vdots \\ \alpha^{n_0}_n \end{array} \right. \quad \Leftarrow_1 \quad \underbrace{0 \ldots 0}_{n}1 \end{array} \right.$$

Em $\psi(M)$ chamamos a uma regra

$$\left[\begin{array}{c} \alpha^1_{n-i} \\ \vdots \\ \alpha^{n_i}_{n-i} \end{array} \right. \quad \Leftarrow_1 \underbrace{0 \ldots 0}_{n-i}1$$

[1] Para não sobrecarregar a notação no que se segue fazemos a extensão somente para combinadores sem efeito cancelador. A generalização para combinadores próprios em $\lambda\mathcal{K}$ é trivial.

independente quando $n_i = 1$, i.e. as regras independentes são as da forma

$$\alpha_{n-i} \Leftarrow_1 \underbrace{0\ldots0}_{n-i}1.$$

Exemplo 4.3.1 *O combinador* S *definido por* $\mathsf{S}xyz \geq_{1\omega} xz(yz)$ *é representado por*

$$\psi(\mathsf{S}) \;=\; \left\{ \begin{array}{lll} \left[\begin{array}{l} 01 \\ 11 \end{array} \right. & \Leftarrow_1 & 1 \\ 10 & \Leftarrow_1 & 01 \\ 00 & \Leftarrow_1 & 001 \end{array} \right.$$

com regras independentes

$$10 \Leftarrow_1 01 \qquad e \qquad 00 \Leftarrow_1 001.$$

Consideremos agora um termos aplicativo Y e a função correspondente

$$t_Y : A_Y \longrightarrow \mathcal{V} \cup \mathcal{C}.$$

É imediato que Y pode ser expandido para um termo X com o combinador M se e só se existir $\alpha \in \mathcal{S}$ tal que para $0 \leq i \leq n$ se tem

$$A_Y(\alpha\alpha^1_{n-i}) = \ldots = A_Y(\alpha\alpha^{n_i}_{n-i}) = \{\beta_1, \ldots, \beta_{m_i}\} \neq \emptyset$$

e para $1 \leq j \leq m_i$

$$t_Y(\alpha\alpha^1_{n-i}\beta_j) = \ldots = t_Y(\alpha\alpha^{n_i}_{n-i}\beta_j).$$

Então obtém-se t_X a partir de t_Y substituindo em A_Y as sequências

$$\alpha\alpha^1_{n-i}\beta_j, \ldots, \alpha\alpha^{n_i}_{n-i}\beta_j \quad \text{por} \quad \alpha\underbrace{0\ldots0}_{n-i}1\beta_j$$

para $1 \leq j \leq m_i$, e mantendo o valor de t_Y para estes argumentos, i.e. tomando

$$t_X(\alpha\underbrace{0\ldots0}_{n-i}1\beta_j) = t_Y(\alpha\alpha^1_{n-i}\beta_j)$$

e juntando ainda a sequência $\alpha\underbrace{0\ldots0}_{n+1}$ ao domínio, com

$$t_X(\alpha\underbrace{0\ldots0}_{n+1}) = M.$$

Exemplo 4.3.2 *A expansão do termo*

$$Y = u(y(\mathsf{B}x)(z(\mathsf{B}x)))$$

com o combinador S *na posição* 1 *é possível por*

$$A_Y(1\cdot 10) = \{\epsilon\} \neq \emptyset,$$

$$A_Y(1\cdot 00) = \{\epsilon\} \neq \emptyset,$$

$$A_Y(1\cdot 01) = A_Y(1\cdot 11) = \{0,1\} \neq \emptyset,$$

$$t_Y(1\cdot 01\cdot 0) = t_Y(1\cdot 11\cdot 0) = \mathsf{B}$$

e

$$t_Y(1\cdot 01\cdot 1) = t_Y(1\cdot 11\cdot 1) = x$$

e traduz-se no quadro seguinte.

	$u(y(\mathsf{B}x)(z(\mathsf{B}x)))$	$\leq_{1\omega}$	$u(\mathsf{S}yz(\mathsf{B}x))$
u :	0	$\Leftarrow_1$	0
z :	110	$\Leftarrow_1$	101
y :	100	$\Leftarrow_1$	1001
x :	1011	$\Leftarrow_1$	111
x :	1111		
B :	1010	$\Leftarrow_1$	110
B :	1110		
S :			1000

Consideremos agora um conjunto finito $\Gamma \subseteq \Lambda^0_{\mathcal{I}}$ de combinadores próprios, ao qual está associado o conjunto de combinadores primitivos Σ num sistema de lógica combinatória $\mathcal{C}\ell_\Sigma$, cf. lema 2.3.13. Decorre do lema 4.2.3 e do teorema 4.2.5 que Γ é uma base para $\lambda\mathcal{I}$ se e só se Σ contiver um combinador com efeito duplicador simples e se for possível definir a partir de Σ todos os combinadores lineares, ou equivalentemente os combinadores I, B e C. Uma vez que não existem combinadores com efeito cancelador em Σ, pode-se concluir que terá que ser possível re-escrever qualquer sequência $\alpha \in \mathcal{S}$ para 1, usando para isso somente as regras independentes de $\psi(\Sigma)$. Denotamos, daqui em diante, o conjunto formado pelas regras independentes de $\psi(\Sigma)$ por $Ind(\Sigma)$, e classificamo-lo como linearmente completo se

$$Ind(\Sigma): \ \alpha \Leftarrow 1 \qquad \forall \alpha \in \mathcal{S}.$$

Lema 4.3.3 *Seja Σ um conjunto finito de combinadores próprios. Se todo o combinador próprio em $\mathcal{C}\ell_{\mathcal{I}}$ for definível a partir de Σ, então o conjunto das regras independentes $Ind(\Sigma)$ é linearmente completo.* •

Relembramos no entanto que o conjunto de regras $Ind(\Sigma)$ não corresponde, em geral, a um conjunto de combinadores próprios lineares. Chamamos ainda atenção para que o facto de $Ind(\Sigma)$ ser linearmente completo não implicar a completude de Σ, mesmo quando Σ contenha um combinador próprio com efeito duplicador simples (como no exemplo seguinte), pois a aplicabilidade das regras nem sempre é garantida.

Exemplo 4.3.4 *Considere-se o conjunto de combinadores próprios*

$$\Gamma = \{\lambda x.x, \lambda xy.yx, \lambda xy.x(xy)\}$$

ao qual associamos $\Sigma = \{\mathsf{I}, \mathsf{C}_*, \mathsf{A}\}$ tal que

$$\begin{array}{rlll} & \mathsf{I}x & \geq_{1\omega} & x \\ & \mathsf{C}_*xy & \geq_{1\omega} & yx \\ e & \mathsf{A}xy & \geq_{1\omega} & x(xy). \end{array}$$

Statman mostrou que o combinador B, com $\mathsf{B}xyz \geq_{1\omega} x(yz)$, não é definível somente a partir de combinadores próprios de ordem ≤ 2, onde a ordem de um combinador com regra de redução

$$Mx_1 \ldots x_n \geq_{1\omega} X$$

é o inteiro n, cf. (Statman, 1986). Consequentemente Γ *não pode ser combinatoriamente completo para* $\lambda\mathcal{I}$*. No entanto tem-se*

$$\psi(\mathsf{I}) = \{\epsilon \Leftarrow_1 1 \qquad \psi(\mathsf{C}_*) = \begin{cases} 0 \Leftarrow_1 1 \\ 1 \Leftarrow_1 01 \end{cases} \qquad \psi(\mathsf{A}) = \begin{cases} 11 \Leftarrow_1 1 \\ \left[\begin{array}{l} 0 \\ 10 \end{array}\right. \Leftarrow_1 01 \end{cases}$$

donde

$$Ind(\Sigma) = \{\epsilon \Leftarrow_1 1, 0 \Leftarrow_1 1, 1 \Leftarrow_1 01, 11 \Leftarrow_1 1\}$$

e verifica-se facilmente que o conjunto de regras independentes $Ind(\Sigma)$ *é linearmente completo, uma vez que* $0 \Leftarrow_1 1 \in Ind(\Sigma)$ *e* $11 \Leftarrow_1 1 \in Ind(\Sigma)$.

No resultado seguinte determinamos condições necessárias e suficientes para um conjunto de regras independentes $\Delta = Ind(\Sigma)$ ser linearmente completo. Para esse efeito definimos a equação diofantina do conjunto de regras Δ como na definição 3.2.4, i.e. representando pelos coeficientes da equação diofantina a variação do número de 0's na aplicação de cada regra em Δ.

Teorema 4.3.5 *Considere-se um conjunto finito* Δ *constituído pelas regras*

$$\begin{array}{lll} \epsilon & \Leftarrow_1 & 1, \\ \alpha_1 & \Leftarrow_1 & \underbrace{0\ldots0}_{n_1}1, \\ & \ldots & \\ \alpha_m & \Leftarrow_1 & \underbrace{0\ldots0}_{n_m}1 \end{array}$$

com $\alpha_i \in \mathcal{S}^+$ *e* $n_i \geq 0$*, para* $1 \leq i \leq m$*. Então tem-se*

$$\Delta : \alpha \Leftarrow 1 \quad \forall \alpha \in \mathcal{S}$$

se e só se as condições seguintes são satisfeitas.

1. *A equação diofantina de* Δ *é solúvel;*

2. $\exists \gamma, \delta \in \mathcal{S} : \ \gamma 1 \delta 1 \Leftarrow_1 0\ldots01 \ \in \Delta$*;*

3. **a)** $\underbrace{1\ldots1}_{\geq 2} \Leftarrow_1 1 \;\in \Delta$

ou

b) $\begin{cases} \exists \alpha, \beta \in \mathcal{S} : \;\alpha 0 \beta 1 \Leftarrow_1 1 \;\in \Delta \\ \underbrace{1\ldots1}_{\geq 1} \Leftarrow_1 \underbrace{0\ldots0}_{\geq 1} 1 \;\in \Delta \end{cases}$

4. **a)** $\underbrace{1\ldots1}_{\geq 0} 0 \underbrace{1\ldots1}_{\geq 0} \Leftarrow_1 1 \;\in \Delta$

ou

b) $\exists \alpha, \beta, \gamma \in \mathcal{S} : \;\alpha 0 \beta 0 \gamma \Leftarrow_1 1 \;\in \Delta$

$$\textbf{e} \begin{cases} \underbrace{1\ldots1}_{\geq 0} 0 \underbrace{1\ldots1}_{\geq 0} \Leftarrow_1 \underbrace{0\ldots0}_{\geq 2} 1 \;\in \Delta \\ \textbf{ou} \\ \underbrace{1\ldots1}_{\geq 1} \Leftarrow_1 \underbrace{0\ldots0}_{\geq 1} 1 \;\in \Delta \end{cases}$$

Prova Começamos por mostrar que as condições enumeradas são necessárias.

A existência de solução da equação diofantina de Δ resulta de $\Delta : 01 \Leftarrow 1$ pelas considerações feitas na prova do lema 3.2.6.

Também de $\Delta : 01 \Leftarrow 1$ decorre a condição **4.** uma vez que em

$$01 = \beta_0 \Leftarrow_1 \beta_1 \Leftarrow_1 \ldots \Leftarrow_1 \beta_k \Leftarrow_1 1$$

se tem $\sharp_0(\beta_i) = 0$ e $\sharp_0(\beta_{i-1}) = 1$ para algum $i \in \{1, \ldots, k\}$, ou então existe $i \in \{1, \ldots, k\}$ tal que $\sharp_0(\beta_i) \geq 2$ e $\sharp_0(\beta_{i-1}) = 1$. No primeiro caso, β_i resulta de β_{i-1} por aplicação de uma regra da forma

$$\underbrace{1\ldots1}_{\geq 0} 0 \underbrace{1\ldots1}_{\geq 0} \Leftarrow_1 1$$

enquanto que no segundo caso β_i resulta de β_{i-1} por aplicação de uma regra da forma

$$\underbrace{1\ldots1}_{\geq 0} 0 \underbrace{1\ldots1}_{\geq 0} \Leftarrow_1 \underbrace{0\ldots0}_{\geq 2} 1 \qquad \text{ou} \qquad \underbrace{1\ldots1}_{\geq 1} \Leftarrow_1 \underbrace{0\ldots0}_{\geq 1} 1.$$

Como $\sharp_0(\beta_i) \geq 2$ e se não existir nenhuma regra $\underbrace{1\ldots1}_{\geq 0} 0 \underbrace{1\ldots1}_{\geq 0} \Leftarrow_1 1$ em Δ, então é claro que existem $\alpha, \beta, \gamma \in \mathcal{S}$ tal que $\alpha 0 \beta 0 \gamma \Leftarrow_1 1 \;\in \Delta$.

Temos também $\Delta : 11 \Leftarrow 1$ ou equivalentemente uma sequência da forma

$$11 = \beta_0 \Leftarrow_1 \beta_1 \Leftarrow_1 \ldots \Leftarrow_1 \beta_k \Leftarrow_1 1.$$

Verifica-se facilmente que pelo menos uma das regras aplicadas é da forma $\gamma 1 \delta 1 \Leftarrow_1 0 \ldots 01$ com $\gamma, \delta \in \mathcal{S}$.

Se $\sharp_0(\beta_i) = 0$ para $0 \leq i \leq k$, então existe $\underbrace{1 \ldots 1}_{\geq 2} \Leftarrow_1 1 \ \in \Delta$. Caso contrário, existem duas regras $\underbrace{1 \ldots 1}_{\geq 1} \Leftarrow_1 \underbrace{0 \ldots 0}_{\geq 1} 1$ e $\alpha 0 \beta 1 \Leftarrow_1 1$ em Δ com $\alpha, \beta \in \mathcal{S}$.

Veremos agora que as condições enumeradas são suficientes para que

$$\Delta : \ \alpha \Leftarrow 1 \quad \forall \alpha \in \mathcal{S}.$$

Como $\epsilon \Leftarrow_1 1 \ \in \Delta$, basta mostrar

$$\Delta : 11 \Leftarrow 1 \qquad \text{e} \qquad \Delta : 0 \Leftarrow 1.$$

Se a condição **3.a)** for válida, então temos $\Delta : 11 \Leftarrow 1$ por $\epsilon \Leftarrow_1 1$. Caso contrário, verifica-se **3.b)**, i.e. $\alpha 0 \beta 1 \Leftarrow_1 1$ e $\underbrace{1 \ldots 1}_{p \geq 1} \Leftarrow_1 \underbrace{0 \ldots 0}_{m \geq 1} 1$. Para $d = \sharp_0(\alpha 0 \beta 1) \geq 1$ segue $\underbrace{0 \ldots 0}_{d} 1 \Leftarrow 1$ por $\epsilon \Leftarrow_1 1$. Se $p \geq 2$, então obtemos

$$\begin{array}{rllll} 11 & \Leftarrow \underbrace{1 \ldots 1}_{p} & \Leftarrow \underbrace{0 \ldots 0}_{m} 1 & \Leftarrow & \underbrace{0 \ldots 0}_{m} \underbrace{1 \ldots 1}_{p} \\ & \Leftarrow \underbrace{0 \ldots 0}_{2m} 1 & \Leftarrow \ \ldots & \Leftarrow & \underbrace{0 \ldots 0}_{dm} 1 \\ & \Leftarrow \underbrace{0 \ldots 0}_{d(m-1)} 1 & \Leftarrow \ \ldots & \Leftarrow & 1. \end{array}$$

Suponhamos agora que $p = 1$, i.e. $1 \Leftarrow_1 \underbrace{0 \ldots 0}_{m} 1 \ \in \Delta$. Sejam $a = \sharp_0(\gamma) \geq 0$ e $b = \sharp_0(\delta) \geq 0$ para $\gamma 1 \delta 1 \Leftarrow_1 \underbrace{0 \ldots 0}_{e \geq 0} 1$ da condição 2. Então resulta $\underbrace{0 \ldots 0}_{a} 1 \underbrace{0 \ldots 0}_{b} 1 \Leftarrow \underbrace{0 \ldots 0}_{e} 1$ de $\epsilon \Leftarrow_1 1$.

Usando $1 \Leftarrow_1 \underbrace{0\ldots0}_{m}1$, quando necessário obtemos

$$\begin{aligned}
11 &\Leftarrow \underbrace{0\ldots0}_{mda}1\underbrace{0\ldots0}_{mdb}1 \\
&\Leftarrow \underbrace{0\ldots0}_{(md-1)a}\ \underbrace{0\ldots0}_{a}1\underbrace{0\ldots0}_{b}1\ \underbrace{0\ldots0}_{(md-1)b}\ 1 \\
&\Leftarrow \underbrace{0\ldots0}_{e}\ \underbrace{0\ldots0}_{(md-1)a}\ 1\ \underbrace{0\ldots0}_{(md-1)b}\ 1 \\
&\Leftarrow \ldots \Leftarrow \underbrace{0\ldots0}_{mde}1 \\
&\Leftarrow \underbrace{0\ldots0}_{(me-1)d}\ 1 \Leftarrow \ldots \Leftarrow 1.
\end{aligned}$$

Concluímos a prova com a demonstração de $\Delta : 0 \Leftarrow 1$, que se verifica facilmente por $\epsilon \Leftarrow_1 1$ se $\underbrace{1\ldots1}_{\geq 0}0\underbrace{1\ldots1}_{\geq 0} \Leftarrow_1 1 \in\ \Delta$.

Caso contrário, verificam-se as condições em **4.b)**. Veremos primeiro que existe $k_0 \geq 0$ tal que $k \geq k_0$ implica $\underbrace{0\ldots0}_{k+1} \Leftarrow \underbrace{0\ldots0}_{k}1$. De facto, para k suficientemente grande resulta $\underbrace{0\ldots0}_{k+1} \Leftarrow \underbrace{0\ldots0}_{k}1\ldots1$ por aplicações sucessivas das regras correspondentes a uma solução da equação diofantina de Δ, usando $\epsilon \Leftarrow_1 1$ quando necessário. Assim segue $\underbrace{0\ldots0}_{k+1} \Leftarrow \underbrace{0\ldots0}_{k}1$ por $11 \Leftarrow 1$.

Seja agora $d = \sharp_0(\alpha 0 \beta 0 \gamma) \geq 2$ para $\alpha 0 \beta 0 \gamma \Leftarrow_1 1$ da condição **4.b)**. Então obtemos

$$0 \Leftarrow \left\{ \begin{array}{c} 1\ldots101\ldots1 \\ \text{ou} \\ 01\ldots1 \end{array} \right. \Leftarrow_1 \underbrace{0\ldots0}_{\geq 2}1.$$

Iterando decorre, por $11 \Leftarrow 1$,

$$\begin{aligned}
0 &\Leftarrow \underbrace{0\ldots0}_{t\geq k_0}1 &&\Leftarrow \underbrace{0\ldots0}_{qd\geq k_0}1 \\
&\Leftarrow \underbrace{0\ldots0}_{(q-1)d}\alpha 0\beta 0\gamma 1 &&\Leftarrow \underbrace{0\ldots0}_{(q-1)d}11 \\
&\Leftarrow 1\ldots1 &&\Leftarrow 1.
\end{aligned}$$

•

Exemplo 4.3.6 *O conjunto de combinadores próprios*

$$\Gamma = \{\lambda x.x, \lambda xyz.x(yz), \lambda xy, yyx\}$$

ao qual fica associado $\Sigma = \{\mathsf{I}, \mathsf{B}, \mathsf{W}'\}$ *com*

$$\begin{array}{rlll} & \mathsf{I}x & \geq_{1\omega} & x \\ & \mathsf{B}xyz & \geq_{1\omega} & x(yz) \\ e & \mathsf{W}'xy & \geq_{1\omega} & yyx \end{array}$$

não é completo para $\lambda\mathcal{I}$*, pois*

$$Ind(\Sigma) = \{\epsilon \Leftarrow_1 1, 11 \Leftarrow_1 1, 10 \Leftarrow_1 01, 0 \Leftarrow_1 001, 1 \Leftarrow_1 01\}$$

não satisfaz a condição **4.** *do teorema 4.3.5.*

Capítulo 5

Sistemas tipados de λ-calculus

5.1 Introdução

Existe um vasto número de versões tipadas do λ-calculus que se divide essencialmente em dois grupos. Os sistemas tipados à la Curry e os sistemas tipados à la Church, respectivamente baseados nas abordagens feitas por Curry em (Curry, 1934)[1] e por Church em (Church, 1940). Os tipos são objectos sintácticos, que podem ser associados a λ-termos para especificar o seu carácter funcional. Assim, um termo ao qual está associado o tipo $A \to B$ representa geralmente uma função de A em B. As duas abordagens referidas diferem principalmente no facto dos termos tipados à la Church estarem explicitamente anotados com tipos, enquanto que os termos dos sistemas de Curry são λ-termos não tipados aos quais podem ser atribuídos tipos segundo regras estabelecidas. Os sistemas de Curry designam-se, por essa razão, também por sistemas de λ-calculus com atribuição de tipos.

Entre sistemas tipados de λ-calculus e lógicas intuicionistas há uma correspondência surpreendente, geralmente conhecida sob o nome de *isomorfismo de Curry-Howard* ou também como *interpretação de proposições-como-tipos*. Este isomorfismo surgiu em 1968 como resultado de estudos independentes efectuados por Howard e de Bruijn (expostos em (Howard, 1980) e (de Bruijn, 1970)), motivados por observações em (Curry & Feys, 1958) e (Tait, 1965). Dado um sistema de λ-calculus e a lógica correspondente, estabelece-se uma bijecção entre

[1]Neste artigo Curry introduziu a atribuição de tipos para a lógica combinatória, generalizando-a mais tarde em (Curry & Feys, 1958) e (Curry *et al.*, 1972) para o λ-calculus.

o conjunto de termos tipados do sistema de λ-calculus e o conjunto de deduções naturais, conceito introduzido por Prawitz em (Prawitz, 1965), dos teoremas (proposições) da lógica. Esta bijecção é, de facto, um isomorfismo, uma vez que preserva conceitos importantes como os de redução e forma normal, existentes e introduzidos independentemente nos dois sistemas, cf. (Kreisel, 1968) e (Ungar, 1992). Sob este ponto de vista o λ-calculus é por vezes visto como sistema adequado para estudar a natureza funcional implicitamente inerente às deduções formais de proposições em lógicas intuicionistas, cf. (Girard *et al.*, 1989). Outra aplicação importante desta correspondência encontra-se em sistemas de verificação de demonstrações, baseados na interpretação de proposições-como-tipos, como por exemplo o sistema AUTOMATH descrito em (de Bruijn, 1980).

Neste capítulo estudamos algumas propriedades do λ-calculus puro e seus subsistemas, simplesmente tipados à la Curry, que introduzimos na subsecção 5.2.1. A versão tipada do λ-calculus puro corresponde, via isomorfismo de Curry-Howard, ao cálculo proposicional minimal[2] $P(\rightarrow)$, também designado por cálculo proposicional implicativo positivo. Em particular verifica-se que um tipo é atribuível a um λ-termo fechado se e só se o mesmo tipo, interpretado como fórmula, for um teorema do cálculo $P(\rightarrow)$. Igualmente, relacionam-se os subsistemas linear, afim e o sistema $\lambda\mathcal{I}$ com subcálculos importantes de $P(\rightarrow)$, que são descritos na subsecção 5.2.2. Na secção 5.3 generalizamos o conceito de base de combinadores para sistemas tipados e, analogamente ao que foi feito para o sistema $\lambda\mathcal{K}$ por Statman, cf. (Statman, 1986), mostramos a equivalência entre a completude combinatória de um conjunto $\Sigma(\rightarrow)$ de combinadores tipados, para qualquer um dos subsistemas referidos, e a completude axiomática do conjunto de tipos que ocorrem em $\Sigma(\rightarrow)$, quando interpretados como fórmulas, para o subcálculo proposicional correspondente. Este resultado é utilizado na secção 5.4, onde estabelecemos, para cada um dos quatro casos, as relações entre a completude combinatória de um conjunto de combinadores não tipados $\{M_1, \ldots, M_n\}$ e a completude axiomática do conjunto de todos os tipos atribuíveis a $M_1, \ldots, M_n$, interpretados como fórmulas, para o cálculo proposicional correspondente. Finalmente, na secção 5.5, mostramos que todo o λ-termo afim M em forma normal fica completamente caracterizado pelo conjunto de tipos que lhe podem ser atribuídos, representados por um único tipo, o tipo principal de M. A partir de um resultado sobre termos lineares demonstrado por Hindley

[2]Minimal no sentido de ter $\rightarrow$ como única conectiva.

em (Hindley, 1989) e uma vez que cada termo linear é, em particular, também afim, segue, como corolário, que a equivalência módulo-β no λ-calculus linear corresponde, a menos de variantes, à igualdade de tipos principais.

5.2 Preliminares

5.2.1 Atribuição de tipos à la Curry

Nesta subsecção descrevemos o λ-calculus simplesmente tipado à la Curry[3], começando por descrever o conjunto de tipos $\mathcal{T}$. Utilizaremos, daqui em diante, as expressões *λ-termo tipado* e *tipo* respectivamente como sinónimos de *λ-termo simplesmente tipado à la Curry* e *tipo em $\mathcal{T}$*.

Definição 5.2.1 *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 $\rightarrow$:*

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

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 $\rightarrow$ vamos considerar associatividade à direita abreviando

$$(\alpha_1 \rightarrow (\alpha_2 \rightarrow \ldots \rightarrow (\alpha_{n-1} \rightarrow \alpha_n) \ldots))$$

por

$$\alpha_1 \rightarrow \alpha_2 \rightarrow \ldots \rightarrow \alpha_n.$$

Definição 5.2.2 *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 finito, possivelmente vazio, de declarações que têm variáveis distintas como sujeitos chamamos uma base.*

[3] Para não sobrecarregar a notação iremos frequentemente identificar os diversos sistemas tipados de λ-calculus pelos respectivos conjuntos de termos tipados.

Definição 5.2.3 *Dizemos que um tipo $\alpha \in \mathcal{T}$ é inferível para um termo $M \in \Lambda$ 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:

$$\boxed{\begin{array}{rcl} x : \alpha \in \mathcal{B} & \Rightarrow & \mathcal{B} \vdash x : \alpha; \\ \mathcal{B} \vdash M : (\alpha \to \beta), \mathcal{B} \vdash N : \alpha & \Rightarrow & \mathcal{B} \vdash (MN) : \beta; \\ \mathcal{B}, x : \alpha \vdash M : \beta & \Rightarrow & \mathcal{B} \vdash (\lambda x.M) : \alpha \to \beta. \end{array}}$$

Na definição anterior $\mathcal{B}, x : \alpha$ representa o conjunto $\mathcal{B} \cup \{x : \alpha\}$. Analogamente escrevemos também $x_1 : \alpha_1, \ldots, x_n : \alpha_n \vdash M : \alpha$ em vez de $\{x_1 : \alpha_1, \ldots, x_n : \alpha_n\} \vdash M : \alpha$. Designamos o conjunto de declarações $M : \alpha$, tal que o tipo α é inferível para M a partir de uma base $\mathcal{B}$, por $\Lambda(\to)$ ou, ocasionalmente, por $\Lambda_{\mathcal{K}}(\to)$. Analogamente escrevemos $\Lambda_{\mathcal{L}}(\to)$, $\Lambda_{\mathcal{A}}(\to)$ ou $\Lambda_{\mathcal{I}}(\to)$ quando restringimos o conjunto $\Lambda(\to)$ respectivamente a termos lineares, afins ou pertencentes a $\Lambda_{\mathcal{I}}$. Os respectivos subconjuntos de termos fechados denotam-se por $\Lambda^0_{\mathcal{K}}(\to)$, $\Lambda^0_{\mathcal{L}}(\to)$, $\Lambda^0_{\mathcal{A}}(\to)$ e $\Lambda^0_{\mathcal{I}}(\to)$.

Exemplo 5.2.4 *O tipo $b \to a$ é inferível para o termo $(\lambda xy.xz)(\lambda u.u)$ a partir da base $\mathcal{B} = \{z : a\}$:*

$$\begin{array}{rcl} z : a, u : a & \vdash & u : a \\ z : a & \vdash & (\lambda u.u) : a \to a \\ x : a \to a, y : b, z : a & \vdash & x : a \to a \\ x : a \to a, y : b, z : a & \vdash & z : a \\ x : a \to a, y : b, z : a & \vdash & xz : a \\ x : a \to a, z : a & \vdash & (\lambda y.xz) : b \to a \\ z : a & \vdash & (\lambda xy.xz) : (a \to a) \to b \to a \\ z : a & \vdash & (\lambda xy.xz)(\lambda u.u) : b \to a \end{array}$$

Um termo M diz-se tipável, se existir uma base $\mathcal{B}$ e um tipo α tal que $\mathcal{B} \vdash M : \alpha$. Note-se, que nem todos os λ-termos são tipáveis, e.g. o termo $\lambda x.xx$ não admite tipo. No entanto, como foi sugerido por Bunder e Meyer em (Bunder & Meyer, 1985) e mais tarde mostrado por Hindley em (Hindley, 1989), todos os λ-termos afins e consequentemente também os λ-termos lineares são tipáveis. Inversamente dizemos que um par $(\mathcal{B}, \alpha)$ é um par admissível, se existir um termo M tal que $\mathcal{B} \vdash M : \alpha$. Quando $\mathcal{B} = \emptyset$, dizemos também que α é um tipo admissível. No que se segue enumeramos algumas propriedades importantes de λ-termos tipados que irão ser usadas no resto deste capítulo e que se encontram demonstrados em (Barendregt, 1992).

Teorema 5.2.5 *Todo o λ-termo tipável admite forma normal.*

Definição 5.2.6 *Seja $\mathcal{B} = \{x_1 : \alpha_1, \ldots, x_n : \alpha_n\}$ uma base. O conjunto*

$$dom(\mathcal{B}) = \{x_1, \ldots, x_n\}$$

diz-se o domínio de $\mathcal{B}$. Se $\mathcal{V}_0 \subseteq \mathcal{V}$ for um conjunto de variáveis, então define-se a base $\mathcal{B}$ restrita às variáveis em $\mathcal{V}_0$ por

$$\mathcal{B} \restriction \mathcal{V}_0 = \{x : \alpha \mid x \in \mathcal{V}_0 \text{ e } x : \alpha \in \mathcal{B}\}.$$

Proposição 5.2.7 *Seja $\mathcal{B}$ uma base, $\alpha \in \mathcal{T}$ e $M \in \Lambda$ tal que $\mathcal{B} \vdash M : \alpha$.*

1. *Se $\mathcal{B}'$ for outra base e $\mathcal{B} \subseteq \mathcal{B}'$, então $\mathcal{B}' \vdash M : \alpha$.*
2. *$FV(M) \subseteq dom(\mathcal{B})$.*
3. *$\mathcal{B} \restriction FV(M) \vdash M : \alpha$.*

Proposição 5.2.8 *Qualquer subtermo de um termo tipável é também tipável.*

Proposição 5.2.9 (Propriedade da redução do sujeito) *Sejam M e M' dois λ-termos tais que $M \rightarrow_\beta M'$. Então*

$$\mathcal{B} \vdash M : \alpha \quad \Rightarrow \quad \mathcal{B} \vdash M' : \alpha.$$

Note-se que o inverso deste resultado, que se chama a propriedade da expansão do sujeito, nem sempre é válido. De facto, se $\mathcal{B} \vdash M' : \alpha$, então M pode nem ser tipável e mesmo sendo tipável pode não ser possível inferir o tipo α para M a partir de $\mathcal{B}$.

Exemplo 5.2.10 *Considerem-se os λ-termos*

$$\begin{array}{rrcl} & M_1 & = & (\lambda xyzw.xz(yw))(\lambda uv.u) \\ e & M_2 & = & (\lambda u.uu)(\lambda v.v)(\lambda yzw.z). \end{array}$$

Tem-se

$$\begin{array}{rl} & M_1 \to_\beta \lambda yzw.z \\ & M_2 \to_\beta \lambda yzw.z \\ e & \vdash \lambda yzw.z : a \to b \to c \to b. \end{array}$$

No entanto, mostra-se que o termo M_1 admite um tipo, mas que o tipo $a \to b \to c \to b$ não é inferível para M_1. Mais ainda, que o termo M_2 não é tipável, uma vez que o seu subtermo $(\lambda u.uu)$ não o é, cf. proposição 5.2.8.

Lema 5.2.11

1. $\mathcal{B} \vdash x : \alpha \quad \Rightarrow \quad (x : \alpha) \in \mathcal{B}$.
2. $\mathcal{B} \vdash MN : \beta \quad \Rightarrow \quad \exists \alpha\ [\mathcal{B} \vdash M : (\alpha \to \beta)$ *e* $\mathcal{B} \vdash N : \alpha]$.
3. $\mathcal{B} \vdash \lambda x.M : \gamma \quad \Rightarrow \quad \exists \alpha, \beta\ [\mathcal{B}, x : \alpha \vdash M : \beta$ *e* $\gamma = (\alpha \to \beta)]$.

5.2.2 Sistemas do cálculo proposicional minimal

O λ-calculus simplesmente tipado (à la Curry) descrito na secção anterior, está fortemente relacionado com o cálculo proposicional minimal que denotamos por $P(\to)$ e passamos a descrever formalmente.

Definição 5.2.12 *O conjunto $\mathcal{F}$ de fórmulas de $P(\rightarrow)$ é indutivamente definido a partir de um conjunto de variáveis proposicionais $\mathcal{V}ar = \{a, b, c, \ldots\}$ e da conectiva $\rightarrow$:*

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

É fácil ver que o conjunto de fórmulas de $P(\rightarrow)$ é precisamente o conjunto de tipos $\mathcal{T}$ do λ-calculus tipado, quando interpretamos variáveis proposicionais como variáveis para tipos. A seguir definimos o conceito de substituição no conjunto $\mathcal{F}$ de fórmulas de $P(\rightarrow)$ que ficará então igualmente definido para o conjunto de tipos $\mathcal{T}$, através da identificação referida dos dois conjuntos.

Definição 5.2.13 *Uma substituição é uma operação*

$$\star : \mathcal{V}ar \longrightarrow \mathcal{F}$$

que se generaliza naturalmente para o conjunto de fórmulas por

$$\star(\alpha \rightarrow \beta) = \star(\alpha) \rightarrow \star(\beta).$$

Notação 5.2.14 *Se uma substituição $\star$ tiver um suporte finito, i.e. se existir um número finito de variáveis $a_1, \ldots, a_n \in \mathcal{V}ar$ tal que*

$$\forall a \in \mathcal{V}ar \setminus \{a_1, \ldots, a_n\} \quad \star(a) = a,$$

então escrevemos $\star$ como

$$\star = [a_1 := \star(a_1), \ldots, a_n := \star(a_n)]$$

e denotamos $\star(\alpha)$ por $\alpha[a_1 := \star(a_1), \ldots, a_n := \star(a_n)]$. Por vezes usamos também a notação $\alpha^\star$ em vez de $\star(\alpha)$.

Definição 5.2.15 *Uma fórmula $\alpha \in \mathcal{F}$ é um teorema de $P(\to)$ se for possível obter α aplicando um número finito de vezes as regras seguintes:*

A1 *as fórmulas*

$$(a \to b \to c) \to (a \to b) \to a \to c$$
$$e \qquad a \to b \to a$$

são teoremas;

A2 *se $\alpha \to \beta$ e α forem teoremas, então β é um teorema;*

A3 *se α for um teorema e $\star$ uma substituição, então $\alpha^\star$ é um teorema.*

As duas regras de inferência **A2** e **A3** chamam-se respectivamente a regra do modus ponens e a regra da substituição, enquanto que as fórmulas em **A1** se designam por axiomas. Uma sucessão finita de fórmulas $\alpha_1, \ldots, \alpha_n$, onde cada α_i, para $1 \leq i \leq n$, é um axioma ou foi obtido de (dois) anteriores por aplicação da regra de inferência modus ponens ou substituição, diz-se uma dedução do teorema α_n.

Note-se que existem diversas maneiras alternativas de definir o conjunto dos teoremas de $P(\to)$, em particular sem fazer uso da regra de substituição, sem usar axiomas e também escolhendo conjuntos de axiomas diferentes, mas obviamente equivalentes no sentido de permitirem deduzir exactamente os mesmos teoremas, o que nos leva à definição seguinte.

Definição 5.2.16 *Um conjunto de fórmulas $A \subseteq \mathcal{F}$ gera o conjunto $B \subseteq \mathcal{F}$ se e só se B for o menor conjunto de fórmulas contendo A e fechado para as regras de inferência do modus ponens e da substituição. Dizemos que A é um conjunto completo de axiomas para um cálculo proposicional $P_{\mathcal{X}}(\to)$ quando A gera o conjunto dos teoremas de $P_{\mathcal{X}}(\to)$.*

Escolhendo ainda entre as fórmulas em $\mathcal{F}$, outros conjuntos de axiomas, obtêm-se diversos cálculos proposicionais, dos quais apresentamos três que nos interessam particularmente, pelo facto dos respectivos conjuntos de teoremas corresponderem exactamente aos tipos inferíveis para os λ-termos fechados lineares, afins ou pertencentes a $\lambda\mathcal{I}$.

A lógica cujo fragmento proposicional minimal, actualmente designado por $R_{\rightarrow}$, corresponde à versão tipada de $\lambda\mathcal{I}$ é a *lógica de relevância* descrita em (Anderson & Belnap, 1975). Teve origem no *cálculo proposicional implicativo positivo fraco* introduzido por Church em (Church, 1951) e foi, entre outros, estudado em (Moh, 1950), (Helman, 1977) e (Meredith, 1978). O conjunto de teoremas deste cálculo, que designamos por $P_{\mathcal{I}}(\rightarrow)$, obtém-se a partir dos axiomas

$$
\begin{aligned}
&a \rightarrow a\\
&(b \rightarrow c) \rightarrow (a \rightarrow b) \rightarrow a \rightarrow c\\
&(a \rightarrow b \rightarrow c) \rightarrow b \rightarrow a \rightarrow c\\
&(a \rightarrow b \rightarrow c) \rightarrow (a \rightarrow b) \rightarrow a \rightarrow c.
\end{aligned}
$$

Tomando por sua vez os conjuntos de axiomas

$$
\begin{aligned}
&a \rightarrow b \rightarrow a\\
&(b \rightarrow c) \rightarrow (a \rightarrow b) \rightarrow a \rightarrow c\\
&(a \rightarrow b \rightarrow c) \rightarrow b \rightarrow a \rightarrow c
\end{aligned}
$$

e

$$
\begin{aligned}
&a \rightarrow a\\
&(b \rightarrow c) \rightarrow (a \rightarrow b) \rightarrow a \rightarrow c\\
&(a \rightarrow b \rightarrow c) \rightarrow b \rightarrow a \rightarrow c.
\end{aligned}
$$

obtemos respectivamente subsistemas da *lógica afim* e da *lógica linear*, cf. (Kopylov, 1995), (Troelstra, 1992), (Girard, 1987) e (Wadler, 1993). Em (Fitch, 1936), (Meredith & Prior, 1963), (Meredith, 1978), (Bunder & Meyer, 1985), (Ono & Komori, 1985) e (Komori, 1986) encontram-se alguns dos estudos destes dois cálculos, que são frequentemente designados por *cálculo-BCK* e *cálculo-BCI* respectivamente. Para facilitar a leitura, utilizaremos aqui a notação $P_{\mathcal{A}}(\rightarrow)$ e $P_{\mathcal{L}}(\rightarrow)$, pois correspondem respectivamente às versões tipadas do λ-calculus afim e do λ-calculus linear. Igualmente usaremos a notação $P_{\mathcal{K}}(\rightarrow)$ em vez de $P(\rightarrow)$ para evidenciar a correspondência com o sistema $\lambda\mathcal{K}$.

Relativamente à relação existente entre os quatro (sub)-sistemas de λ-calculus e os quatro cálculos proposicionais apresentados, referimos somente um resultado particular, essencial para o estudo efectuado neste capítulo.

Proposição 5.2.17 *Seja $\mathcal{X} \in \{\mathcal{K}, \mathcal{I}, \mathcal{L}, \mathcal{A}\}$. Uma fórmula $\alpha \in \mathcal{F}$ é um teorema de $P_{\mathcal{X}}(\to)$ se e só se existir um termo fechado $M \in \Lambda^0_{\mathcal{X}}$ tal que $\vdash M : \alpha$, para α interpretado como tipo em $\mathcal{T}$.*

Chamamos atenção para o facto dos conjuntos de axiomas, apresentados para os quatro (sub)-cálculos de $P(\to)$, serem respectivamente constituídos por tipos inferíveis para elementos de bases de combinadores standard dos respectivos sistemas não tipados de λ-calculus. A relação entre conjuntos de axiomas e bases do sistema correspondente de λ-calculus (não) tipado será estudada nas duas secções seguintes.

Exemplo 5.2.18 *Os tipos*

$$\begin{array}{ll} & (a \to b \to c) \to (a \to b) \to a \to c \\ e & a \to b \to a \end{array}$$

são respectivamente inferíveis para os combinadores $\mathbf{S} = \lambda xyz.xz(yz)$ e $\mathbf{K} = \lambda xy.x$, que juntamente formam uma base do sistema $\lambda\mathcal{K}$. Com efeito, tem-se

$$\begin{array}{rcl}
x : a \to b \to c, y : a \to b, z : a & \vdash & z : a \\
x : a \to b \to c, y : a \to b, z : a & \vdash & x : a \to b \to c \\
x : a \to b \to c, y : a \to b, z : a & \vdash & xz : b \to c \\
x : a \to b \to c, y : a \to b, z : a & \vdash & y : a \to b \\
x : a \to b \to c, y : a \to b, z : a & \vdash & yz : b \\
x : a \to b \to c, y : a \to b, z : a & \vdash & xz(yz) : c \\
x : a \to b \to c, y : a \to b & \vdash & \lambda z.xz(yz) : a \to c \\
x : a \to b \to c & \vdash & \lambda yz.xz(yz) : (a \to b) \to a \to c \\
& \vdash & \lambda xyz.xz(yz) : (a \to b \to c) \to (a \to b) \to a \to c
\end{array}$$

e

$$\begin{array}{rcl}
x : a, y : b & \vdash & x : a \\
x : a & \vdash & \lambda y.x : b \to a \\
& \vdash & \lambda xy.x : a \to b \to a.
\end{array}$$

5.3 Bases para sistemas tipados de λ-calculus

O conceito de base pode ser generalizado para sistemas tipados de λ-calculus:

Definição 5.3.1 *Considere-se um conjunto de combinadores tipados $\Sigma(\to)$. O conjunto $\Sigma^+(\to)$ dos termos gerados por $\Sigma(\to)$ é o menor conjunto satisfazendo as condições seguintes:*

1. $\Sigma(\to) \subseteq \Sigma^+(\to)$;
2. *se* $M : \alpha \to \beta \ \in \Sigma^+(\to)$ *e* $N : \alpha \ \in \Sigma^+(\to)$, *então* $MN : \beta \ \in \Sigma^+(\to)$;
3. *se* $M : \alpha \ \in \Sigma^+(\to)$, *então* $M : \alpha^\star \ \in \Sigma^+(\to)$.

Um termo tipado $M : \alpha \in \Lambda^0(\to)$ diz-se definível a partir de $\Sigma(\to)$, se existir $N : \alpha \in \Sigma^+(\to)$ tal que $M =_\beta N$.

Definição 5.3.2 *Seja $\Sigma(\to) \subseteq \Lambda^0(\to)$ um conjunto de combinadores tipados. $\Sigma(\to)$ é uma base para um conjunto $\Gamma \subseteq \Lambda^0(\to)$ se*

$$\forall\ M : \alpha \in \Gamma \ \ \exists\ N : \alpha \in \Sigma^+(\to) \quad N =_\beta M,$$

i.e. todo o termo em Γ é definível a partir de $\Sigma(\to)$.

O conjunto $\Sigma(\to) \subseteq \Lambda^0_{\mathcal{X}}(\to)$ diz-se combinatoriamente completo, ou uma base, para o sistema tipado $\Lambda_{\mathcal{X}}(\to)$, com $\mathcal{X} \in \{\mathcal{K}, \mathcal{I}, \mathcal{L}, \mathcal{A}\}$, se e só se for uma base para $\Lambda^0_{\mathcal{X}}(\to)$.

Hindley provou em (Hindley, 1969) que o conjunto

$$\{\mathbf{S} : (a \to b \to c) \to (a \to b) \to a \to c\ ,\ \mathbf{K} : a \to b \to a\}$$

é uma base para o $\lambda\mathcal{K}$-calculus tipado.

Igualmente mostra-se, que os conjuntos

$$\begin{array}{rl} & \{\mathbf{B} : \beta, \mathbf{C} : \gamma, \mathbf{I} : \iota, \mathbf{S} : \sigma\} \\ & \{\mathbf{B} : \beta, \mathbf{C} : \gamma, \mathbf{I} : \iota\} \\ \text{e} & \{\mathbf{B} : \beta, \mathbf{C} : \gamma, \mathbf{K} : \kappa\} \end{array}$$

onde

$$\begin{array}{rcl} \beta & = & (b \to c) \to (a \to b) \to a \to c \\ \gamma & = & (a \to b \to c) \to b \to a \to c \\ \iota & = & a \to a \\ \sigma & = & (a \to b \to c) \to (a \to b) \to a \to c \\ \text{e} \quad \kappa & = & a \to b \to a \end{array}$$

geram respectivamente os sistemas tipados $\Lambda_{\mathcal{I}}(\to)$, $\Lambda_{\mathcal{L}}(\to)$ e $\Lambda_{\mathcal{A}}(\to)$.

Já foi referido por Statman em (Statman, 1986) que um conjunto de combinadores tipados $\Sigma(\to)$ é uma base para $\Lambda(\to)$ se e só se os tipos atribuídos aos elementos de $\Sigma(\to)$ e interpretados como fórmulas, formarem um conjunto completo de axiomas para o cálculo $P(\to)$. Veremos que esta propriedade é também válida para os sistemas tipados $\Lambda_{\mathcal{I}}(\to)$, $\Lambda_{\mathcal{L}}(\to)$ e $\Lambda_{\mathcal{A}}(\to)$, respectivamente em relação a $P_{\mathcal{I}}(\to)$,$P_{\mathcal{L}}(\to)$ e $P_{\mathcal{A}}(\to)$.

Notação 5.3.3 *Seja $\mathcal{B} = \{x_1 : \gamma_1, \ldots, x_n : \gamma_n\}$ uma base e $\star$ uma substituição. Denotamos por $\mathcal{B}^\star$ o conjunto*

$$\mathcal{B}^\star = \{x_1 : \gamma_1^\star, \ldots, x_n : \gamma_n^\star\}.$$

Analogamente escrevemos $\mathcal{B}[a_1 := \alpha_1, \ldots, a_m := \alpha_m]$ em vez de

$$\{x_1 : \gamma_1[a_1 := \alpha_1, \ldots, a_m := \alpha_m], \ldots, x_n : \gamma_n[a_1 := \alpha_1, \ldots, a_m := \alpha_m]\}.$$

O resultado seguinte encontra-se demonstrado em (Barendregt, 1992).

Proposição 5.3.4

1. *Se $\mathcal{B} \vdash M : \alpha$, então $\mathcal{B}[a := \beta] \vdash M : \alpha[a := \beta]$.*

2. *Se $\mathcal{B}, x : \alpha \vdash M : \beta$ e $\mathcal{B} \vdash N : \alpha$, então $\mathcal{B} \vdash M[N/x] : \beta$.*

Lema 5.3.5 *Seja $\alpha \in \mathcal{T}$ um dos tipos*

$$\begin{array}{rcl} \iota & = & a \to a \\ \kappa & = & a \to b \to a \\ \beta & = & (b \to c) \to (a \to b) \to a \to c \\ \gamma & = & (a \to b \to c) \to b \to a \to c \\ ou \quad \sigma & = & (a \to b \to c) \to (a \to b) \to a \to c. \end{array}$$

Então existe um e um só λ-termo M em forma normal tal que $\vdash M : \alpha$.

Prova Faremos a prova para o tipo

$$\sigma = (a \to b \to c) \to (a \to b) \to a \to c.$$

Pela proposição 5.2.7 podemos afirmar que qualquer termo M para qual é inferível

$$\vdash M : (a \to b \to c) \to (a \to b) \to a \to c$$

é necessariamente um termo fechado. Do lema 2.1.5 resulta então que M é da forma

$$M = \lambda x_1 \ldots x_n . y N_1 \ldots N_m$$

com $n \geq 1$, $m \geq 0$, $\{y\} \cup FV(N_1) \cup \ldots \cup FV(N_m) \subseteq \{x_1, \ldots, x_n\}$ e tal que os subtermos $N_1, \ldots, N_m$ estão também em forma normal. Aplicando o lema 5.2.11 obtemos

$$x_1 : a \to b \to c \vdash \lambda x_2 \ldots x_n . y N_1 \ldots N_m : (a \to b) \to a \to c.$$

Se $n = 1$, donde $y = x_1$, resulta

$$x_1 : a \to b \to c \vdash x_1 N_1 \ldots N_m : (a \to b) \to a \to c$$

o que é impossível, igualmente pelo lema 5.2.11. Assim temos $n \geq 2$ e

$$x_1 : a \to b \to c, x_2 : a \to b \vdash \lambda x_3 \ldots x_n . y N_1 \ldots N_m : a \to c.$$

Por considerações semelhantes às anteriores, excluímos a hipótese de $n = 2$ e para $n \geq 3$ obtemos

$$x_1 : a \to b \to c, x_2 : a \to b, x_3 : a \vdash \lambda x_4 \ldots x_n . y N_1 \ldots N_m : c.$$

Uma vez que o tipo de $\lambda x_4 \ldots x_n . y N_1 \ldots N_m$ não é da forma $\alpha \to \beta$ segue $n = 3$, donde

$$x_1 : a \to b \to c, x_2 : a \to b, x_3 : a \vdash y N_1 \ldots N_m : c$$

com $y \in \{x_1, x_2, x_3\}$. Do lema 5.2.11 segue $y = x_1$ e $m = 2$ com

$$x_1 : a \to b \to c, x_2 : a \to b, x_3 : a \vdash N_1 : a$$

e

$$x_1 : a \to b \to c, x_2 : a \to b, x_3 : a \vdash N_2 : b.$$

Analisando estes dois casos de forma análoga e relembrando que os termos N_1 e N_2 estão também em forma normal, concluímos finalmente que o termo

$$M = \lambda x_1 x_2 x_3 . x_1 x_3 (x_2 x_3)$$

é o único termo para o qual é possível inferir o tipo $(a \to b \to c) \to (a \to b) \to a \to c$. •

Proposição 5.3.6 *Seja $\mathcal{X} \in \{\mathcal{K}, \mathcal{I}, \mathcal{L}, \mathcal{A}\}$ e $\{M_1 : \alpha_1, \ldots, M_n : \alpha_n\} \subseteq \Lambda^0_{\mathcal{X}}(\to)$. O conjunto $\{M_1 : \alpha_1, \ldots, M_n : \alpha_n\}$ é uma base para $\Lambda_{\mathcal{X}}(\to)$ se e só se $\{\alpha_1, \ldots, \alpha_n\} \subseteq \mathcal{F}$ formar um conjunto completo de axiomas para $P_{\mathcal{X}}(\to)$.*

Prova Fazemos a demonstração somente para o sistema $\mathcal{X} = \mathcal{A}$. Os restantes casos são análogos.

A condição de $\{\alpha_1, \ldots, \alpha_n\}$ formar um conjunto completo de axiomas para $P_{\mathcal{A}}(\to)$ é obviamente necessária uma vez que, para ser base de $\Lambda_{\mathcal{A}}(\to)$, o conjunto $\{M_1 : \alpha_1, \ldots, M_n : \alpha_n\}$ terá que gerar os combinadores tipados

$$\begin{aligned} \lambda xyz.x(yz) &: (b \to c) \to (a \to b) \to a \to c \\ \lambda xyz.xzy &: (a \to b \to c) \to b \to a \to c \\ \lambda xy.x &: a \to b \to a \end{aligned}$$

somente por aplicação e substituição. Esquecendo os termos neste processo obtemos então deduções das fórmulas

$$\begin{aligned} \beta &= (b \to c) \to (a \to b) \to a \to c \\ \gamma &= (a \to b \to c) \to b \to a \to c \\ \kappa &= a \to b \to a \end{aligned}$$

a partir dos axiomas $\alpha_1, \ldots, \alpha_n$, onde que cada aplicação de termos se traduz numa aplicação da regra do modus ponens. Por outro lado, as três fórmulas formam um conjunto completo de axiomas para $P_{\mathcal{A}}(\to)$.

Suponhamos agora que $\{\alpha_1, \ldots, \alpha_n\}$ é um conjunto completo de axiomas para $P_{\mathcal{A}}(\rightarrow)$. Então é possível deduzir as fórmulas β, γ e κ a partir de $\alpha_1, \ldots, \alpha_n$ usando somente as regras do modus ponens e da substituição. Usando a aplicação dos termos correspondentes para cada aplicação da regra de modus ponens, podemos transformar estas deduções em inferências dos três tipos, respectivamente para termos U, V e W, combinações de $M_1, \ldots, M_n$. Pelo teorema 5.2.5 existem formas normais de U, V e W que, pela proposição 5.2.9, admitem respectivamente os tipos β, γ e κ. Donde, pelo lema 5.3.5, são necessariamente da forma $\lambda xyz.x(yz)$, $\lambda xyz.xzy$ e $\lambda xy.x$. •

Ronald Harrop mostrou para todo o subcálculo de $P(\rightarrow)$, baseado num conjunto finito de axiomas contendo as fórmulas

$$a \rightarrow b \rightarrow a \quad \text{e} \quad a \rightarrow a,$$

ser indecidível se um conjunto finito de fórmulas é ou não um conjunto completo de axiomas para esse cálculo, cf. (Harrop, 1964) e (Harrop, 1965). A partir deste resultado Statman concluiu, [(Statman, 1986), Teorema 5], a indecidibilidade de um conjunto finito de combinadores tipados $\Sigma(\rightarrow) \subseteq \Lambda^0(\rightarrow)$ ser uma base para $\Lambda(\rightarrow)$. Da proposição 5.3.6 e do facto de

$$\{(b \rightarrow c) \rightarrow (a \rightarrow b) \rightarrow a \rightarrow c,\ (a \rightarrow b \rightarrow c) \rightarrow b \rightarrow a \rightarrow c,\ a \rightarrow b \rightarrow a,\ a \rightarrow a\}$$

formar um conjunto de axiomas para $P_{\mathcal{A}}(\rightarrow)$, resulta o mesmo para $\Lambda_{\mathcal{A}}(\rightarrow)$:

Corolário 5.3.7 *Seja $\Sigma_{\mathcal{A}}(\rightarrow) \subseteq \Lambda^0_{\mathcal{A}}(\rightarrow)$ um conjunto finito de combinadores afins tipados. Então é indecidível se $\Sigma_{\mathcal{A}}(\rightarrow)$ é uma base para $\Lambda_{\mathcal{A}}(\rightarrow)$.*

5.4 Completude combinatória vs. completude axiomática

Nesta secção estudamos a relação entre a completude combinatória de um conjunto de combinadores tipáveis $\{M_1, \ldots, M_n\}$ para um dos sistemas $\Lambda_{\mathcal{X}}$, com $\mathcal{X} \in \{\mathcal{K}, \mathcal{I}, \mathcal{L}, \mathcal{A}\}$, e a completude axiomática para o cálculo $P_{\mathcal{X}}(\rightarrow)$ do conjunto de todos os tipos inferíveis para $M_1, \ldots, M_n$. Note-se que os tipos inferíveis para um λ-termo tipável M podem ser representados por um único tipo, o tipo principal de M, a partir do qual todos os outros se obtêm por substituição, cf. (Hindley, 1969) e (Curry, 1969).

Definição 5.4.1 *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.*

$$\forall \sigma \in \mathcal{T} \quad \star_1(\sigma) = \star_2(\star(\sigma)).$$

Se existirem duas substituições $\star_1$ e $\star_2$ tais que

$$\alpha^{\star_1} = \beta \qquad e \qquad \beta^{\star_2} = \alpha$$

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

$$(\mathcal{B}_1^{\star_1}, \alpha^{\star_1}) = (\mathcal{B}_2, \beta) \qquad e \qquad (\mathcal{B}_2^{\star_2}, \beta^{\star_2}) = (\mathcal{B}_1, \alpha)$$

para duas substituições $\star_1$ e $\star_2$.

Definição 5.4.2 *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 $(\mathcal{B}, \alpha)$. O resultado seguinte, provado independentemente por Curry em (Curry, 1969), Hindley em (Hindley,

1969) e Milner em (Milner, 1978), mostra 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) de qualquer termo tipável e que falha quando aplicado a um termo que não admite tipo.

Teorema 5.4.3 (Teorema do tipo principal)

Existe uma função recursiva pp tal que

1. *se M for um termo tipável, então $pp(M) = (\mathcal{B}, \alpha)$, onde $(\mathcal{B}, \alpha)$ é um par principal de M;*
2. *se M não admitir tipo, então $pp(M) = fail$.*

Note-se que a função *pp* retorna, quando aplicada a um termo fechado tipável M, um par da forma $(\emptyset, \alpha)$, onde α é o tipo principal de M. A partir do lema 5.2.11, mostra-se facilmente a seguinte propriedade que utilizaremos mais tarde.

Lema 5.4.4 *Sejam M e N dois λ-termos fechados tais que o termo MN é tipável. Se α_1 e α_2 forem respectivamente tipos principais de M e N, então existem substituições $\star_1$ e $\star_2$ tal que $\alpha_1^{\star_1} = \beta \to \gamma$ e $\alpha_2^{\star_2} = \beta$, onde γ é um tipo principal de MN.*

Voltando ao problema inicial posto nesta secção, consideremos então um conjunto finito de combinadores tipáveis $M_1, \ldots, M_n \in \Lambda^0_{\mathcal{X}}(\to)$, com $\mathcal{X} \in \{\mathcal{K}, \mathcal{I}, \mathcal{L}, \mathcal{A}\}$, respectivamente com os tipos principais $\alpha_1, \ldots, \alpha_n$. Da proposição 5.3.6 segue como corolário que a completude axiomática de $\{\alpha_1, \ldots, \alpha_n\}$ para $P_{\mathcal{X}}(\to)$ implica a completude combinatória de $\{M_1, \ldots, M_n\}$ para o sistema não tipado $\Lambda_{\mathcal{X}}$.

Corolário 5.4.5 *Seja $\{M_1, \ldots, M_n\} \subseteq \Lambda^0_{\mathcal{X}}$, com $\mathcal{X} \in \{\mathcal{K}, \mathcal{I}, \mathcal{L}, \mathcal{A}\}$, um conjunto de combinadores tipáveis respectivamente com tipos principais $\alpha_1, \ldots, \alpha_n$. Se o conjunto $\{\alpha_1, \ldots, \alpha_n\}$, interpretado como subconjunto de $\mathcal{F}$, formar um conjunto completo de axiomas para o cálculo $P_{\mathcal{X}}(\to)$, então $\{M_1, \ldots, M_n\}$ é uma base para $\Lambda_{\mathcal{X}}$.* •

A relação entre as duas propriedades no sentido inverso é menos clara, sendo em particular falsa para os sistemas $\lambda\mathcal{K}$ e $\lambda\mathcal{I}$. Começamos por referir um resultado que foi independentemente provado por Hindley em (Hindley, 1969) e por Curry em (Curry, 1969).

Teorema 5.4.6 *Seja $M' \in \Lambda^0$ e $\alpha \in \mathcal{T}$ tal que $\vdash M' : \alpha$. Então existe um termo $M \in \Lambda^0$ com $M \to_\beta M'$ e tal que α é um tipo principal de M.*

Proposição 5.4.7 *A completude combinatória de um conjunto de combinadores tipáveis para o $\lambda\mathcal{K}$-calculus, não implica a completude axiomática do conjunto dos respectivos tipos principais, interpretados como fórmulas em $\mathcal{F}$, para o cálculo $P_{\mathcal{K}}(\to)$.*

Prova Consideremos os λ-termos

$$\begin{array}{rrcl} & \mathbf{K} & = & \lambda xy.x \\ \text{e} & \mathbf{S} & = & \lambda xyz.xz(yz) \end{array}$$

para os quais são respectivamente inferíveis os tipos

$$\begin{array}{rrcl} & \tilde{\kappa} & = & a \to a \to a \\ \text{e} & \tilde{\sigma} & = & (a \to a \to a) \to (a \to a) \to a \to a. \end{array}$$

Da proposição 5.4.6 segue a existência de termos $\tilde{\mathbf{K}}$ e $\tilde{\mathbf{S}}$, respectivamente com tipos principais $\tilde{\kappa}$ e $\tilde{\sigma}$, e tal que $\tilde{\mathbf{K}} \to_\beta \mathbf{K}$ e $\tilde{\mathbf{S}} \to_\beta \mathbf{S}$. De $\{\mathbf{S}, \mathbf{K}\}$ ser base para $\lambda\mathcal{K}$, resulta a completude combinatória do conjunto $\{\tilde{\mathbf{S}}, \tilde{\mathbf{K}}\}$. Por outro lado, é fácil ver que o conjunto dos tipos principais de $\tilde{\mathbf{K}}$ e $\tilde{\mathbf{S}}$, i.e. o conjunto $\{\tilde{\kappa}, \tilde{\sigma}\}$, não forma um conjunto completo de axiomas para $P_{\mathcal{K}}(\to)$, uma vez que não é possível deduzir a fórmula $a \to b \to a$ a partir de $\{\tilde{\kappa}, \tilde{\sigma}\}$. •

No resultado anterior não fizemos nenhuma restrição acerca da forma dos combinadores. Mostramos para o caso do sistema $\lambda\mathcal{I}$ que o resultado se verifica mesmo quando nos restringimos a formas normais ou até a combinadores próprios. Durante a prova ficará claro que o mesmo é válido para $\lambda\mathcal{K}$.

Proposição 5.4.8 *A completude combinatória de um conjunto de combinadores próprios tipáveis para o $\lambda\mathcal{I}$-calculus, não implica a completude axiomática do conjunto dos respectivos tipos principais, interpretados como fórmulas em $\mathcal{F}$, para o cálculo $P_{\mathcal{I}}(\to)$.*

Prova Consideremos os combinadores próprios

$$\begin{array}{rcl} \tilde{\mathbf{W}} & = & \lambda mxy.m(m(x(my)(my))) \\ \tilde{\mathbf{B}} & = & \lambda mxyz.m(x(m(m(y(mz))))) \\ \tilde{\mathbf{C}} & = & \lambda mxyz.m(m(x(mz)(my))) \\ \text{e } \mathbf{I} & = & \lambda x.x \end{array}$$

respectivamente com tipos principais

$$\begin{array}{l} (a \to a) \to (a \to a \to a) \to a \to a \\ (a \to a) \to (a \to a) \to (a \to a) \to a \to a \\ (a \to a) \to (a \to a \to a) \to a \to a \to a \\ \text{e} \quad a \to a. \end{array}$$

É claro que o conjunto dos tipos principais não forma um conjunto completo de axiomas para o cálculo $P_{\mathcal{I}}(\to)$ uma vez que, em particular, não permite deduzir o teorema

$$(a \to a \to b) \to a \to b.$$

Por outro lado, tem-se

$$\begin{array}{rcccl} \tilde{\mathbf{W}}\mathbf{I} & \to_\beta & \mathbf{W} & = & \lambda xy.xyy \\ \tilde{\mathbf{B}}\mathbf{I} & \to_\beta & \mathbf{B} & = & \lambda xyz.x(yz) \\ \text{e } \tilde{\mathbf{C}}\mathbf{I} & \to_\beta & \mathbf{C} & = & \lambda xyz.xzy \end{array}$$

donde $\{\tilde{\mathbf{W}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}}, \mathbf{I}\}$ é base de $\lambda\mathcal{I}$ uma vez que $\{\mathbf{W}, \mathbf{B}, \mathbf{C}, \mathbf{I}\}$ o é. •

Note-se que a prova da proposição 5.3.6, do qual resulta o corolário 5.4.5, se baseia no facto da propriedade da redução do sujeito ser válida no λ-calculus. Mostraremos agora que obtemos o resultado inverso ao corolário 5.4.5 para sistemas onde é válida a propriedade da expansão do sujeito.

Definição 5.4.9 *Dizemos que num sistema de λ-calculus $\Lambda_{\mathcal{X}}$ é válida a propriedade da expansão do sujeito se e só se para $M, N \in \Lambda_{\mathcal{X}}$ e $\alpha \in \mathcal{T}$,*

$$\mathcal{B} \vdash N : \alpha \quad e \quad M \to_\beta N \quad \Rightarrow \quad \mathcal{B} \vdash M : \alpha.$$

Proposição 5.4.10 *Considere-se um sistema de λ-calculus $\Lambda_{\mathcal{X}} \subseteq \Lambda$ no qual é válida a propriedade da expansão do sujeito. Seja $\{M_1, \ldots, M_n\} \subseteq \Lambda^0_{\mathcal{X}}$ um conjunto de combinadores tipáveis, completo para $\Lambda_{\mathcal{X}}$ e sejam $\alpha_1, \ldots, \alpha_n \in \mathcal{T}$ os respectivos tipos principais de $M_1, \ldots, M_n$. Então o conjunto de fórmulas $\{\alpha_1, \ldots, \alpha_n\}$ forma um conjunto completo de axiomas para $P_{\mathcal{X}}(\rightarrow)$, onde $P_{\mathcal{X}}(\rightarrow)$ é o subcálculo proposicional de $P(\rightarrow)$, cujo conjunto de teoremas é o conjunto de tipos admissíveis para $\Lambda_{\mathcal{X}}$.*

Prova Seja α um teorema de $P_{\mathcal{X}}(\rightarrow)$. Então existe um termo $N \in \Lambda^0_{\mathcal{X}}$ em forma normal tal que $\vdash N : \alpha$. Da completude do conjunto $\{M_1, \ldots, M_n\}$ segue a existência de uma combinação M de $M_1, \ldots, M_n$ que define N. Consequentemente tem-se $M \rightarrow_\beta N$, donde $\vdash M : \alpha$. Pelo lema 5.4.4 é possível inferir $\vdash M : \gamma$, onde γ é um tipo principal de M, a partir de

$$\vdash M_1 : \alpha_1, \ldots, \vdash M_n : \alpha_n$$

utilizando substituições nos tipos quando necessário. Logo existe uma substituição $\star$ tal que $\alpha = \gamma^\star$. Esquecendo os termos neste processo de inferência obtemos uma dedução de γ a partir das fórmulas $\alpha_1, \ldots, \alpha_n$ da qual resulta uma dedução de α pela regra de substituição.

•

Do resultado seguinte, provado por Hindley em (Hindley, 1989), concluímos que no caso linear a completude combinatória implica a completude axiomática.

Teorema 5.4.11 (Hindley) *Sejam M e N dois λ-termos lineares tais que $M =_\beta N$. Então M e N têm os mesmos pares principais.*

Corolário 5.4.12 *Seja $\{M_1, \ldots, M_n\}$ uma base de combinadores lineares para o λ-calculus linear. Então o conjunto $\{\alpha_1, \ldots, \alpha_n\}$ dos tipos principais de $M_1, \ldots, M_n$ forma um conjunto completo de axiomas para o cálculo $P_{\mathcal{L}}(\rightarrow)$.*

Prova Do teorema 5.4.11 resulta que a propriedade da expansão do sujeito é válida no λ-calculus linear. Assim obtém-se o resultado pela proposição 5.4.10. •

No exemplo 5.2.10 vimos que a propriedade da expansão do sujeito não se verifica no λ-calculus afim. Mesmo assim, podemos obter um resultado análogo ao anterior para este sistema, uma vez que se podem estabelecer restrições sobre os tipos principais de λ-termos afins. Para isso necessitamos dos dois resultados seguintes, que foram demonstrados por Hindley em (Hindley, 1989).

Teorema 5.4.13 *Todo o λ-termo afim é tipável.*

Lema 5.4.14 *Sejam M e M' dois λ-termos afins, respectivamente da forma*

$$M = (\lambda x.U)VP_1 \ldots P_t \quad e \quad M' = (U[V/x])P_1 \ldots P_t \qquad (t \geq 0)$$

tais que $\mathcal{B} \vdash M' : \alpha$ para alguma base $\mathcal{B}$ e $\alpha \in \mathcal{T}$.

1. *Se $x \in FV(U)$, então $\mathcal{B} \vdash M : \alpha$.*

2. *Se $x \notin FV(U)$, mas existe $\beta \in \mathcal{T}$ tal que $\mathcal{B} \vdash V : \beta$, então $\mathcal{B} \vdash M : \alpha$.*

Lema 5.4.15 *Seja M um λ-termo afim com $FV(M) = \{x_1, \ldots, x_n\}$. Então existem tipos $\alpha_1, \ldots, \alpha_n, \alpha$ tais que*

$$x_1 : \alpha_1, \ldots, x_n : \alpha_n \vdash M : \alpha$$

e tal que cada $\beta \in \{\alpha_1, \ldots, \alpha_n\}$ é da forma $\beta_1 \to \ldots \to \beta_k \to b$ onde a variável b não ocorre nos esquemas $\beta_1, \ldots, \beta_k$.

Prova Procedemos por indução sobre o comprimento de M, pelo qual entendemos o número de todas as ocorrências de símbolos em M, incluindo variáveis, λ's, parenteses e símbolos de aplicação.

Para $M = x$ temos $x : a \vdash x : a$, onde a é uma variável para tipos.

Suponhamos agora $M = \lambda x.M_1$ e $x \notin FV(M_1)$. Por hipótese de indução existe uma base $\mathcal{B}$, satisfazendo a condição do lema, bem como um tipo α tal que $\mathcal{B} \vdash M_1 : \alpha$. Segue-se $\mathcal{B} \vdash \lambda x.M_1 : a \to \alpha$, onde a é uma variável para tipos nova.

Seja $M = \lambda x.M_1$ e $x \in FV(M_1)$. Então existem uma base $\mathcal{B}$ e tipos β e α tal que $\mathcal{B}, x : \beta \vdash M_1 : \alpha$, donde $\mathcal{B} \vdash \lambda x.M_1 : \beta \to \alpha$. Por hipótese de indução é óbvio que $\mathcal{B}$ está nas condições requeridas.

Seja $M = (\lambda x.U)VP_1 \ldots P_t$, com $t \geq 0$ e $x \notin FV(U)$. Pelo teorema 5.4.13 e por hipótese de indução existem bases $\mathcal{B}_1$ e $\mathcal{B}_2$ nas condições do lema tal que

$$\mathcal{B}_1 \vdash UP_1 \ldots P_t : \alpha$$

e

$$\mathcal{B}_2 \vdash V : \beta.$$

Pela proposição 5.2.7 e uma vez que M é um termo afim, podemos supor que $dom(\mathcal{B}_1) \cap dom(\mathcal{B}_2) = \emptyset$, e também que $x \notin dom(\mathcal{B}_1) \cup dom(\mathcal{B}_2)$. Segue

$$\mathcal{B}_1, \mathcal{B}_2 \vdash V : \beta,$$

igualmente da proposição 5.2.7. Por outro lado, concluímos da proposição 5.2.11, que existem tipos $\alpha_1, \ldots, \alpha_t$ tal que

$$\mathcal{B}_1 \vdash U : \alpha_1 \to \ldots \to \alpha_t \to \alpha$$

e

$$\mathcal{B}_1 \vdash P_i : \alpha_i \quad (1 \leq i \leq t),$$

donde

$$\mathcal{B}_1, \mathcal{B}_2, x : \beta \vdash U : \alpha_1 \to \ldots \to \alpha_t \to \alpha$$

e

$$\mathcal{B}_1, \mathcal{B}_2 \vdash P_i : \alpha_i \quad (1 \leq i \leq t).$$

Segue

$$\mathcal{B}_1, \mathcal{B}_2 \vdash \lambda x.U : \beta \to \alpha_1 \to \ldots \to \alpha_t \to \alpha$$

pelo que

$$\mathcal{B}_1, \mathcal{B}_2 \vdash (\lambda x.U)V : \alpha_1 \to \ldots \to \alpha_t \to \alpha$$

e finalmente

$$\mathcal{B}_1, \mathcal{B}_2 \vdash (\lambda x.U)VP_1 \ldots P_t : \alpha.$$

Para $M = (\lambda x.U)VP_1 \ldots P_t$, com $t \geq 0$ e $x \in FV(U)$, resulta do lema 5.4.14 que o termo $(\lambda x.U)VP_1 \ldots P_t$ admite exactamente os mesmos pares como $U[V/x]P_1 \ldots P_t$. Por outro lado, verifica-se facilmente que $U[V/x]P_1 \ldots P_t$ tem comprimento menor do que $(\lambda x.U)VP_1 \ldots P_t$, uma vez que a variável x ocorre exactamente uma vez em U, donde o resultado se verifica por hipótese de indução.

Finalmente suponhamos $M = xL_1 \ldots L_m$ com $m \geq 1$. Por hipótese de indução existem $\mathcal{B}_i$ e α_i tal que

$$\mathcal{B}_i \vdash L_i : \alpha_i,$$

para $1 \leq i \leq m$. Note-se que também neste caso podemos supor que os domínios das bases $\mathcal{B}_1, \ldots, \mathcal{B}_m$ são dois a dois disjuntos e não contêm a variável x. Escolhendo uma variável para tipos nova a obtemos

$$\mathcal{B}_1, \ldots, \mathcal{B}_m, x : \alpha_1 \to \ldots \alpha_m \to a \vdash xL_1 \ldots L_m : a. \quad \bullet$$

Lema 5.4.16 *Seja M um λ-termo afim fechado, tal que a forma normal de M é um combinador próprio $\lambda x_1 \ldots x_n.N$ com tipo principal $\alpha = \alpha_1 \to \ldots \alpha_n \to a$. Então existem tipos $\tilde{\alpha}_1, \ldots, \tilde{\alpha}_n$ tal que*

- $\vdash M : \tilde{\alpha}_1 \to \ldots \to \tilde{\alpha}_n \to a$;
- *se $x_i \in FV(N)$, então $\tilde{\alpha}_i = \alpha_i$, para $1 \leq i \leq n$;*
- *se $x_j \notin FV(N)$, então $\tilde{\alpha}_j = \beta_1 \to \ldots \to \beta_k \to b$, com $k \geq 0$ e $1 \leq j \leq n$, e tal que a variável b não ocorre em $\beta_1, \ldots, \beta_k$ e o tipo $\tilde{\alpha}_j$ não tem variáveis para tipos em comum com nenhum tipo α_i tal que $x_i \in FV(N)$ e $1 \leq i \leq n$.*

Prova Começamos por notar que α_i é uma variável para tipos a_i, que não ocorre mais nenhuma vez em α, quando $x_i \notin FV(N)$ e $i \in \{1, \ldots, n\}$.

Iremos provar o resultado por indução sobre o número k de passos da redução em ordem normal de M para $\lambda x_1 \ldots x_n.N$, mostrando que um tipo de M pode ser obtido de α por substituição de algumas das variáveis para tipos a_i, para os quais $x_i \notin FV(N)$, mas tais que

x_i ocorre livre em algum subtermo de M, por tipos da forma $\beta_1 \to \ldots \to \beta_k \to b$ constituídos por variáveis novas e tais que b não ocorre em $\beta_1, \ldots, \beta_k$.[4]

O resultado verifica-se trivialmente para $k = 0$.

Consideramos então uma redução em ordem normal de M para $\lambda x_1 \ldots x_n.N$ de comprimento $k > 0$ e suponhamos que no primeiro passo M é reduzido para M_1 substituindo um subtermo em M da forma $(\lambda x.P)Q$ por $P[Q/x]$. Do lema 5.4.14 conclui-se facilmente que M admite os mesmos tipos como M_1 quando $x \in FV(P)$, donde obtemos o resultado por hipótese de indução.

Suponhamos então que $x \notin FV(P)$, i.e. $P[Q/x] = P$ e sejam $y_1, \ldots, y_m$ as variáveis livres de Q. Tratando-se de uma redução em ordem normal e uma vez que M reduz para o combinador próprio $\lambda x_1 \ldots x_n.N$, é fácil ver que M é necessariamente da forma

$$\lambda x_1 \ldots x_t.(X_1(X_2(...(X_l((\lambda x.P)Q) \ldots) \ldots) \ldots) \ldots)$$

com $0 \leq t \leq n$ e $l = 0$ quando $t < n$, e tal que os termos $X_1, \ldots, X_l$ são termos aplicativos sobre $\{x_1, \ldots, x_n\} \setminus \{y_1, \ldots, y_m\}$. Note-se ainda que $\{y_1, \ldots, y_m\} \subseteq \{x_1, \ldots, x_t\}$ por M ser um termo fechado. O termo

$$M_1 = \lambda x_1 \ldots x_t.(X_1(X_2(...(X_l P \ldots) \ldots) \ldots) \ldots)$$

admite, por hipótese de indução o tipo

$$\tilde{\alpha}_1 \to \ldots \to \tilde{\alpha}_n \to a$$

obtido a partir de α por substituição de algumas variáveis para tipos a_i por tipos constituídos por variáveis para tipos novas, que correspondem a variáveis x_i, com $1 \leq i \leq n$, tal que x_i ocorre livre em algum subtermo de M_1 enquanto que $x_i \notin FV(N)$. Para $x_i \in \{y_1, \ldots, y_m\}$ segue então $\tilde{\alpha}_i = \alpha_i = a_i$, que não ocorre mais nenhuma vez em $\tilde{\alpha}_1 \to \ldots \to \tilde{\alpha}_n \to a$. Do lema 5.4.15 conclui-se que existem tipos

$$\vec{\beta}_1 = \beta_{11} \to \ldots \to \beta_{1k_1} \to b_1 \;, \ldots ,\; \vec{\beta}_m = \beta_{m1} \to \ldots \to \beta_{mk_m} \to b_m \quad \text{e} \quad \gamma$$

[4]Note-se que para uma substituição $\star = [a_1 := \beta_{11} \to \ldots \to \beta_{1k_1} \to b_1, a_2 := \beta_{21} \to \ldots \to \beta_{2k_2} \to b_2]$ não excluímos a hipótese dos tipos $\beta_{11} \to \ldots \to \beta_{1k_1} \to b_1$ e $\beta_{21} \to \ldots \to \beta_{2k_2} \to b_2$ terem variáveis em comum, nem o facto da variável b_1 poder ocorrer em $\beta_{21} \to \ldots \to \beta_{2k_2} \to b_2$.

contendo variáveis para tipos que supomos diferentes das ocorrentes em $\tilde{\alpha}_1 \to \ldots \to \tilde{\alpha}_n \to a$, tal que b_i não ocorre em $\beta_{i1}, \ldots, \beta_{ik_i}$ para $1 \leq i \leq m$, e tal que

$$y_1 : \vec{\beta}_1, \ldots, y_m : \vec{\beta}_m \vdash Q : \gamma.$$

Consideramos agora o processo de inferência do tipo $\tilde{\alpha}_1 \to \ldots \to \tilde{\alpha}_n \to a$ para M_1. Como as variáveis $y_1, \ldots, y_m$ não ocorrem em $(X_1(X_2(...(X_l P \ldots) \ldots) \ldots) \ldots)$ resulta que todos os passos de inferência até ao tipo de $(X_1(X_2(...(X_l P \ldots) \ldots) \ldots) \ldots)$ são da forma

$$\mathcal{B}_U, y_1 : a_1, \ldots, y_m : a_m \vdash U : \sigma$$

sem que a atribuição de tipos $y_1 : a_1, \ldots, y_m : a_m$ seja usada alguma vez. Assim, é possível substituir cada a_i por $\vec{\beta}_i$, para $1 \leq i \leq m$. Na parte que corresponde à inferência de um tipo δ para P juntamos ainda a declaração $x : \gamma$ à base obtendo assim

$$\mathcal{B}_P, y_1 : \vec{\beta}_1, \ldots, y_m : \vec{\beta}_m, x : \gamma \vdash P : \delta$$

e inferimos

$$\mathcal{B}_P, y_1 : \vec{\beta}_1, \ldots, y_m : \vec{\beta}_m \vdash \lambda x.P : \gamma \to \delta.$$

Da proposição 5.2.7 decorre

$$\mathcal{B}_P, y_1 : \vec{\beta}_1, \ldots, y_m : \vec{\beta}_m \vdash Q : \gamma$$

donde

$$\mathcal{B}_P, y_1 : \vec{\beta}_1, \ldots, y_m : \vec{\beta}_m \vdash (\lambda x.P)Q : \delta.$$

É óbvio que a partir daqui inferimos o tipo $\tilde{\tilde{\alpha}}_1 \to \ldots \to \tilde{\tilde{\alpha}}_n \to a$ para M, onde $\tilde{\tilde{\alpha}}_i = \tilde{\alpha}_i$ quando $x_i \notin \{y_1, \ldots, y_m\}$. •

Teorema 5.4.17 *Seja $\{M_1, \ldots, M_n\}$ uma base de combinadores afins para o λ-calculus afim. Então o conjunto $\{\alpha_1, \ldots, \alpha_n\}$ dos tipos principais de $M_1, \ldots, M_n$ forma um conjunto completo de axiomas para o cálculo proposicional $P_{\mathcal{A}}(\to)$.*

Prova De $\{M_1, \ldots, M_n\}$ ser base, segue a existência de λ-termos fechados $\tilde{\mathbf{B}}, \tilde{\mathbf{C}}, \tilde{\mathbf{K}}$, construídos a partir de $M_1, \ldots, M_n$, tal que

$$\begin{array}{lll}\tilde{\mathbf{B}}xyz & \to_\beta & x(yz)\\ \tilde{\mathbf{C}}xyz & \to_\beta & xzy\\ \tilde{\mathbf{K}}xy & \to_\beta & x\end{array}$$

É óbvio que o mesmo se verifica para as formas normais de $\tilde{\mathbf{B}}, \tilde{\mathbf{C}}$ e $\tilde{\mathbf{K}}$, cuja existência decorre dos teoremas 5.4.13 e 5.2.5. Note-se ainda que os termos $\lambda xyz.x(yz)$, $\lambda xyz.xzy$ e $\lambda xy.x$ são as únicas formas normais fechadas nestas condições e que os seus tipos principais são respectivamente

$$\begin{array}{ll} & (b \to c) \to (a \to b) \to a \to c \\ & (a \to b \to c) \to b \to a \to c \\ \text{e} & a \to b \to a \end{array}$$

que por sua vez formam um conjunto completo de axiomas para o cálculo $P_{\mathcal{A}}(\to)$.

Os termos $\tilde{\mathbf{B}}, \tilde{\mathbf{C}}$ e $\tilde{\mathbf{K}}$ estão nas condições do lema 5.4.16, o que nos permite concluir

$$\begin{array}{l} \vdash \tilde{\mathbf{B}} : (b \to c) \to (a \to b) \to a \to c \\ \vdash \tilde{\mathbf{C}} : (a \to b \to c) \to b \to a \to c \\ \vdash \tilde{\mathbf{K}} : a \to \beta \to a \end{array}$$

com $\beta = \beta_1 \to \ldots \beta_k \to b$ e $k \geq 0$, tal que $a \neq b$ e nem a nem b ocorrem em $\beta_1, \ldots, \beta_k$.

Concluímos a prova, mostrando que um termo com tipo $a \to b \to a$ é definível a partir de $\tilde{\mathbf{B}}, \tilde{\mathbf{C}}$ e $\tilde{\mathbf{K}}$. Para isso construiremos um termo que admite um tipo da forma

$$a \to (\star(\beta_2) \to \ldots \to \star(\beta_k) \to b) \to a$$

onde $\star$ é uma substituição por tipos constituídos por variáveis novas. Consequentemente, a variável b não ocorre nos tipos $\star(\beta_2), \ldots, \star(\beta_k)$, enquanto que o número de implicações "principais" em $\star(\beta_2) \to \ldots \to \star(\beta_k) \to b$ é estritamente menor do que no tipo $\beta_1 \to \ldots \to \beta_k \to b$. Assim basta iterar este processo até obter um termo com um tipo da forma $a \to b \to a$.

Com efeito,

$$\begin{array}{l} \vdash \tilde{\mathbf{C}} : (((\beta \to a) \to c) \to (a \to (\beta \to a)) \to a \to c) \to (a \to (\beta \to a)) \to ((\beta \to a) \to c) \to a \to c \\ \vdash \tilde{\mathbf{B}} : ((\beta \to a) \to c) \to (a \to (\beta \to a)) \to a \to c \\ \vdash \tilde{\mathbf{C}}\tilde{\mathbf{B}} : (a \to (\beta \to a)) \to ((\beta \to a) \to c) \to a \to c \\ \vdash \tilde{\mathbf{K}} : a \to \beta \to a \\ \vdash \tilde{\mathbf{C}}\tilde{\mathbf{B}}\tilde{\mathbf{K}} : ((\beta \to a) \to c) \to a \to c \\ \vdash \tilde{\mathbf{C}}\tilde{\mathbf{B}}\tilde{\mathbf{K}} : ((\beta' \to a') \to c') \to a' \to c' \\ \vdash \tilde{\mathbf{C}}\tilde{\mathbf{B}}\tilde{\mathbf{K}}(\tilde{\mathbf{C}}\tilde{\mathbf{B}}\tilde{\mathbf{K}}) : \star_1(a \to c) \end{array}$$

tal que β' é β onde cada variável para tipos d é substituida por uma variável nova d' e tal que $\star_1$ é a substituição que resulta da unificação de

$$(\beta \to a) \to c \qquad \text{com} \qquad ((\beta' \to a') \to c') \to a' \to c'.$$

Note-se que esta unificação é possível pelo facto de qualquer λ-termo afim ser tipável (teorema 5.4.13). Seja então $\star_1$ a substituição que resulta das equações

$$\begin{aligned} & a = c', \\ & c = a' \to c', \\ & a' = \beta_2 \to \ldots \to \beta_k \to b \\ \text{e} \quad & \beta_1 = \beta_1' \to \ldots \to \beta_k' \to b'. \end{aligned}$$

Concluímos que

$$\star_1(a \to c) = c' \to \star(\beta_2 \to \ldots \to \beta_k \to b) \to c'$$

quando $\star$ resulta da unificação de β_1 com $\beta_1' \to \ldots \to \beta_k' \to b'$. Uma vez que a variável b não ocorre no esquema β_1 segue ainda

$$\vdash \tilde{\mathbf{C}}\tilde{\mathbf{B}}\tilde{\mathbf{K}}(\tilde{\mathbf{C}}\tilde{\mathbf{B}}\tilde{\mathbf{K}}) : c' \to (\beta_2^\star \to \ldots \to \beta_k^\star \to b) \to c'.$$

Como já referimos, repetimos este processo até obter uma combinação M sobre o conjunto $\{M_1, \ldots M_n\}$ tal que $\vdash M : a \to b \to a$. Do lema 5.4.4 decorre então que a fórmula $a \to b \to a$ pode ser obtida a partir do conjunto $\{\alpha_1, \ldots, \alpha_n\}$, utilizando as regras do modus ponens e da substituição . $\bullet$

Aplicação: no cálculo $P_{\mathcal{A}}(\to)$ afim é indecidível se um conjunto de tautologias forma um conjunto completo de axiomas. Mas esta propriedade é decidível, quando estas tautologias, interpretadas como tipos, corresponderem a tipos principais de λ-termos que se reduzem a combinadores próprios.

5.5 Tipos principais no λ-calculus afim

Acabamos este capítulo com a demonstração do resultado inverso do teorema 5.4.11 para formas normais no λ-calculus afim, que não se verifica no caso geral como mostra o exemplo seguinte:

Exemplo 5.5.1 *As formas normais distintas*

$$\lambda xyzu.x(yz)(yu) \quad e \quad \lambda xyzu.x(yu)(yz)$$

têm ambos o mesmo tipo principal

$$(a \to a \to b) \to (c \to a) \to c \to c \to b.$$

Note-se que o mesmo resultado já tinha sido demonstrado independentemente em (Hirokawa, 1993) e que apresentamos aqui uma prova alternativa, também exposta em (Broda, 1993), que nos parece ser bastante mais simples. Começamos por provar um resultado simples sobre pares principais de λ-termos afins em forma normal que será utilizado mais tarde.

Lema 5.5.2 *Seja M um λ-termo afim em forma normal com um par principal $(\mathcal{B}, \alpha)$. Então cada variável para tipos tem quando muito duas ocorrências em $(\mathcal{B}, \alpha)$.*

Prova Por indução estrutural sobre M.

Se $M = x$, então $(\mathcal{B}, \alpha)$ é da forma $(\{x : a\}, a)$.

Seja $M = \lambda x.P$, onde P é um λ-termo afim em forma normal. Do lema 5.2.11 resulta que α é da forma $\alpha = \beta \to \gamma$ e $\mathcal{B}, x : \beta \vdash P : \gamma$. Se $x \in FV(P)$, então $(\mathcal{B} \cup \{x : \beta\}, \gamma)$ é um par principal para P e o resultado verifica-se por hipótese de indução. Caso contrário, i.e. se $x \notin FV(P)$, então $(\mathcal{B}, \gamma)$ é um par principal para P e $\beta = b$ onde b é uma variável para tipos que não aparece mais nenhuma vez em $\mathcal{B}$ e γ. Pela hipótese de indução cada variável para tipos ocorre quando muito duas vezes em $(\mathcal{B}, \gamma)$ e consequentemente também em $(\mathcal{B}, b \to \gamma)$.

Suponhamos então que $M = xN_1 \ldots N_n$, com $n \geq 1$. Como M é um termo afim, é óbvio que a variável x não ocorre em $N_1, \ldots, N_n$, que $\alpha = a$ e que

$$\mathcal{B} = \mathcal{B}_1 \cup \ldots \cup \mathcal{B}_n \cup \{x : \alpha_1 \to \ldots \alpha_n \to a\}$$

tal que $(\mathcal{B}_i, \alpha_i)$ e $(\mathcal{B}_j, \alpha_j)$ não têm variáveis para tipos em comum, para $1 \leq i \neq j \leq n$, e que cada par $(\mathcal{B}_i, \alpha_i)$ é um par principal de N_i. Por hipótese de indução podemos então concluir que cada variável para tipos ocorre quando muito duas vezes em cada par $(\mathcal{B}_i, \alpha_i)$ e consequentemente também em $(\mathcal{B}, a)$. $\bullet$

Proposição 5.5.3 *Sejam M e N dois λ-termos afins em forma normal. Se M e N admitirem os mesmos pares principais, então $M =_\alpha N$.*

Prova Começamos por mostrar por indução sobre o comprimento de M e N, que se $\mathcal{B} = \{z_1 : \gamma_1, \ldots, z_k : \gamma_k\}$ for uma base tal que nenhuma variável para tipos ocorre mais do que duas vezes em $(\mathcal{B}, \alpha)$ e se $(\mathcal{B} \upharpoonright FV(M), \alpha)$ e $(\mathcal{B} \upharpoonright FV(N), \alpha)$ forem respectivamente pares principais de M e N, então tem-se $M =_\alpha N$. Note-se que isto é suficiente, uma vez que qualquer par principal de M e N satisfaz as condições da hipótese de indução pelos lemas 5.2.7 e 5.5.2.

O resultado é trivial quando M e N são variáveis. Caso contrário, resulta do lema 2.1.5 que os termos M e N são da forma

$$\begin{array}{rl} & M = \lambda u_1 \ldots u_{n_1}.y_1 M_1 \ldots M_{m_1} \\ \text{e} & N = \lambda v_1 \ldots v_{n_2}.y_2 N_1 \ldots N_{m_2} \end{array}$$

onde $n_1 > 0$, $n_2 > 0$, $m_1 > 0$ ou $m_2 > 0$ e que admitem os pares principais

$$\begin{array}{rl} & (\mathcal{B}_1, \beta_1^1 \to \ldots \beta_{n_1}^1 \to b_1) \\ \text{e} & (\mathcal{B}_2, \beta_1^2 \to \ldots \beta_{n_2}^2 \to b_2) \end{array}$$

respectivamente variantes de $(\mathcal{B} \upharpoonright FV(M), \alpha)$ e $(\mathcal{B} \upharpoonright FV(N), \alpha)$, donde resulta que α é da forma

$$\alpha = \alpha_1 \to \ldots \to \alpha_n \to a$$

onde $n = n_1 = n_2 \geq 0$.

A seguir substituímos todas as ocorrências de $u_1, \ldots, u_n$ em M e de $v_1, \ldots, v_n$ em N por variáveis novas $x_1, \ldots, x_n$ obtendo $M' =_\alpha M$ e $N' =_\alpha N$. Logo

$$(\mathcal{B} \upharpoonright FV(M'), \alpha) \quad \text{e} \quad (\mathcal{B} \upharpoonright FV(N'), \alpha)$$

são respectivamente pares principais de

$$\lambda x_1 \ldots x_n.y_1' M_1' \ldots M_{m_1}' \quad \text{e} \quad \lambda x_1 \ldots x_n.y_2' N_1' \ldots N_{m_2}'$$

e consequentemente

$$\begin{array}{rl} & (\mathcal{B} \cup \{x_1 : \alpha_1, \ldots, x_n : \alpha_n\} \upharpoonright FV(y_1' M_1' \ldots M_{m_1}'), a) \\ \text{e} & (\mathcal{B} \cup \{x_1 : \alpha_1, \ldots, x_n : \alpha_n\} \upharpoonright FV(y_2' N_1' \ldots N_{m_2}'), a) \end{array}$$

são respectivamente pares principais de

$$y'_1 M'_1 \dots M'_{m_1} \quad \text{e} \quad y'_2 N'_1 \dots N'_{m_2}$$

onde a variável para tipos a ocorre nos tipos de ambos

$$y'_1, y'_2 \in dom(\mathcal{B}) \cup \{x_1, \dots, x_n\}.$$

Como qualquer variável para tipos ocorre quando muito duas vezes em $\{\gamma_1, \dots, \gamma_k, \alpha\}$, resulta $y'_1 = y'_2 = y$, com

$$y : \beta \in \mathcal{B} \cup \{x_1 : \alpha_1, \dots, x_n : \alpha_n\}.$$

Note-se ainda que a variável y não ocorre nos termos $M'_1, \dots, M'_{m_1}, N'_1, \dots, N'_{m_2}$. Por outro lado, tem-se $FV(M'_i) \cap FV(M'_j) = \emptyset$, para $1 \leq i \neq j \leq m_1$, bem como $FV(N'_l) \cap FV(N'_k) = \emptyset$, para $1 \leq l \neq k \leq m_2$. Logo,

$$\beta = \beta_1 \to \dots \to \beta_m \to a$$

com $m = m_1 = m_2$ e para $i = 1, \dots, m$ os pares

$$\begin{array}{ll} & (\mathcal{B} \cup \{x_1 : \alpha_1, \dots, x_n : \alpha_n\} \restriction FV(M'_i), \beta_i) \\ \text{e} & (\mathcal{B} \cup \{x_1 : \alpha_1, \dots, x_n : \alpha_n\} \restriction FV(N'_i), \beta_i) \end{array}$$

são respectivamente pares principais de M'_i e N'_i. Então $M'_i =_\alpha N'_i$ resulta da hipótese de indução. •

Deste resultado segue, que no λ-calculus linear existe uma correspondência biunívoca entre pares principais e conjuntos de λ-termos que são equivalentes módulo-β, como também já tinha sido notado em (Hirokawa, 1993).

Corolário 5.5.4 *Sejam M e N dois λ-termos lineares. Então tem-se*

$$M =_\beta N \quad \Leftrightarrow \quad M \text{ e } N \text{ têm os mesmos pares principais.}$$

Capítulo 6

O sistema $\mathcal{C}\ell(\mathcal{K})$

Implementações de sistemas de λ-calculus são utilizadas em diversas áreas, entre as quais se destaca a programação funcional, que se baseia no facto de qualquer função recursiva poder ser definida por um λ-termo. Mas têm também aplicação noutros campos, como por exemplo em teorias para representação de conhecimento, (Cardelli, 1984), e em particular em alguns sistemas para processamento de linguagem natural, onde os valores semânticos de expressões sintácticas se representam por λ-termos, cf. (Dowty *et al.*, 1985) e (van Benthem, 1991). As dificuldades principais na implementação de um sistema de λ-calculus, devem-se à presença de variáveis ligadas nos λ-termos e surgem tanto na compilação como durante a redução dos termos. Estes problemas podem ser evitados utilizando a correspondência com um sistema de lógica combinatória equivalente, onde, como já referimos por diversas vezes, não há ocorrências de variáveis ligadas. Os sistemas combinatórios são de facto muito mais simples de implementar do que o λ-calculus, devido à redução de termos combinatórios ser extremamente simples. Mas este método só pode ter sucesso, se a representação dos λ-termos no sistema utilizado não for muito complexa, como acontece na maior parte dos casos. Os termos combinatórios, para os quais os λ-termos são traduzidos devem, em particular, conter um número pequeno de combinadores primitivos para reduzir o tempo e o espaço de memória temporário utilizados durante sua redução.

Introduzimos neste capítulo um sistema combinatório, com um conjunto infinito e numerável de constantes primitivas $\mathcal{K}$, e que denotamos por $\mathcal{C}\ell(\mathcal{K})$. Apresentamos também um algoritmo de abstracção correspondente, i.e. um algoritmo que permite simular a abstracção em $\mathcal{C}\ell(\mathcal{K})$

e a partir do qual definimos a tradução de λ-termos para $\mathcal{C}\ell(\mathcal{K})$-termos da maneira habitual. Este algoritmo baseia-se no apresentado por Schönfinkel em 1924 e permite representar cada símbolo de abstracção por um só combinador. Desde modo evita o crescimento explosivo da tradução de um λ-termo durante abstracções sucessivas. Mostramos de facto, que λ-termos de tamanho n se traduzem para $\mathcal{C}\ell(\mathcal{K})$-termos de tamanho $\mathcal{O}(n^2)$. Provamos ainda a correcção do algoritmo apresentado e estabelecemos algumas relações entre o sistema combinatório $\mathcal{C}\ell(\mathcal{K})$ e o λ-calculus puro. Em particular, mostramos que o número de passos de redução-β em ordem normal, necessário para reduzir um λ-termo M à sua forma normal N, é maior ou igual ao número de passos de redução fraca, necessário para reduzir M_H a $N_H = N$, quando N não contém abstracções. Esta propriedade afecta directamente o número de estados intermédios durante a redução dos termos e torna o sistema especialmente adequado para aplicações prácticas. Finalmente apresentamos três subsistemas de $\mathcal{C}\ell(\mathcal{K})$, que correspondem respectivamente ao $\lambda\mathcal{I}$-calculus, ao λ-calculus linear e ao λ-calculus afim.

6.1 Introdução

O primeiro algoritmo de abstracção surgiu com o artigo de Schönfinkel, (Schönfinkel, 1924), e estabelece a correspondência entre o λ-calculus puro e o sistema de lógica combinatória baseado nos combinadores primitivos B, C, I, K e S, que já apresentamos no capítulo 2. O algoritmo utilizado é extremamente simples; contudo tem resultados desastrosos em termos de complexidade, quando aplicado repetidamente, i.e. na tradução de λ-termos que contêm mais do que uma abstracção. Até hoje fizeram-se diversas tentativas para melhorar o algoritmo original, mudando para isso o conjunto de combinadores primitivos, de modo a obter termos combinatórios mais compactos[1]. Abdali em (Abdali, 1976), Statman em (Statman, 1983), e Kennaway & Sleep em (Kennaway & Sleep, 1984) definiram respectivamente operadores de abstracção múltipla, capazes de abstrair qualquer número de variáveis simultaneamente. Com estes algoritmos obtêm-se bons resultados desde que o número de abstracções simultâneas seja grande. Mas, como já foi referido por Turner em (Turner, 1979a), a necessidade de abstrair variáveis diferentes surge na práctica normalmente em estados sucessivos, o que torna a utilização de operadores de abstracção simples, i.e.

[1] O algoritmo de abstracção de Curry, que apresentamos no capítulo 2 é muito ineficiente, mas presta-se, devido a sua simplicidade, a estudos teóricos.

abstraindo uma variável de cada vez, geralmente mais vantajosa. Outro método, que é frequentemente utilizado na implementação de linguagens funcionais, é baseado na noção de *supercombinador*, conceito introduzido por Hugues, (Hugues, 1982). Turner propõe em (Turner, 1979a) um algoritmo de abstracção simples com boas propriedades para aplicações prácticas (ver comentários em (Mulder, 1985), (Peyton Jones, 1987) e (Barendregt, 1990)) que é actualmente utilizado na implementação da linguagem funcional SASL, cf. (Turner, 1979b). Nesta proposta Turner apresenta uma extensão do sistema combinatório utilizado por Schönfinkel com três combinadores primitivos, respectivamente designados por C', B' e S'. Com esta modificação consegue representar λ-termos de tamanho n por termos combinatórios de tamanho $\mathcal{O}(n^2)$, enquanto que o tamanho da representação no sistema de Schönfinkel é $\mathcal{O}(n^3)$. A mesma relação existe entre os números de passos de redução em ordem normal, necessários para reduzir termos correspondentes nos três sistemas.

O sistema que apresentamos neste capítulo é igualmente baseado no sistema de Schönfinkel. Como exemplo, consideremos a expressão

$$M = \lambda xy.u(z(xv)(yv)).$$

Para traduzir o termo M num termo combinatório do sistema de Schönfinkel, é necessário abstrair primeiro a variável y da expressão $u(z(xv)(yv))$ e em seguida abstrair a variável x do resultado obtido. No primeiro passo, o algoritmo de Schönfinkel produz o termo

$$\mathsf{B}u(\mathsf{B}(z(xv))(\mathsf{CI}v))$$

que contém a sequência de combinadores primitivos B, B, C e I. Esta sequência irá ser codificada em $\mathcal{C}\ell(\mathcal{K})$ através do combinador indexado Φ_{bbc} definido pela regra de redução

$$\Phi_{bbc}x_1x_2x_3x_4 \geq_{1w} x_1(x_2(x_4x_3)).$$

Desde modo obtemos o termo

$$\Phi_{bbc}u(z(xv))v$$

em vez de $\mathsf{B}(u(\mathsf{B}(z(xv))(\mathsf{CI}v))$. Repetindo o mesmo processo na abstracção da variável x

obtemos

$$\Phi_{cbbc} v (\Phi_{bbc} u) z v$$

como representação de M em $\mathcal{C}\ell(\mathcal{K})$. O resultado do algoritmo de Schönfinkel produz o termo

$$\mathsf{B}(\mathsf{B}u)(\mathsf{C}(\mathsf{BB}(\mathsf{B}z(\mathsf{CI}v)))(\mathsf{CI}v))$$

e a representação de M no sistema de Turner é

$$\mathsf{B}(\mathsf{B}u)(\mathsf{C}'\mathsf{B}(\mathsf{B}z(\mathsf{CI}v))(\mathsf{CI}v)).$$

Consideremos agora a redução do termo Mxy em cada um dos três sistemas. Em $\mathcal{C}\ell(\mathcal{K})$ obtém-se a forma normal em apenas dois passos.

$$\begin{aligned}
\Phi_{cbbc} v (\Phi_{bbc} u) z v x y &\geq_{1w} \Phi_{bbc} u (z(xv)) v y \\
&\geq_{1w} u(z(xv)(yv)).
\end{aligned}$$

Comparativamente consideremos a redução no sistema de Turner que envolve bastante mais passos:

$$\begin{aligned}
\mathsf{B}(\mathsf{B}u)(\mathsf{C}'\mathsf{B}(\mathsf{B}z(\mathsf{CI}v))(\mathsf{CI}v))xy &\geq_{1w} \mathsf{B}u(\mathsf{C}'\mathsf{B}(\mathsf{B}z(\mathsf{CI}v))(\mathsf{CI}v)x)y \\
&\geq_{1w} u(\mathsf{C}'\mathsf{B}(\mathsf{B}z(\mathsf{CI}v))(\mathsf{CI}v)xy) \\
&\geq_{1w} u(\mathsf{B}(\mathsf{B}z(\mathsf{CI}v)x)(\mathsf{CI}v)y) \\
&\geq_{1w} u(\mathsf{B}z(\mathsf{CI}v)x(\mathsf{CI}vy)) \\
&\geq_{1w} u(z(\mathsf{CI}vx)(\mathsf{CI}vy)) \\
&\geq_{1w} u(z(\mathsf{I}xv)(\mathsf{CI}vy)) \\
&\geq_{1w} u(z(\mathsf{I}xv)(\mathsf{I}yv)) \\
&\geq_{1w} u(z(xv)(\mathsf{I}yv)) \\
&\geq_{1w} u(z(xv)(yv)).
\end{aligned}$$

E é óbvio que a redução no sistema de Schönfinkel é ainda mais complexa.

6.2 O sistema combinatório $\mathcal{C}\ell(\mathcal{K})$

Nesta secção apresentamos o sistema combinatório $\mathcal{C}\ell(\mathcal{K})$. Começamos por descrever o seu conjunto de combinadores primitivos $\mathcal{K}$ e as respectivas regras de redução.

Definição 6.2.1 *O conjunto de índices $\mathcal{A}$ é o menor conjunto satisfazendo as regras seguintes:*

- $\epsilon \in \mathcal{A}$;
- $\alpha \in \mathcal{A} \quad \Rightarrow \quad c\cdot\alpha \in \mathcal{A}$;
- $\alpha \in \mathcal{A} \quad \Rightarrow \quad b\cdot\alpha \in \mathcal{A}$;
- $\alpha_1, \alpha_2 \in \mathcal{A} \quad \Rightarrow \quad (\alpha_1, \alpha_2) \in \mathcal{A}$.

Como já anteriormente, denotamos por ϵ e $\cdot$ respectivamente a palavra vazia e a operação de concatenação.

O conjunto das constantes primitivas de $\mathcal{C}\ell(\mathcal{K})$ é

$$\mathcal{K} = \{\mathsf{K}\} \cup \{\Phi_\alpha\}_{\alpha\in\mathcal{A}}.$$

Definição 6.2.2 *O conjunto $\mathbf{Cl}(\mathcal{K})$ dos $\mathcal{C}\ell(\mathcal{K})$-termos define-se como habitualmente por:*

- $\mathcal{V} \subseteq \mathbf{Cl}(\mathcal{K})$;
- $\mathcal{K} \subseteq \mathbf{Cl}(\mathcal{K})$;
- $X, Y \in \mathbf{Cl}(\mathcal{K}) \quad \Rightarrow \quad XY \in \mathbf{Cl}(\mathcal{K})$.

Definição 6.2.3 *Definimos o grau do combinador primitivo K como 1 e o grau de um combinador da forma Φ_α como $\sharp\alpha$, que denota o número de ocorrências das letras c e b em α. Um termo da forma $\tau X_1 \ldots X_{n+1}$, onde τ é um combinador primitivo de grau n, diz-se um redex fraco.*

O contractum $\kappa(R)$ de um $\mathcal{C}\ell(\mathcal{K})$-redex R define-se indutivamente por

- $\kappa(\mathsf{K}XY) = X$;
- $\kappa(\Phi_\epsilon X) = X$;
- $\kappa(\Phi_{c\cdot\alpha} X X_1 \dots X_{\sharp\alpha+1}) = (\kappa(\Phi_\alpha X_1 \dots X_{\sharp\alpha+1}))X$;
- $\kappa(\Phi_{b\cdot\alpha} X X_1 \dots X_{\sharp\alpha+1}) = X(\kappa(\Phi_\alpha X_1 \dots X_{\sharp\alpha+1}))$;
- $\kappa(\Phi_{(\alpha_1,\alpha_2)} X_1 \dots X_{\sharp\alpha_1} Y_1 \dots Y_{\sharp\alpha_2} Z) = (\kappa(\Phi_{\alpha_1} X_1 \dots X_{\sharp\alpha_1} Z))\ (\kappa(\Phi_{\alpha_2} Y_1 \dots Y_{\sharp\alpha_2} Z))$;

e a cada combinador $\tau \in \mathcal{K}$ de grau $n \geq 0$ fica então associado a regra de redução

$$\tau x_1 \dots x_{n+1} \geq_{1w} \kappa(\tau x_1 \dots x_{n+1}).$$

A relação $\geq_{1\omega}$ estende-se ao conjunto de $\mathcal{C}\ell(\mathcal{K})$-termos da maneira habitual, i.e. como na definição 2.2.2.

Exemplo 6.2.4 *Aos combinadores Φ_c, Φ_{cc}, Φ_{bb} e $\Phi_{(b,b)}$ estão respectivamente associadas as seguintes regras de redução:*

$$\begin{array}{rlll} & \Phi_c xy & \geq_{1w} & yx \\ & \Phi_{cc} xyz & \geq_{1w} & zyx \\ & \Phi_{bb} xyz & \geq_{1w} & x(yz) \\ e & \Phi_{(b,b)} xyz & \geq_{1w} & xz(yz) \end{array}$$

6.3 Abstracção em $\mathcal{C}\ell(\mathcal{K})$

A função $(\ )_H$ que traduz cada λ-termo $M \in \Lambda$ para um $\mathcal{C}\ell(\mathcal{K})$-termo correspondente $M_H \in \mathbf{Cl}(\mathcal{K})$ define-se como habitualmente (definição 2.2.7), a menos do operador de abstracção λ^* que passamos a descrever em seguida.

Como já referimos anteriormente, cada combinador indexado $\Phi_\alpha \in \mathcal{K}$ irá representar a sequência de combinadores primitivos introduzidos durante a abstracção de uma variável pelo algoritmo de Schönfinkel[2]. Consequentemente o cálculo de $\lambda^* x.X$ é constituído por duas fases.

Primeiro abstrai-se a variável x de X utilizando para isso o algoritmo de Schönfinkel, cujo resultado denotamos por $[x]X$:

$$
\begin{array}{lcll}
[x]\,U & = & \mathsf{K}U & \text{se } x \notin FV(U) \\
[x]\,Ux & = & U & \text{se } x \notin FV(U) \\
[x]\,x & = & \mathsf{I} & \\
[x]\,UV & = & \mathsf{C}([x]\,U)V & \text{se } x \in FV(U) \text{ e } x \notin FV(V) \\
[x]\,UV & = & \mathsf{B}U([x]\,V) & \text{se } x \notin FV(U) \text{ e } x \in FV(V) \\
[x]\,UV & = & \mathsf{S}([x]\,U)([x]\,V) & \text{se } x \in FV(U) \text{ e } x \in FV(V)
\end{array}
$$

Numa segunda fase calcula-se o termo $\lambda^* x.X$ a partir de $[x]X$:

Se nenhum dos combinadores I, C, B ou S aparecer em $[x]X$, então toma-se

$$\lambda^* x.X = [x]X.$$

Caso contrário, seja

$$\lambda^* x.X = \Phi_\alpha X_1 \ldots X_n$$

onde

$$
\begin{array}{rcl}
\alpha & = & \mathbf{i}([x]X) \\
\text{e} \quad X_1, \ldots, X_n & = & \mathbf{r}([x]X)
\end{array}
$$

com $\mathbf{i}$ e $\mathbf{r}$ definidos abaixo.

Analisando o algoritmo de Schönfinkel descrito acima, é fácil ver, que o termo $[x]X$ só contém um combinador I, C, B ou S como subtermo, se estiver na forma

$$\mathsf{I},\ \mathsf{C}YZ,\ \mathsf{B}ZY,\ \text{ ou }\ \mathsf{S}Y_1Y_2$$

e que neste caso a sequência de combinadores primitivos introduzidos pelo algoritmo continua em Y, respectivamente em Y_1 e Y_2. Estes combinadores são então, um por um, coleccionados

[2] Ocorrências das letras c e b em α representam os combinadores C e B, enquanto que I e S estão respectivamente associados à palavra vazia ϵ e ao aparecimento de pares ordenados em α.

pela função $\mathbf{i}$ para formarem o índice α. Os restantes subtermos Z são ao mesmo tempo guardados na sequência $X_1, \ldots, X_n$ pela função $\mathbf{r}$. As funções $\mathbf{i}$ e $\mathbf{r}$ definem-se através dos dois algoritmos seguintes:

$$\begin{array}{lcl}
\mathbf{i}(\mathsf{I}) & = & \epsilon \\
\mathbf{i}(\mathsf{C}XY) & = & c\cdot\mathbf{i}(X) \\
\mathbf{i}(\mathsf{B}XY) & = & b\cdot\mathbf{i}(Y) \\
\mathbf{i}(\mathsf{S}XY) & = & (\mathbf{i}(X), \mathbf{i}(Y)) \\
\mathbf{i}(X) & = & b \quad \text{caso contrário.}
\end{array}$$

$$\begin{array}{lcl}
\mathbf{r}(\mathsf{I}) & = & \epsilon \\
\mathbf{r}(\mathsf{C}XY) & = & Y, \mathbf{r}(X) \\
\mathbf{r}(\mathsf{B}XY) & = & X, \mathbf{r}(Y) \\
\mathbf{r}(\mathsf{S}XY) & = & \mathbf{r}(X), \mathbf{r}(Y) \\
\mathbf{r}(X) & = & X \quad \text{caso contrário.}
\end{array}$$

Sem qualquer dificuldade verifica-se o seguinte:

- se $\Phi_{i([x]X)}$ for de grau $n \geq 0$, então a sequência $\mathbf{r}([x]X)$ tem exactamente n elementos;
- se $X \in \mathbf{Cl}(\mathcal{K})$, então $\lambda^* x.X \in \mathbf{Cl}(\mathcal{K})$, e em particular $\lambda^* x.X$ não contém nenhum dos combinadores I, C, B ou S como subtermo.

Exemplo 6.3.1 *Tem-se*

$$\lambda^* y.u(z(xv)(yv)) = \Phi_{bbc}u(z(xv))v$$

uma vez que

$$[y]u(z(xv)(yv)) = \mathsf{B}u(\mathsf{B}(z(xv))(\mathsf{CI}v)),$$

e por

$$\begin{array}{lcl}
\mathbf{i}(\mathsf{B}u(\mathsf{B}(z(xv))(\mathsf{CI}v))) & = & b\cdot\mathbf{i}(\mathsf{B}(z(xv))(\mathsf{CI}v)) \\
 & = & bb\cdot\mathbf{i}(\mathsf{CI}v) \\
 & = & bbc\cdot\mathbf{i}(\mathsf{I}) \\
 & = & bbc
\end{array}$$

e

$$\begin{aligned}\mathbf{r}(\mathsf{B}u(\mathsf{B}(z(xv))(\mathsf{CI}v))) &= u, \mathbf{r}(\mathsf{B}(z(xv))(\mathsf{CI}v)) \\ &= u, z(xv), \mathbf{r}(\mathsf{CI}v) \\ &= u, z(xv), v, \mathbf{r}(\mathsf{I}) \\ &= u, z(xv), v.\end{aligned}$$

Lema 6.3.2 *Seja $x \in \mathcal{V}$ uma variável e $X \in \mathbf{Cl}(\mathcal{K})$. Então,*

1. $x \notin \lambda^* x.X$
2. $(\lambda^* x.X)x \geq_w X$.

Prova

1. Trivial.
2. Se $x \notin FV(X)$, então $(\lambda^* x.X)x = \mathsf{K}Xx \geq_{1w} X$.

   Daqui em diante, suponhamos $x \in FV(X)$, o que exclui a aplicação da primeira equação do algoritmo de Schönfinkel durante o cálculo de $[x]X$.

   Se $\lambda^* x.X = [x]X$, segue $X = Ux$, $x \notin FV(U)$ e $(\lambda^* x.X)x = Ux$.

   Finalmente, mostramos por indução sobre n que, se $X \in \mathbf{Cl}(\mathcal{K})$, $x \in FV(X)$, $\alpha' = \mathbf{i}([x]X)$ e $X_1, \ldots, X_n = \mathbf{r}([x]X)$, então $\kappa(\Phi_{\alpha'} X_1 \ldots X_n x) = X$.

   Seja $n = 0$, donde $\alpha' = \epsilon$ e $[x]X = \mathsf{I}$. Logo $X = x$ e $\kappa(\Phi_\epsilon x) = x$.

   Caso contrário, α' é da forma $\alpha' = c \cdot \alpha$, $\alpha' = b \cdot \alpha$ ou $\alpha' = (\alpha_1, \alpha_2)$. Provamos o resultado para $\alpha' = b \cdot \alpha$, que é o mais complexo dos três casos. De facto existem duas possibilidades:

   (i) Se o índice $\alpha' = b \cdot \alpha$ resultar de uma aplicação da terceira equação do algoritmo, então verifica-se $[x]X = \mathsf{B}UY$. E consequentemente $X = UV$, tal que $x \notin FV(U)$ e $x \in FV(V)$, $Y = [x]V$ e $X_1 = U$. Por outro lado, temos $\mathbf{i}([x]V) = \alpha$ e $\mathbf{r}([x]V) = X_2, \ldots, X_n$. Da hipótese de indução decorre $\kappa(\Phi_\alpha X_2 \ldots X_n x) = V$, donde $\kappa(\Phi_{b \cdot \alpha} X_1 \ldots X_n x) = X_1(\kappa(\Phi_\alpha X_2 \ldots X_n x)) = UV$.

(ii) Se α' resultar da última equação do algoritmo, então verifica-se $\alpha' = b$ e $[x]X = X_1$. Podemos sem qualquer dificuldade concluir que $[x]X$ não é da forma I, $\mathsf{C}UV$, $\mathsf{B}UV$ ou $\mathsf{S}UV$. Consequentemente, $[x]X = X_1$ resulta da segunda equação do algoritmo de Schönfinkel. Daqui decorre $X = X_1x$, onde $x \notin FV(X_1)$ e $\kappa((\lambda^* x.X)x) = \kappa(\Phi_b X_1 x) = X_1 x$. •

O resultado que se segue mostra a correcção do algoritmo de abstracção definido para o sistema $\mathcal{C}\ell(\mathcal{K})$.

Proposição 6.3.3 *O termo $(\lambda^* x.X)Y$ comporta-se como um β-redex, i.e.*

$$(\lambda^* x.X)Y \geq_w X[Y/x].$$

Prova Começamos por notar que

$$X \geq_w Z \quad \Rightarrow \quad X[Y/x] \geq_w Z[Y/x] \tag{6.1}$$

uma vez que esta relação se verifica para cada redex em X e por

$$(AB)[Y/x] = (A[Y/x])(B[Y/x]).$$

Daqui resulta,

$$\begin{aligned}
(\lambda^* x.X)Y &= (\lambda^* x.X)(x[Y/x]) & \\
&= ((\lambda^* x.X)x)[Y/x] & \text{lema 6.3.2 1.} \\
&\geq_w X[Y/x] & \text{de (6.1) e lema 6.3.2 2.} \quad \bullet
\end{aligned}$$

Veremos na proposição 6.3.5 que, em comparação com sistemas tradicionais, existe uma correspondência mais estreita entre a redução fraca em $\mathcal{C}\ell(\mathcal{K})$ e a redução-β no λ-calculus. Nomeadamente, cada redução fraca da representação M_H para Y em $\mathcal{C}\ell(\mathcal{K})$ de um λ-termo $M \in \Lambda$ corresponde a uma redução-β de M, tal que a tradução do resultado para $\mathcal{C}\ell(\mathcal{K})$ produz o mesmo termo Y.

Lema 6.3.4 *Sejam $M, N \in \Lambda$ e $X, Y \in \mathbf{Cl}(\mathcal{K})$. Então,*

1. $(\lambda^* y.X)[Y/x] = \lambda^* y.(X[Y/x])$, *se* $x \notin \{y\} \cup FV(Y)$;

2. $(M[N/x])_H = M_H[N_H/x]$.

Prova

1. Por indução estrutural em X.

2. Por indução estrutural em M utilizando 1. $\bullet$

Proposição 6.3.5 *Seja $M \in \Lambda$ e $Y \in \mathbf{Cl}(\mathcal{K})$. Então,*

$$M_H \geq_w Y \quad \Rightarrow \quad \exists N \in \Lambda \ : \ M \to_\beta N \text{ e } N_H = Y.$$

Prova Começamos por mostrar que o algoritmo de abstracção definido em $\mathcal{C}\ell(\mathcal{K})$ não cria nenhum redex novo. Por outras palavras, se um redex R ocorrer em $\lambda^* x.P_H$, então R é também subtermo de P_H. De facto, verifica-se ou

$$P_H = Xx \qquad \text{e} \qquad \lambda^* x.P_H = X$$

ou então

$$\lambda^* x.P_H = \tau' Y_1' \dots Y_m'$$

onde τ' é um combinador primitivo de grau m e tal que cada Y_i' é um subtermo de P_H. No primeiro caso o resultado verifica-se trivialmente. Para o outro caso basta notar que $\tau' Y_1' \dots Y_m'$ não é um redex. Logo R é subtermo de Y_i', para algum $i \in \{1, \dots, m\}$, e de Y_i' ser subtermo de P_H resulta que o redex R é subtermo de P_H.

Consideremos então um redex $\tau X_1 \dots X_{n+1}$ em M_H. Das considerações anteriores decorre que $\tau X_1 \dots X_{n+1}$ resulta de uma aplicação em M, i.e. de um subtermo da forma UV em M tal que

$$U_H = \tau X_1 \dots X_n \qquad \text{e} \qquad V_H = X_{n+1}.$$

Uma vez que o termo $U_H V_H$ não foi destruído durante a restante construção de M_H, resulta ainda que o termo M não contém nenhuma abstracção sobre as variáveis livres de UV.

A seguir, provamos por indução sobre o número k de aplicações da regra

$$\lambda^* x.Xx = [x]Xx = X, \quad x \notin FV(X) \tag{6.2}$$

durante a construção do termo U_H, que existe $P \in \Lambda$ tal que

$$U \to_\beta \lambda x.P \quad \text{onde} \quad P_H \neq Zx \quad \text{quando} \quad x \notin FV(Z)$$

e tal que

$$\lambda^* x.P_H = \tau X_1 \ldots X_n.$$

Se $k = 0$, então U é necessariamente da forma $\lambda x.P$, com $P_H \neq Zx$ ou então $x \in FV(Z)$, e $\lambda^* x.P_H = \tau X_1 \ldots X_n$.

Suponhamos agora, que o resultado se verifica para k ou menos aplicações de (6.2) e que durante a construção de $U_H = \tau X_1 \ldots X_n$ a regra (6.2) foi utilizada $k+1$ vezes. É claro que

$$U = (\lambda x.Q)V_1 \ldots V_j, \qquad \text{com } j \geq 0$$

e tal que

$$(\lambda^* x.Q_H) = \tau X_1 \ldots X_m \quad \text{e} \quad (V_1)_H = X_{m+1}, \ldots, (V_j)_H = X_n.$$

Se $\lambda^* x.Q_H$ não resultar de (6.2), segue $j = 0$ e o resultado é válido. Caso contrário, tem-se $Q_H = \tau X_1 \ldots X_m x$, onde $x \notin FV(\tau X_1 \ldots X_m)$ e

$$(\lambda^* x.Q_H)(V_1)_H = \tau X_1 \ldots X_m X_{m+1} = Q_H[(V_1)_H/x] = (Q[V_1/x])_H.$$

Esta última equação decorre do lema 6.3.4. Consequentemente,

$$U \to_\beta (Q[V_1/x])V_2 \ldots V_j = N'$$

e

$$N'_H = \tau X_1 \ldots X_n,$$

onde N'_H foi calculado utilizando a regra (6.2) no máximo k vezes. Logo, $N' \to_\beta N$ tal que $N = \lambda x.P$, com $P_H \neq Zx$ ou então $x \in FV(Z)$, e $\lambda^* x.P_H = N_H = \tau X_1 \ldots X_n$.

$$\begin{array}{ccccccc}
M = M_1 & \rightarrow_\beta & M_2 & \rightarrow_\beta & \ldots & \rightarrow_\beta & M_n = N \\
\downarrow (\,)_H & & \downarrow (\,)_H & & & & \downarrow (\,)_H \\
M_H = Y_1 & \geq_{1w} & Y_2 & \geq_{1w} & \ldots & \geq_{1w} & Y_n = Y
\end{array}$$

Figura 6.3

Da prova do lema 6.3.2 decorre ainda

$$\kappa(\tau X_1 \ldots X_{n+1}) = \kappa((\lambda^* x.P_H)V_H) = P_H[V_H/x] = (P[V/x])_H.$$

Consequentemente, existe para cada passo de redução fraca a partir de M_H uma redução-β de M, nomeadamente a substituição de UV por $P[V/x]$, tal que se obtém a igualdade via $(\,)_H$, uma vez que as variáveis livres de $P[V/x]$ ocorrem também livres em UV.

Finalmente consideremos uma redução fraca do termo M_H para Y, i.e. $M_H \geq_w Y$. Então existe uma sequência finita de termos $Y_1, \ldots Y_n \in \mathbf{Cl}(\mathcal{K})$ tal que

$$M_H = Y_1 \geq_{1w} \ldots \geq_{1w} Y_n = Y.$$

Aplicando o resultado obtido a cada passo nesta redução, obtemos a a existência de N por transitividade, como está ilustrado na figura 6.3. $\bullet$

Note-se, que o recíproco da propriedade anterior não é em geral verdadeiro, uma vez que, para $M \in \Lambda$ pode acontecer que o termo M_H está em forma normal fraca, enquanto que M não está em forma normal-β. Contudo, mostramos que isto não acontece desde que o termo M se reduz a uma forma normal que não contém abstracções.

Lema 6.3.6 *Sejam $M, N \in \Lambda$ tal que $M \rightarrow_\beta N$. Se o termo N não contiver nenhum símbolo de abstracção, então $M_H \geq_w N_H$.*

Prova De $M \rightarrow_\beta N$, decorre pelo teorema 2.1.10 a existência de uma redução em ordem normal de M a N, i.e.

$$M = M_0 \rightarrow_{1l\beta} M_1 \rightarrow_{1l\beta} \ldots \rightarrow_{1l\beta} M_n = N.$$

Uma vez que N não contém nenhum símbolo de abstracção, é claro que nenhum dos termos M_i, para $i \in \{0, \ldots, n\}$, é uma abstracção, i.e. da forma $\lambda x.P$, onde $x \in \mathcal{V}$ e $P \in \Lambda$. Mais ainda, note-se que com a prova da alínea 2. do lema 6.3.2, mostramos na verdade o resultado mais forte

$$((\lambda^* x.X)x = X \ \vee \ (\lambda^* x.X)x \geq_{1w} X)$$

e consequentemente podemos concluir para a proposição 6.3.3 também

$$((\lambda^* x.X)Y = X[Y/x] \ \vee \ (\lambda^* x.X)Y \geq_{1w} X[Y/x]).$$

A seguir mostramos por indução estrutural em $M = M_0$ que se verifica

$$(M_0)_H = (M_1)_H \quad \text{ou} \quad (M_0)_H \geq_{1w} (M_1)_H.$$

Se $M_0 = (\lambda x.P)QU_1 \ldots U_k$, com $k \geq 0$, então

$$\begin{array}{rcl} (M_0)_H & = & (\lambda^* x.P_H)Q_H(U_1)_H \ldots (U_k)_H \\ & = \text{ou} \geq_{1w} & P_H[Q_H/x](U_1)_H \ldots (U_k)_H \\ & = & (P[Q/x])_H(U_1)_H \ldots (U_k)_H \\ & = & (M_1)_H. \end{array}$$

Se $M_0 = xU_1 \ldots U_k$, para $k \geq 1$, e $M_1 = xU_1 \ldots U_i' \ldots U_k$ tal que $U_i \rightarrow_{1l\beta} U_i'$, então o resultado decorre directamente da hipótese de indução, uma vez que U_i não pode ser nenhuma abstracção.

Repetindo o processo sucessivamente para os passos de redução $M_1 \rightarrow_{1l\beta} M_2$ até $M_{n-1} \rightarrow_{1l\beta} M_n = N$ obtemos uma redução fraca de M_H para N_H

$$M_H = (M_{k_0})_H \geq_{1w} \ldots \geq_{1w} (M_{k_m})_H = N_H$$

tal que $m \leq n$ e $0 = k_0 < k_1 < \ldots < k_m = n$. $\bullet$

Corolário 6.3.7 *Para qualquer redução em ordem normal de comprimento n de um termo $M \in \Lambda$ a outro termo $N \in \Lambda$ que não contém abstracções, existe uma redução fraca de comprimento m de M_H para $N_H = N$ tal que $m \leq n$.*

6.3.1 Complexidade

A seguir fazemos uma breve análise da complexidade dos termos produzidos pelo algoritmo de tradução apresentado. Para isso consideramos um λ-termo M contendo k abstracções e $l+1$ átomos, donde l aplicações. No $\mathcal{C}\ell(\mathcal{K})$-termo correspondente M_H contabilizamos o número de átomos e os símbolos que ocorrem nos índices dos combinadores, onde contamos as ocorrências de c's, b's, parenteses e vírgulas em todos os índices e denotamos esta medida pela letra a.

Começamos por notar que a aplicação das regras $x_H = x$ e $(MN)_H = M_H N_H$ durante a tradução não altera a complexidade dos termos.

Consideramos então a primeira abstracção efectuada correspondendo à tradução de um subtermo da forma $\lambda x.P$, onde P não contém qualquer símbolo de abstracção. Suponhamos que $\lambda^* x.P$ foi obtido por

$$\begin{array}{lrcll} & \lambda^* x.P & = & \mathsf{K}P & \text{se } x \notin FV(P) \\ \text{ou} & \lambda^* x.P & = & P' & \text{se } P = P'x \ \text{ e } x \notin FV(P') \end{array}$$

O primeiro caso é obviamente o pior e aumenta o número de átomos de 1. Caso contrário, foi introduzido um combinador indexado Φ_α e o número de átomos em $\lambda^* x.P$ é o mesmo ou decresce relativamente a $\lambda x.P$, dependendo da variável x ter uma ou mais ocorrências em P. No pior dos casos, cada aplicação em P dá origem a 3 símbolos no índice α. Logo tem-se $a = 3l$, enquanto que o número de átomos em $\lambda^* x.P$ é na pior hipótese $l+2$ com $l+1$ aplicações.

Iterando, obtemos depois de duas abstracções efectuadas $a = 3l + 3(l+1)$ e $l+3$ átomos. Resulta então, que o termo M_H contém no pior caso, e depois de efectuadas as k abstracções, $l+k+1$ átomos, enquanto que

$$a = 3l + 3(l+1) + \ldots + 3(l+k-1) = 3kl + \frac{3(k-1)k}{2}.$$

Concluímos que cada λ-termo M com n símbolos é traduzido para um $\mathcal{C}\ell(\mathcal{K})$-termo M_H cujo número de símbolos é $\mathcal{O}(n^2)$.

6.4 Mais relações entre o λ-calculus puro e $\mathcal{C}\ell(\mathcal{K})$

Nesta secção estabelecemos mais algumas relações entre o λ-calculus puro e o sistema de lógica combinatória $\mathcal{C}\ell(\mathcal{K})$. Começamos por definir a função $(\ )_\lambda$ que transforma cada $\mathcal{C}\ell(\mathcal{K})$-termo num λ-termo correspondente.

Definição 6.4.1 *Para $X \in \mathbf{Cl}(\mathcal{K})$ define-se $X_\lambda \in \Lambda$ por indução estrutural em X:*

$$
\begin{array}{lcl}
(x)_\lambda & = & x \\
(UV)_\lambda & = & U_\lambda V_\lambda \\
(\mathsf{K})_\lambda & = & \lambda xy.x \\
(\Phi_\epsilon)_\lambda & = & \lambda x.x \\
(\Phi_{c\cdot\alpha})_\lambda & = & \lambda x_1 x_2 \ldots x_n.M x_1, \\
 & & \textit{se } (\Phi_\alpha)_\lambda = \lambda x_2 \ldots x_n.M \\
(\Phi_{b\cdot\alpha})_\lambda & = & \lambda x_1 x_2 \ldots x_n.x_1 M, \\
 & & \textit{se } (\Phi_\alpha)_\lambda = \lambda x_2 \ldots x_n.M \\
(\Phi_{(\alpha_1,\alpha_2)})_\lambda & = & \lambda x_1 \ldots x_m y_1 \ldots y_n z.MN, \\
 & & \textit{se } (\Phi_{\alpha_1})_\lambda = \lambda x_1 \ldots x_m z.M \textit{ e } (\Phi_{\alpha_2})_\lambda = \lambda y_1 \ldots y_n z.N
\end{array}
$$

Verifica-se facilmente que a função $(\ )_\lambda$ está bem definida, i.e. $\forall X \in \mathbf{Cl}(\mathcal{K})\ \exists^1 M \in \Lambda$ tal que $(X)_\lambda = M$.

Os dois lemas que se seguem irão ser usados nas provas dos restantes resultados deste capítulo e podem ser demonstrados por indução sobre n.

Lema 6.4.2 *Seja $\tau \in \mathcal{K}$ um combinador de grau n. Então, o termo $(\tau)_\lambda$ é da forma $\lambda x_1 \ldots x_{n+1}.M$, tal que $M \in \Lambda$ é um termo que não contém abstracções.*

Lema 6.4.3 *Seja $\tau \in \mathcal{K}$ um combinador de grau n e $(\tau)_\lambda = \lambda x_1 \ldots x_{n+1}.M$. Se $\kappa(\tau X_1 \ldots X_{n+1}) = Y$, então $(Y)_\lambda = M[(X_1)_\lambda/x_1, \ldots, (X_{n+1})_\lambda/x_{n+1}]$.*

Proposição 6.4.4 *Sejam $X, Y \in \mathbf{Cl}(\mathcal{K})$. Então,*

1. $X \geq_w Y \;\Rightarrow\; X_\lambda \to_\beta Y_\lambda$;
2. $X =_w Y \;\Rightarrow\; X_\lambda =_\beta Y_\lambda$.

Prova

1. Basta mostrar que para qualquer combinador $\tau \in \mathcal{K}$ de grau n e qualquer sequência de termos $X_1, \ldots, X_{n+1} \in \mathbf{Cl}(\mathcal{K})$ se verifica o seguinte: se $\tau X_1 \ldots X_{n+1} \geq_{1w} Y$, então $(\tau X_1 \ldots X_{n+1})_\lambda \to_\beta Y_\lambda$. De facto temos,

$$\begin{array}{rcll} (\tau X_1 \ldots X_{n+1})_\lambda & = & (\tau)_\lambda (X_1)_\lambda \ldots (X_{n+1})_\lambda & \text{def. 6.4.1} \\ & = & (\lambda x_1 \ldots x_{n+1}.M)(X_1)_\lambda \ldots (X_{n+1})_\lambda & \text{lema 6.4.2} \\ & \to_\beta & M[(X_1)_\lambda/x_1, \ldots, (X_{n+1})_\lambda/x_{n+1}] & \\ & = & (Y)_\lambda & \text{lema 6.4.3} \end{array}.$$

2. Tem-se $X =_w Y$ se e só se existir uma sequência finita de $\mathcal{C}\ell(\mathcal{K})$-termos $X = X_0, \ldots, X_n = Y$, com $n \geq 0$, tal que para $0 \leq i \leq n-1$ se verifica

$$X_i \geq_\omega X_{i+1} \qquad \text{ou} \qquad X_{i+1} \geq_\omega X_i.$$

Logo, o resultado decorre da alínea anterior. •

Proposição 6.4.5 $M \in \Lambda \;\Rightarrow\; (M_H)_\lambda =_{\beta\eta} M$.

Prova Procedemos por indução estrutural em M. Os casos $M = x$ e $M = M_1 M_2$ são triviais. Seja então $M = \lambda x.P$, donde

$$\begin{array}{rll} ((\lambda x.P)_H)_\lambda & = (\lambda^* x.P_H)_\lambda & \text{def. 2.2.7} \\ & =_\eta \lambda x.(\lambda^* x.P_H)_\lambda x & \\ & = \lambda x.((\lambda^* x.P_H)x)_\lambda & \text{def. 6.4.1} \\ & =_\beta \lambda x.(P_H)_\lambda & \text{prop. 6.4.4 e lema 6.3.2} \\ & =_{\beta\eta} \lambda x.P & \text{hipótese de indução.} \; \bullet \end{array}$$

Proposição 6.4.6 $M \in \Lambda \;\Rightarrow\; ((M_H)_\lambda)_H = M_H$.

Prova Analisando a definição do algoritmo de abstracção, verifica-se que para qualquer λ-termo M o combinador Φ_b não ocorre em M_H: de facto Φ_b ocorreria em M_H se e só se existisse um $\mathcal{C}\ell(\mathcal{K})$-termo $X \in \mathbf{Cl}(\mathcal{K})$ tal que o resultado da abstracção pelo algoritmo de Schönfinkel é da forma $[x]X = \mathsf{B}Y\mathsf{I}$. A impossibilidade dessa hipótese decorre da segunda regra do algoritmo e por B e I não ocorrerem em X.

A seguir mostramos por indução estrutural em α, que para qualquer combinador $\Phi_\alpha \neq \Phi_b$ tal que $(\Phi_\alpha)_\lambda = \lambda x_1 \ldots x_{n+1}.N$ se verifica

$$\lambda^* x_{n+1}.N = \Phi_\alpha x_1 \ldots x_n.$$

De facto tem-se,

$$\begin{array}{lll} & (\Phi_\epsilon)_\lambda = \lambda x.x & \text{e} \quad \lambda^* x.x = \Phi_\epsilon, \\ & (\Phi_c)_\lambda = \lambda xy.yx & \text{e} \quad \lambda^* y.yx = \Phi_c x, \\ & (\Phi_{cb})_\lambda = \lambda xyz.yzx & \text{e} \quad \lambda^* z.yzx = \Phi_{cb} xy, \\ & (\Phi_{bb})_\lambda = \lambda xyz.x(yz) & \text{e} \quad \lambda^* z.x(yz) = \Phi_{bb} xy, \\ \text{bem como} & (\Phi_{(b,b)})_\lambda = \lambda xyz.xz(yz) & \text{e} \quad \lambda^* z.xz(yz) = \Phi_{(b,b)} xy. \end{array}$$

Suponhamos então que o resultado é válido para $\alpha \neq b$ e $(\Phi_\alpha)_\lambda = \lambda x_1 \ldots x_{n+1}.N$, i.e.

$$\begin{array}{lrcl} & \lambda^* x_{n+1}.N & = & \Phi_\alpha x_1 \ldots x_n \\ & \alpha & = & \mathbf{i}([x_{n+1}]N) \\ \text{e} & \mathbf{r}([x_{n+1}]N) & = & x_1, \ldots, x_n. \end{array}$$

Então,

$$\begin{array}{lrcl} & (\Phi_{c\alpha})_\lambda & = & \lambda x x_1 \ldots x_{n+1}.Nx \\ & [x_{n+1}](Nx) & = & \mathsf{C}([x_{n+1}]N)x \\ & \mathbf{i}([x_{n+1}](Nx)) & = & c \cdot \mathbf{i}([x_{n+1}]N) \\ \text{e} & \mathbf{r}([x_{n+1}](Nx)) & = & x, \mathbf{r}([x_{n+1}]N) \end{array}$$

donde,

$$\lambda^* x_{n+1}.Nx = \Phi_{c\alpha} x x_1 \ldots x_n.$$

Por outro lado, para

$$(\Phi_{b\alpha})_\lambda = \lambda x x_1 \ldots x_{n+1}.xN$$

tem-se

$$[x_{n+1}](xN) = x, \qquad \text{se} \quad N = x_{n+1},$$

enquanto que

$$[x_{n+1}](xN) = \mathsf{B}x([x_{n+1}]N),$$

caso contrário. Excluímos o primeiro caso por $b\cdot\alpha \neq b$. Donde,

$$\begin{aligned} \mathbf{i}([x_{n+1}](xN)) &= b\cdot\mathbf{i}([x_{n+1}]N) \\ \text{e} \quad \mathbf{r}([x_{n+1}](xN)) &= x, \mathbf{r}([x_{n+1}]N). \end{aligned}$$

Logo,

$$\lambda^* x_{n+1}.xN = \Phi_{b\alpha} x x_1 \ldots x_n.$$

Consideremos agora

$$(\Phi_{(\alpha,b)})_\lambda = \lambda x_1 \ldots x_n y_1 x_{n+1}.N(y_1 x_{n+1}).$$

Segue-se

$$\begin{aligned} [x_{n+1}](N(y_1x_{n+1})) &= \mathsf{S}([x_{n+1}]N)y_1 \\ \mathbf{i}([x_{n+1}](N(y_1x_{n+1}))) &= (\mathbf{i}([x_{n+1}]N), \mathbf{i}(y_1)) = (\alpha, b) \\ \text{e} \quad \mathbf{r}([x_{n+1}](N(y_1x_{n+1}))) &= \mathbf{r}([x_{n+1}]N), \mathbf{r}(y_1) = x_1, \ldots, x_n, y_1. \end{aligned}$$

Concluímos que

$$\lambda^* x_{n+1}.N(y_1x_{n+1}) = \Phi_{(\alpha,b)} x_1 \ldots x_n y_1.$$

Os casos $\Phi_{(b,\alpha)}$ e $\Phi_{(\alpha_1,\alpha_2)}$ são análogos.

Finalmente mostramos $((M_H)_\lambda)_H = M_H$ por indução estrutural em M_H.

Os casos $M_H = x$, $M_H = \mathsf{K}$ e $X = YZ$ verificam-se sem qualquer dificuldade. Suponhamos que $M_H = \Phi_\alpha$ tal que $\alpha \neq b$. Do lema 6.4.2 decorre $(\Phi_\alpha)_\lambda = \lambda x_1 \ldots x_{n+1}.N$ onde o λ-termo N não contém símbolos de abstracção. Logo,

$$\begin{aligned} ((\Phi_\alpha)_\lambda)_H &= (\lambda x_1 \ldots x_{n+1}.N)_H \\ &= \lambda^* x_1.(\lambda^* x_2.(\ldots \lambda^* x_n.(\lambda^* x_{n+1}.N)\ldots)) \\ &= \lambda^* x_1.(\ldots \lambda^* x_n.(\Phi_\alpha x_1 \ldots x_n)\ldots) \\ &= \Phi_\alpha. \quad \bullet \end{aligned}$$

Corolário 6.4.7 *Seja $X \in \mathbf{Cl}(\mathcal{K})$ tal que Φ_b não ocorre em X. Então, $(X_\lambda)_H = X$.*

6.5 Subsistemas de $\mathcal{C}\ell(\mathcal{K})$

Analisando as transformações $(\ )_H$ e $(\ )_\lambda$ verifica-se facilmente que se obtêm subsistemas de $\mathcal{C}\ell(\mathcal{K})$, que correspondem respectivamente ao $\lambda\mathcal{I}$-calculus, ao λ-calculus linear e ao λ-calculus afim, a partir de subconjuntos do conjunto de combinadores primitivos $\mathcal{K}$.

A partir do conjunto

$$\mathcal{K}_\mathcal{I} = \{\Phi_\alpha\}_{\alpha \in \mathcal{A}}$$

obtemos um sistema de lógica combinatória que corresponde ao $\lambda\mathcal{I}$-calculus.

Consideremos o conjunto de índices $\mathcal{A}'$ definido pelas regras seguintes.

- $\epsilon \in \mathcal{A}'$
- $\alpha \in \mathcal{A}' \Rightarrow c\cdot\alpha \in \mathcal{A}'$
- $\alpha \in \mathcal{A}' \Rightarrow b\cdot\alpha \in \mathcal{A}'$

Então

$$\mathcal{K}_\mathcal{L} = \{\Phi_\alpha\}_{\alpha \in \mathcal{A}'}$$

e

$$\mathcal{K}_\mathcal{A} = \{\mathsf{K}\} \cup \{\Phi_\alpha\}_{\alpha \in \mathcal{A}'}$$

geram respectivamente sistemas combinatórios para o λ-calculus linear e o λ-calculus afim, desde que se restringe a formação de termos aplicativos por

$$X, Y \in \mathbf{Cl}(\mathcal{K}_{\{\mathcal{L};\mathcal{A}\}}) \quad \text{e} \quad FV(X) \cap FV(Y) = \emptyset \quad \Rightarrow \quad XY \in \mathbf{Cl}(\mathcal{K}_{\{\mathcal{L};\mathcal{A}\}}).$$

Capítulo 7

Conclusões e trabalho futuro

Nesta dissertação estudamos alguns aspectos do conceito de completude combinatória no λ-calculus, dando ênfase especial a três dos seus subsistemas, nomeadamente ao $\lambda\mathcal{I}$-calculus, ao λ-calculus linear e ao λ-calculus afim. Em particular, abordamos noções relacionadas com a decidibilidade e caracterização das bases de combinadores (próprios) dos quatro sistemas referidos. Mostramos a decidibilidade da completude combinatória para conjuntos finitos de combinadores próprios no λ-calculus linear e no λ-calculus afim, apresentando provas construtivas que permitem representar qualquer λ-termo linear (resp. afim) a partir de uma base de combinadores próprios dada. A partir destes resultados apresentamos ainda uma condição necessária, facilmente verificável, para bases de combinadores lineares (resp. afins) em geral. No $\lambda\mathcal{I}$-calculus provamos a indecidibilidade do mesmo problema para bases de combinadores em forma normal e consequentemente para bases em geral. Estabelecemos no entanto uma condição simples, necessária e suficiente, sobre as bases deste sistema, que são extensões de bases lineares com combinadores próprios e apresentamos também uma condição necessária para a completude combinatória de conjuntos constituídos somente por combinadores próprios. Com o propósito de obter outras caracterizações das bases de combinadores nos quatro sistemas considerados, relacionamos ainda a completude combinatória de um conjunto de combinadores (não tipados) num sistema de λ-calculus e a completude axiomática do conjunto dos respectivos tipos principais, interpretados como fórmulas, no cálculo proposicional minimal correspondente à versão tipada do sistema de λ-calculus. Mostramos que a equivalência dos respectivos problemas se verifica somente no sistema linear e no sistema afim, enquanto que nos restantes casos a completude axiomática é condição suficiente, mas não necessária, para a completude combinatória. Provamos ainda que qualquer λ-termo afim em forma normal fica completamente determinado pelo seu tipo principal, donde resulta a equivalência da igualdade, módulo-β, entre λ-termos lineares e da igualdade, a menos de variantes, entre os respectivos tipos principais. Finalmente apresentamos o sistema combinatório $\mathcal{C}\ell(\mathcal{K})$ e um algoritmo de abstracção associado, que consideramos, por razões de complexidade, particularmente indicado para aplicações prácticas.

A questão de decidibilidade da completude combinatória não ficou, em nenhum dos quatro sistemas considerados, completamente resolvida, i.e. tanto para conjuntos de combinadores em geral como para conjuntos de combinadores próprios, como ilustra a tabela 7.

Decidibilidade de completude combinatória				
	$\lambda\mathcal{K}$-calc.	$\lambda\mathcal{I}$-calc.	λ-calc. linear	λ-calc. afim
Combinadores em forma normal	Não	Não	?	?
Combinadores próprios	?	?	Sim	Sim

Tabela 7

Contudo apresentamos diversas condições necessárias e condições suficientes, que juntamente com as condições anteriormente estabelecidas por Craig e Statman, cf. capítulo 2.3, ajudarão a resolver esta questão para a maioria dos casos particulares. Futuramente gostaríamos, no entanto, de estudar os casos que ficaram em aberto, esperando, em particular, obter alguns resultados para as bases de combinadores próprios nos sistemas $\lambda\mathcal{K}$ e $\lambda\mathcal{I}$, utilizando para isso as representações de combinadores próprios a partir de domínios aplicativos. Note-se que nestes dois casos não se verifica o lema 3.1.12, pelo que a aplicação das linhas numa regra $\psi(M)$, que corresponde a um combinador próprio M, terá que ser analisada em bloco e não individualmente como acontece para os combinadores lineares e afins.

Outra questão relacionada com o nosso trabalho, e que não abordamos nesta dissertação, é a definibilidade ou geração de λ-termos fechados a partir de conjuntos de combinadores não necessariamente completos. Este problema encontra aplicação em diversos campos, como, por exemplo, no processamento de linguagem natural. Sendo, em geral indecidível, estamos convictos de que se podem obter resultados parciais para casos restritos. Um trabalho que aponta neste sentido foi apresentado por Trigg, Hindley e Bunder em (Trigg *et al.*, 1994).

Apêndice A

Funções recursivas parciais e definibilidade em $\lambda\mathcal{I}$

Neste apêndice enumeramos as definições e alguns resultados sobre a definibilidade de funções recursivas parciais no $\lambda\mathcal{I}$-calculus, que utilizámos durante a prova do teorema 4.1.2. As demonstrações dos resultados apresentados podem ser encontradas em (Barendregt, 1981), (Shoenfield, 1993) e (Rogers, 1967).

Definição A.0.1 *Uma função numérica parcial é uma aplicação parcial*

$$\varphi : \mathbb{N}^p \longrightarrow \mathbb{N}$$

onde $p \in \mathbb{N}$.

As funções numéricas U_i^p, S^+ *e* Z *definidas por*

$$\begin{array}{rrcll} & U_i^p(n_0, \ldots, n_p) & = & n_i & \quad 0 \leq i \leq p, \\ & S^+(n) & = & n+1, & \\ e & Z(n) & = & 0 & \end{array}$$

chamam-se funções iniciais.

Seja $P(n)$ *uma relação numérica. Então denota-se por*

$$\mu m[P(m)]$$

o menor número m para o qual se verifica $P(m)$; se não existir tal número, então $\mu m[P(m)]$ não está definido.

Notação A.0.2 *Habitualmente abrevia-se uma sequência $n_0, \ldots, n_p$ por $\vec{n}$. Se a função φ não estiver definida em $\vec{n}$, então escrevemos $\varphi(\vec{n}) \uparrow$; o caso contrário, i.e. se existir $m \in \mathbb{N}$ tal que $\varphi(\vec{n}) = m$, denota-se por $\varphi(\vec{n}) \downarrow$.*

Sejam $\chi, \psi_1, \ldots, \psi_m$ funções numéricas parciais, onde χ tem m argumentos e $\psi_1, \ldots, \psi_m$ têm todos o mesmo número de argumentos. A expressão

$$\chi(\psi_1(\vec{n}), \ldots, \psi_m(\vec{n}))$$

denota a função numérica parcial φ tal que

$$\varphi(\vec{n}) = a \quad \textit{sse} \quad \exists a_1 \ldots \exists a_m \; [\psi_1(\vec{n}) = a_1, \ldots, \psi_m(\vec{n}) = a_m, \; \chi(a_1, \ldots, a_m) = a].$$

Definição A.0.3 *Seja $\Re$ uma classe de funções numéricas parciais.*

(i) *$\Re$ diz-se fechado para a composição, se para todo φ definido por*

$$\varphi(\vec{n}) = \chi(\psi_1(\vec{n}), \ldots, \psi_m(\vec{n}))$$

com $\chi, \psi_1, \ldots, \psi_m \in \Re$, se tem $\varphi \in \Re$.

(ii) *$\Re$ diz-se fechado para a recursão primitiva, se para todo φ definido por*

$$\begin{aligned} \varphi(0, \vec{n}) &= \chi(\vec{n}), \\ \varphi(k+1, \vec{n}) &= \psi(\varphi(k, \vec{n}), k, \vec{n}) \end{aligned}$$

com $\chi, \psi \in \Re$, se tem $\varphi \in \Re$.

(iii) *$\Re$ diz-se fechado para a minimização, se para todo φ definido por*

$$\varphi(\vec{n}) = \mu m[\chi(\vec{n}, m) = 0]$$

com $\chi \in \Re$, se tem $\varphi \in \Re$.

Definição A.0.4 *A classe $\mathcal{PR}$ das funções recursivas parciais é a menor classe de funções numéricas parciais, que contém as funções iniciais e é fechada para a composição, para a recursão primitiva e para a minimização.*

Definição A.0.5 (Números de Gödel para $\mathcal{PR}$) *Usando técnicas standard é possível definir uma função injectiva $\sharp : \mathcal{PR} \longrightarrow \mathbb{N}$, associando a cada função $\varphi \in \mathcal{PR}$ um inteiro $\sharp(\varphi)$, que se chama o número de Gödel de φ.*

Proposição A.0.6 *Para $\sharp(\varphi) = e$ escreve-se φ como φ_e. Então a função F definida por*

$$\begin{array}{rccc} F: & \mathbb{N} & \longrightarrow & \mathbb{N} \\ & e & \rightsquigarrow & \varphi_e(e) \end{array}$$

é uma função recursiva parcial e é indecidível se $F(e)\downarrow$.

Definição A.0.7 *No $\lambda\mathcal{I}$-calculus definem-se os numerais de Church c_n por*

$$\begin{cases} c_0 & = & \lambda xy.x\mathbf{II}y \\ c_n & = & \lambda fx.f^n(x) \quad (n \geq 1) \end{cases}$$

onde $f^1(x) = f(x)$ e $f^{n+1}(x) = f(f^n(x))$.

Notação A.0.8 *Em geral utilizam-se as abreviaturas seguintes.*

$$\begin{array}{rcl} \llcorner n \lrcorner & = & c_n \\ G\llcorner \vec{n} \lrcorner & = & G\llcorner n_0 \lrcorner \ldots \llcorner n_p \lrcorner \end{array}$$

Lema A.0.9 *Para todo $n \geq 0$, verifica-se*

$$c_n\mathbf{II} \rightarrow_\beta \mathbf{I}.$$

Definição A.0.10 *Uma função númerica parcial* $\varphi : \mathbb{N}^p \longrightarrow \mathbb{N}$ *com* $p \in \mathbb{N}$ *é definível em* $\lambda\mathcal{I}$ *se existir* $G \in \Lambda^0_{\mathcal{I}}$ *tal que*

$$\begin{array}{ll} G\llcorner\vec{n}\lrcorner = \llcorner m\lrcorner & \textit{se } \varphi(\vec{n}) = m, \\ G\llcorner\vec{n}\lrcorner \textit{ não tem forma normal} & \textit{se } \varphi(\vec{n}) \uparrow . \end{array}$$

Neste caso diz-se que em $\lambda\mathcal{I}$ *a função* φ *é definida por* G*.*

Teorema A.0.11 (Kleene) *Uma função númerica parcial é definível em* $\lambda\mathcal{I}$ *se e só se for uma função recursiva parcial.*

Referências

Abdali, S. K. (1976). An abstraction algorithm for combinatory logic. *Journal of Symbolic Logic*, **41**, 222–224.

Anderson, A. R., & Belnap, N. D. (1975). *Entailment. The Logic of Relevance and Necessity, Vol. 1.* Princeton University Press, New Jersey.

Barendregt, H. (1981). *The Lambda Calculus, its Syntax and Semantics.* North-Holland, Amsterdam.

Barendregt, H. (1990). Functional Programming and Lambda Calculus. Pag. 322–63 in: van Leeuwen, J. (ed), *Formal Models and Semantics, Vol. B.* Handbook of Theoretical Computer Science. Elsevier.

Barendregt, H. (1992). Lambda calculi with types. Pag. 117–309 in: Abramsky, Gabbay, & Maibaum (eds), *Background: Computational Structures.* Handbook of Logic in Computer Science, vol. 2. Oxford Science Publications.

Broda, S. (1993). *Some results on the linear λ-calculus.* Research report. LIACC, Universidade do Porto.

Broda, S., & Damas, L. (1995). A new translation algorithm from lambda calculus into combinatory logic. Pag. 359–370 in: Mamede, & Pinto-Ferreira (eds), *7th Portuguese Conference on Artifical Intelligence.* LNAI, vol. 990. Springer Verlag.

Broda, S., & Damas, L. (1996a). *Compact bracket abstraction in combinatory logic.* Aceite para publicação no Journal of Symbolic Logic.

Broda, S., & Damas, L. (1996b). *On combinatory complete sets of proper combinators.* Aceite para publicação no Journal of Functional Programming.

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

Cardelli, Lucas. (1984). Semantics of multiple inheritance. Kahn, MacQueen, & Plotkin (eds), *Semantics of Data Types*. LNCS, vol. 173. Springer Verlag.

Church, A. (1932). A set of postulates for the foundation of logic. *Annals of Math.*, **33(2)**, 346–366.

Church, A. (1933). A set of postulates for the foundation of logic. (Second Paper). *Annals of Maths.*, **34(2)**, 839–864.

Church, A. (1940). A formulation of the simple theory of types. *Journal of Symbolic Logic*, **5**, 56–68.

Church, A. (1941). *The Calculi of Lambda Conversion*. Princeton University Press, Princeton.

Church, A. (1951). The weak theory of implication. Pag. 22–37 in: Menne A., et al. (ed), *Kontrolliertes Denken, Untersuchungen zum Logikkalkül und zur Logik der Einzelwissenschaften*. Kommissionsverlag Karl Albert, München.

Church, A., & Rosser, J. B. (1936). Some properties of conversion. *Trans. Amer. Math. Soc.*, **39**, 472–482.

Curry, H. B. (1929). An analysis of logical substitution. *Amer. J. Math.*, **51**, 363–384.

Curry, H. B. (1930). Grundlagen der kombinatorischen Logik (Inauguraldissertation). *Amer. J. Math.*, **52**, 509–536 and 789–834.

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

Curry, H. B. (1969). Modified basic functionality in combinatory logic. *Dialectica*, **23**, 83–92.

Curry, H. B., & Feys, R. (1958). *Combinatory Logic*. Vol. 1. North-Holland, Amsterdam.

Curry, H. B., Hindley, J. R., & Seldin, J. P. (1972). *Combinatory Logic*. Vol. 2. North-Holland, Amsterdam.

de Bruijn, N. G. (1970). The mathematical language AUTOMATH, its usage and some of its extensions. Pag. 29–61 in: *Symposium on automatic demonstration.* Lecture Notes in Mathematics, vol. 125. Springer Verlag.

de Bruijn, N. G. (1980). A survey of the AUTOMATH project. *In:* (Seldin & Hindley, 1980).

Dowty, D. R., Wall, R. E., & Peters, S. (1985). *Introduction to Montague Semantics.* D. Reidel Publishing Company, Dordrecht.

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

Girard, J.-Y. (1987). Linear Logic. *Theoretical Computer Science*, **50**, 1–102.

Girard, J.-Y., Lafont, Y., & Taylor, P. (1989). *Proofs and Types.* Cambridge University Press.

Guessarian, Irène. (1981). *Algebraic Semantics.* LNCS, vol. 99. Springer Verlag.

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

Harrop, R. (1965). 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.

Helman, G. H. (1977). *Restricted lambda abstraction and the interpretation of some nonclassical logics.* Ph.D. thesis, Philosophy Dept., University of Pittsburgh.

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

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

Hindley, J. R., & Seldin, J. P. (1986). *Introduction to Combinators and λ-Calculus.* Cambridge University Press.

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

Howard, W. A. (1980). The formulae-as-types notion of construction. *In:* (Seldin & Hindley, 1980).

Hugues, J. (1982). Supercombinators, a new implementation method for applicative languages. *Proc. ACM Symp. on Lisp and Functional Programming*, 1–10.

Kennaway, J. R., & Sleep, M. R. (1984). *Counting Director Strings*. University of East Anglia, Norwich.

Kleene, S. C., & Rosser, J. B. (1935). The inconsistency of certain formal logics. *Annals of Maths.*, **36(2)**, 630–636.

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

Kopylov, A. P. 1995 (26–29 June). Decidability of Linear Affine Logic. Pag. 496–504 in: *Proceedings, Tenth Annual IEEE Symposium on Logic in Computer Science*. IEEE Computer Society Press, San Diego, California.

Kreisel, G. (1968). A Survey of Proof Theory. *Journal of Symbolic Logic*, **33**, 321–388.

Krivine, J. L. (1990). *Lambda-calcul, Types et Modèles*. Masson, Paris.

Meredith, C. A. (1978). Positive Logic and λ-constants. *Studia Logica*, **3**, 267–285.

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

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

Moh, Shaw-Kwei. (1950). The deduction theorems and two new logical systems. *Methodos*, **2**, 56–75.

Mulder, J. C. (1985). *Complexity of combinatory code*. Preprint 389. University Utrecht, Department of Mathematics.

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

Peyton Jones, S. L. (1987). *The implementation of functional programming languages.* Prentice-Hall.

Prawitz, D. (1965). *Natural Deduction: A Proof-theoretical Study.* Almqvist and Wiksell, Stockholm.

Rogers, H. (1967). *The Theory of Recursive Functions and Effective Operations.* MacGraw Hill, New York.

Schönfinkel, M. (1924). Über die Bausteine der mathematischen Logik. *Mathematische Annalen*, **92**, 305–316.

Seldin, J. P., & Hindley, J. R. (eds). (1980). *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism.* Academic Press.

Shoenfield, J. R. (1993). *Recursion Theory.* Lecture Notes in Logic, vol. 1. Springer Verlag.

Stanley, R. P. (1983). *Combinatorics and Commutative Algebra.* Vol. 41. Birkhäuser. Progress in Mathematics.

Statman, R. (1983). *An optimal translation of λ-terms into combinators.* Manuscript, Dept. of Computer Science, Carnegie-Mellon Univ., Pittsburgh, PA.

Statman, R. (1986). On Translating Lambda Terms into Combinators; The Basis Problem. Pag. 378–382 in: *Proceedings, First Annual IEEE Symposium on Logic in Computer Science.* IEEE.

Tait, W. W. (1965). Infinitely long terms of transfinite type. Crossley, J. N., & Dummett, M. A. E. (eds), *Formal Systems and Recursive Functions.* North-Holland, Amsterdam.

Trigg, P., Hindley, J. R., & Bunder, M. W. (1994). Combinatory abstraction using B, B' and friends. *Theoretical Computer Science*, **135**, 405–422.

Troelstra, A. S. (1992). Lectures on Linear Logic. Lecture Notes, vol. 29. CSLI.

Turner, D. A. (1979a). Another Algorithm for Bracket Abstraction. *Journal of Symbolic Logic*, **44**, 267–270.

Turner, D. A. (1979b). A new implementation technique for applicative languages. *Software-Practice and Experience*, **9**, 31–49.

Ungar, A. M. (1992). Normalization, Cut-elimination and the Theory of Proofs. Lecture Notes, vol. 28. CSLI.

van Benthem, J. (1991). *Language in Action: Categories, Lambdas and Dynamic Logic*. North-Holland, Amsterdam.

Wadler, P. (1993). A taste of linear logic. Pag. 185–210 in: *Mathematical Foundations of Computer Science, Gdansk, Poland*. LNCS, vol. 711. Springer Verlag.