Cálculo Lambda Tipado

Prof. Marcus Vinícius Midena Ramos

Universidade Federal do Vale do São Francisco

30 de março de 2019

marcus.ramos@univasf.edu.br
www.univasf.edu.br/~marcus.ramos

Bibliografia

1. *Type Theory and Formal Proof - An Introduction*
   Rob Nederpelt and Herman Geuvers
   Cambridge University Press, 2014
   Capítulo 2
2. *Lambda-Calculus and Combinators - An Introduction*
   J. R. Hindley and J. P. Seldin
   Cambridge University Press, 2008
   Capítulos 10 e 12
3. `https://en.wikipedia.org/wiki/Typed_lambda_calculus`
4. `https://en.wikipedia.org/wiki/Simply_typed_lambda_calculus`

Roteiro

Cálculo Lambda Não-Tipado

Aspectos positivos:

- Simples, conciso e elegante;
- Confluência e unicidade de formas normais β;
- Resolução de equações recursivas por meio de pontos fixos;
- Turing-completo.

Cálculo Lambda Não-Tipado

Aspectos negativos:

- Excessivamente liberal (contra-intuitivo em alguns aspectos);
- Auto-aplicação é permitida (xx) porém é contra-intuitiva;
- A existência de formas normais não é garantida, existe a possibilidade de cálculos infinitos;
- Todo termo possui um ponto fixo, o que não corresponde ao que se conhece do comportamento usual das funções (algumas possuem, outras não).

Cálculo Lambda Tipado

Solução para preservar os aspectos positivos eliminando os aspectos negativos:

- Introdução da noção de tipos;
- Um tipo restringe a coleção de valores que podem ser fornecidos ou retornados de uma função;
- Serve para prevenir as anomalias citadas anteriormente;
- Não é uma ideia original de Church, uma teoria de tipos foi formulada anteriormente por Bertrand Russell; a ideia de Church é uma simplificação da teoria de Russell.

Cálculo Lambda Tipado

Conseqüências

- Elimina os problemas do Cálculo Lambda Não-Tipado:
  - A auto-aplicação não é possível (termos com auto-aplicação não são válidos no Cálculo Lambda Tipado);
  - Todo termo possui uma forma normal β (toda computação é finita);
  - Nem todo termo possui um ponto fixo.
- Os aspectos positivos do Cálculo Lambda Não-Tipado são preservados no Cálculo Lambda Tipado.

Cálculo Lambda Tipado

Conseqüências

- É mais limitado que o Cálculo Lambda Não-Tipado e possui poder reduzido (pois nem todas as funções computáveis podem ser representadas no Cálculo Lamba Tipado);
- Não se presta para a formalização da matemática, como era o objetivo de Church, mas possui diversas aplicações importantes na computação;
- É realizado em uma série de sistemas, com complexidade crescente; o mais simples é o chamado "Simply Typed Lambda Calculus" ($\lambda \rightarrow$); o mais complexo é o chamado "Calculus of Constructions" (λC);
- O "Cubo de Barendregt" representa a relação que existe entre todos estes sistemas;

Cubo de Barendregt

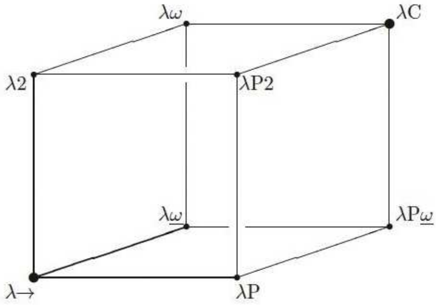
Descoberto e descrito por Henk Barendregt em 1992.
Possui quatro sistemas independentes:

1. $\lambda \rightarrow$ Simply Typed Lambda Calculus (STLC);
2. $\lambda 2$ Second Order Typed Lambda Calculus (STLC + Terms Dependent on Types);
3. $\lambda \underline{w}$ (STLC + Types Dependent on Types);
4. λP (STLC + Types Dependent on Terms);

e quatro combinações:

5. λw (STLC + Terms Dependent on Types + Types Dependent on Types);
6. $\lambda P2$ (STLC + Types Dependent on Terms + Terms Dependent on Types);
7. $\lambda P\underline{w}$ (STLC + Types Dependent on Terms + Types Dependent on Types);
8. λC Calculus of Constructions (STLC + Terms Dependent on Types + Types Dependent on Terms + Types Dependent on Types).

Cubo de Barendregt



Cálculo Lambda Tipado

Conseqüências

- Todos estes sistemas apresentam as boas características do Cálculo Lambda Tipado e preservam as vantagens do Cálculo Lambda Não-Tipado;
- Eles possuem poder crescente e se prestam, no caso geral, para usos diversos e relevantes na lógica e na matemática;
- Os aspectos negativos do Cálculo Lambda Não-Tipado estão ausentes de todos eles.

Calculus of Constructions with Definitions

Além do Cubo de Barendregt:

- Um sistema tipado ainda mais poderoso do que o λC (Calculus of Constructions), incorpora o uso definições, em particular definições indutivas;
- Este sistema, denotado λD, é o sistema utilizado no assistente de provas Coq, o que lhe confere grande poder e flexibilidade.

Tipos Simples

Seja $\mathbb{V} = \{\alpha, \beta, \gamma, ...\}$ um conjunto infinito de identificadores de tipo ("type variables"). Cada identificador de tipo representa um tipo primitivo ou básico (como `nat`, `list`, `int` ou `float`). O conjunto $\mathbb{T}$ de todos os tipos simples é definido como:

- Se $\alpha \in \mathbb{V}$ então $\alpha \in \mathbb{T}$;
- Se $\sigma, \tau \in \mathbb{T}$ então $(\sigma \rightarrow \tau) \in \mathbb{T}$

Tipos Simples

Exemplos de tipos simples:

- γ
- $(\beta \to \gamma)$
- $(\alpha \to (\beta \to \gamma))$
- $((\alpha \to \beta) \to \gamma)$
- $((\alpha \to \tau) \to (\beta \to \gamma))$

O “Simply Typed Lambda Calculus” é um Cálculo Lambda com um único construtor de tipos, o $\to$. Conhecido como “function” ou “arrow type”, ele é usado para representar o tipo das funções.

Associatividade

Para evitar o uso execessivo e desnecessário de parênteses, a seguinte convenção será adotada:

- O tipo função é associativo à direita;
- A aplicação continua associativa à esquerda.

Desta forma,

- $\alpha_1 \to \alpha_2 \to \alpha_3 \to \alpha_4$ denota $(\alpha_1 \to (\alpha_2 \to (\alpha_3 \to \alpha_4)))$;
- $x_1 x_2 x_3 x_4$ denota $(((x_1 x_2) x_3) x_4)$.

Estas convenções se encaixam naturalmente com as regras de inferência de tipos que serão vistas a seguir.

“Typing Statement”

A notação

$$M : \sigma$$

é usada para denotar o fato de que o termo lambda M possui o tipo σ.

- Admite-se que cada identificador de tipo pode ser atribuído para uma quantidade ilimitada de variáveis;
- Considera-se que cada variável possui um único tipo (se $x : \sigma$ e $x : \tau$ então $\sigma \equiv \tau$).

Um termo M é dito tipável se existir um tipo σ tal que $M : \sigma$.

Regras de Inferência

Para inferir o tipo de termos lambda do STLC no caso geral, é necessário adotar as seguintes regras que expressam, de forma natural, a relação entre os tipos dos elementos componentes e o tipo do termo resultante:

- Aplicação: se $M : \sigma \rightarrow \tau$ e $N : \sigma$ então $MN : \tau$;
- Abstração: se $x : \sigma$ e $M : \tau$ então $\lambda x.M : \sigma \rightarrow \tau$.

Exemplos

1. Se $x : \sigma$, então $\lambda x.x : \sigma \rightarrow \sigma$;
2. Se $x : \alpha$, $y : \beta$ e $z : \gamma$, então $\lambda x.\lambda y.z : \alpha \rightarrow \beta \rightarrow \gamma$;
3. A aplicação yx só pode ser tipada se y for do tipo função (por exemplo, $\sigma \rightarrow \tau$) e o tipo de x corresponder ao tipo do domínio de y (ou seja, σ). Se este for o caso, então $yx : \tau$;
4. A auto-aplicação não pode ser tipada (por exemplo xx). Por quê?

Associatividade

As associatividades descritas anteriormente não são acidentais. Elas se complementam de maneira a tornar natural os "typing statements" das abstrações e das aplicações correspondentes.
Por exemplo, considere $f : (\rho \to (\sigma \to \tau))$, $x : \rho$ e $y : \sigma$. Então:

- $f : \rho \to \sigma \to \tau$ e
- $fxy : \tau$

(notar que não foram usados parênteses em nenhum dos casos acima)

Tipagem de um termo

Existem duas formas diferentes para se determinar o tipo de um termo lambda:

- Tipagem no estilo de Church (ou tipagem explícita):
  - Os tipos das variáveis são definidos inicialmente;
  - Os demais tipos podem ser calculados sem dificuldade.
- Tipagem no estilo de Curry (tipagem implícita):
  - Não atribui tipos para as variáveis inicialmente;
  - Os tipos são determinados conforme a análise do termo e subtermos.

Estilo de Church

Exemplos:

- Suponha $x : \alpha \to \alpha$ e $y : (\alpha \to \alpha) \to \beta$;
  Então, $yx : \beta$;
  Suponha, ainda, $z : \beta$ e $u : \gamma$;
  Então, $\lambda zu.z : \beta \to \gamma \to \beta$;
  Continuando, a aplicação $(\lambda zu.z)(yx)$ é permitida e
  $(\lambda zu.z)(yx) : \gamma \to \beta$;
- Suponha que $x : \alpha \to \beta \to \gamma \to \delta$, $y : \alpha$ e $z : \beta$;
  Então, $xyz : \gamma \to \delta$.

Todos os termos foram tipados, partindo-se dos tipos inicialmente informados para as variáveis x, y, z e u.

Estilo de Curry

Exemplos:
(desta vez os tipos das variáveis não serão informados à priori)

- Seja $M \equiv (\lambda xy.xy)$;
  Então, uma possível atribuição de tipos seria $x : \alpha \to \beta$, $y : \alpha$ e
  $M : (\alpha \to \beta) \to \alpha \to \beta$
  Ainda, pode-se considerar $x : \alpha \to \beta \to \alpha$, $y : \alpha$ e
  $M : (\alpha \to \beta \to \alpha) \to \alpha \to (\beta \to \alpha)$;
  Existem várias possibilidades;
- Seja $M \equiv (\lambda zu.z)(yx)$;
  Como atribuir tipos para x, y, z, u e M?
  Será necessário realizar inferências sobre a forma como variáveis e subtermos são usados;
  Detalhes no próximo slide.

Estilo de Curry

$M \equiv (\lambda zu.z)(yx)$

Inferências iniciais:

1. Como se trata de uma aplicação, então devemos supor $(\lambda zu.z) : A \to B$ e $(yx) : A$, para algum A e algum B; portanto, $M : B$;
2. $(\lambda zu.z) : A \to B$ implica $z : A$ e $\lambda u.z : B$; como este último é uma abstração, então devem existir C e D tais que $u : C$, $z : D$, $\lambda u.z : C \to D$ e, portanto, $B \equiv (C \to D)$;
3. Além disso, (yx) também é uma aplicação; logo, devem existir E e F tais que $y : E \to F$, $x : E$ e $(yx) : F$.

Estilo de Curry

$M \equiv (\lambda zu.z)(yx)$

Conclusões até o momento:

1. $x : E$,
2. $y : E \to F$,
3. $z : A$ e $z : D$; portanto, $A \equiv D$,
4. $u : C$,
5. $B \equiv (C \to D)$,
6. $(yx) : A$ e $(yx : F)$; portanto, $A \equiv D \equiv F$;
7. $M : C \to A$ (substituindo-se B por $C \to D$ e D por A).

Estilo de Curry

$M \equiv (\lambda zu.z)(yx)$

Em resumo:

1. $x : E$,
2. $y : E \to A$,
3. $z : A$,
4. $u : C$,
5. $(yx) : A$,
6. $M : C \to A$.

Logo, apenas três tipos são necessários (A, C e E). Mas quais tipos?

Estilo de Curry

$M \equiv (\lambda zu.z)(yx)$

Como A, C e E denotam tipos genéricos, podemos instanciar estas variáveis de forma a obter diversas atribuições de tipos. Algumas possibilidades são apresentadas a seguir:

1. Se $A = \alpha, C = \delta$ e $E = \beta$, então:
   $x : \beta, y : \beta \to \alpha, z : \alpha, u : \delta,\ M : \delta \to \alpha$;
2. Se $A = \beta, C = \gamma$ e $E = \alpha \to \alpha$, então:
   $x : \alpha \to \alpha, y : (\alpha \to \alpha) \to \beta, z : \beta, u : \gamma,\ M : \gamma \to \beta$;
   (esta atribuição coincide com o exemplo anterior no estilo de Church)
3. Se $A = \alpha \to \beta, C = \alpha \to \alpha$ e $E = \alpha$, então:
   $x : \alpha, y : \alpha \to \alpha \to \beta, z : \alpha \to \beta, u : \alpha \to \alpha,\ M : (\alpha \to \alpha) \to \alpha \to \beta$.

De qualquer forma, existe uma atribuição de tipos e M é tipável.

Julgamento e Contexto

Uma fórmula do tipo $\Gamma \vdash e : t$ denota um julgamento.

- A parte Γ que está do lado esquerdo do símbolo $\vdash$ é chamada de contexto, e contém as declarações e os tipos das variáveis livres usadas na expressão lambda da direita (e);
- A parte que está do lado direito do símbolo $\vdash$ é um termo lambda com o seu respectivo tipo $(e : t)$.

Por convenção, em um termo lambda os tipos das variáveis ligadas são indicados nas respectivas declarações, ao passo que os tipos das variáveis livres são indicados no contexto. Diz-se que "e tem o tipo t no contexto Γ".

Exemplo

Considere novamente o termo $(\lambda zu.z)(yx)$. Neste termo, as variáveis z e u são ligadas e x e y são livres. Considerando que os tipos sejam $z:\beta, u:\gamma, x:\alpha\rightarrow\alpha$ e $y:(\alpha\rightarrow\alpha)\rightarrow\beta$, então a anotação dos tipos das variáveis neste termos pode ser representada como:

$$x:\alpha\rightarrow\alpha, y:(\alpha\rightarrow\alpha)\rightarrow\beta\vdash(\lambda z:\beta.\lambda u:\gamma.z)(yx):\gamma\rightarrow\beta$$

Esta expressão pode ser lida como “no contexto:

$$x:\alpha\rightarrow\alpha, y:(\alpha\rightarrow\alpha)\rightarrow\beta$$

o termo:

$$(\lambda z:\beta.\lambda u:\gamma.z)(yx)$$

possui o tipo:

$$\gamma\rightarrow\beta$$

Notações alternativas

Alguns autores (como por exemplo Hindley e Seldin 2008) utilizam notações alternativas, com o uso de superescritos, para fazer a anotação de tipos em um termo lambda. As representações são, no entanto, equivalentes:

- $x : \alpha \to \alpha, y : (\alpha \to \alpha) \to \beta \vdash (\lambda z : \beta.\lambda u : \gamma.z)(yx) : \gamma \to \beta$
- $((\lambda z^{\beta}.\lambda u^{\gamma}.z^{\beta})^{\beta\to\gamma\to\beta}(y^{(\alpha\to\alpha)\to\beta}x^{\alpha\to\alpha})^{\beta})^{\gamma\to\beta}$

Algumas simplificações são admitidas quando não houver risco de ambigüidade:

- $((\lambda z^{\beta}.\lambda u^{\gamma}.z)^{\beta\to\gamma\to\beta}(y^{(\alpha\to\alpha)\to\beta}x^{\alpha\to\alpha})^{\beta})^{\gamma\to\beta}$
- $((\lambda z.\lambda u.z)^{\beta\to\gamma\to\beta}(y^{(\alpha\to\alpha)\to\beta}x^{\alpha\to\alpha})^{\beta})^{\gamma\to\beta}$
- $((\lambda z.\lambda u.z)^{\beta\to\gamma\to\beta}(y^{(\alpha\to\alpha)\to\beta}x^{\alpha\to\alpha}))^{\gamma\to\beta}$

Terminologia

- Uma afirmação ("statement") possui a forma $M : \sigma$ onde M é um termo lambda tipado e σ é um tipo simples; numa afirmação, M é chamado de sujeito ("subject") e σ de tipo ("type");
- Uma declaração ("declaration") é uma afirmação em que o sujeito é uma variável;
- Um contexto ("context") é uma lista de declarações com diferentes sujeitos;
- Um julgamento ("judgement") possui a forma $\Gamma \vdash M : \sigma$, onde Γ é um contexto e $M : \sigma$ é uma afirmação.

Terminologia

Exemplos:

- $(\lambda x : \alpha . y) : \beta$ é uma afirmação; $(\lambda x : \alpha . y)$ é o sujeito e β é o tipo;
- $(\lambda x : \alpha . y) : \beta$ não é uma declaração pois o sujeito não é uma variável;
- $x : \alpha$ é uma declaração pois o sujeito é uma variável;
- $x_1 : \alpha, x_2 : \alpha \to \beta, x_3 : (\beta \to \alpha) \to \beta$ é um contexto formado por três variáveis;
- $x : \alpha \to \alpha, y : (\alpha \to \alpha) \to \beta \vdash (\lambda z : \beta . \lambda u : \gamma . z)(yx) : \gamma \to \beta$ é um julgamento.

Sistema de Derivação

- Como determinar se um termo é tipável?
  É simples e foi antecipado.
- Como determinar o tipo de um termo?
  É simples também. Há uma única resposta do estilo de Church e múltiplas respostas no estilo de Curry.

Um conjunto de regras formais denominado Sistema de Derivação é chave neste processo. Por meio de um Sistema de Derivação é possível determinar se o julgamento $\Gamma \vdash M : \sigma$ é derivável, ou seja, se M possui o tipo σ no contexto Γ.

Sistema de Derivação

Formato geral de uma regra:

$$(nome)\ \frac{\text{premissa 1} \qquad \ldots \qquad \text{premissa n}}{\text{conclusão}}$$

- Se as premissas são todas verdadeiras, então a conclusão pode ser aceita;
- A linha horizontal separa as premissas das conclusões;
- Múltiplas regras podem ser combinadas em uma árvore;
- Podem ser usadas de cima para baixo ou de baixo para cima.

Sistema de Derivação

O sistema é composto por três regras:

$$(\text{var})\ \frac{x:\sigma\in\Gamma}{\Gamma\vdash x:\sigma}$$

$$(appl)\ \frac{\Gamma\vdash M:\sigma\to\tau \qquad \Gamma\vdash N:\sigma}{\Gamma\vdash MN:\tau}$$

$$(abst)\ \frac{\Gamma,x:\sigma\vdash M:\tau}{\Gamma\vdash\lambda x:\sigma.M:\sigma\to\tau}$$

Sistema de Derivação

Exemplo:

$$
(abst)\;\dfrac{(abst)\;\dfrac{(appl)\;\dfrac{(\mathsf{var})\;\dfrac{y:\alpha\to\beta\in(y:\alpha\to\beta, z:\alpha)}{y:\alpha\to\beta, z:\alpha\vdash y:\alpha\to\beta}\qquad(\mathsf{var})\;\dfrac{z:\alpha\in(y:\alpha\to\beta, z:\alpha)}{y:\alpha\to\beta, z:\alpha\vdash z:\alpha}}{y:\alpha\to\beta, z:\alpha\vdash yz:\beta}}{y:\alpha\to\beta\vdash\lambda z:\alpha.yz:\alpha\to\beta}}{\emptyset\vdash\lambda y:\alpha\to\beta.\lambda z:\alpha.yz:(\alpha\to\beta)\to\alpha\to\beta}
$$

Sistema de Derivação

A derivação do exemplo anterior, feita com o uso exclusivo das regras do sistema apresentado, serve simultaneamente para:

- Construir o julgamento;
- Justificar o julgamento.

Ainda no exemplo anterior, diz-se que, no contexto vazio, o termo:

$$\lambda y : \alpha \to \beta . \lambda z : \alpha . yz$$

possui o tipo:

$$(\alpha \to \beta) \to \alpha \to \beta$$

- Termos que são tipáveis com o auxílio de um sistema de derivação são ditos legais.

Relação entre $\rightarrow$ e $\Rightarrow$

- Deve-se notar a similaridade que existe entre as regras do STLC e as regras da Dedução Natural;
- A regra $appl$ está em correspondência direta com a regra *Modus Ponens* (eliminação da implicação);
- A regra $abst$ está em correspondência direta com a *Regra da Prova Condicional* (introdução da implicação);
- Por isso, existe uma conexão muito forte entre o tipo função $\rightarrow$ (no STLC) e a implicação $\Rightarrow$ (na Dedução Natural); esta conexão será explorada mais adiante.

Tipos de Problemas

São de três naturezas diferentes os problemas encontrados numa teoria de tipos:

1. Tipabilidade ("well-typedness" ou "typability"): $? \vdash M : ?$
   Determinar se existe um contexto e um tipo tais que o termo seja legal, ou provar o contrário.
   - Atribuição de Tipos ("type assignment"): É uma variante que ocorre quando o contexto é dado, restando apenas determinar o tipo: $\Gamma \vdash M : ?$
2. Verificação de Tipos ("type checking"): $\Gamma \overset{?}{\vdash} M : \sigma$
   Simplesmente verificar que o termo M possui o tipo σ no contexto Γ.
3. Construção de Termo ("term finding", "term construction" ou "inhabitation"): $\Gamma \vdash ? : \sigma$
   Encontrar um termo M que possua o tipo σ no contexto Γ ou provar que o mesmo não existe. Caso particular $\emptyset \vdash ? : \sigma$.

Tipos de Problemas

- A Tipabilidade e a Atribuição de Tipos são problemas simples e de fácil solução, usando o Sistema de Derivações e inferências adequadas;
- A Construção de Termo, ou seja, encontrar um termo cujo tipo seja σ, equivale ao problema de terminar se existe uma prova para σ; a solução pode não ser trivial;
- Todos os problemas são decidíveis no STLC;
- Nos sistemas mais elaborados, no entanto, a Construção de Termos é indecidível em muitos casos.

Tipabilidade

Exemplo:
Determinar um contexto e um tipo para o termo

$$M \equiv \lambda y : \alpha \to \beta . \lambda z : \alpha . yz$$

Em outras palavras, é possível afirmar que existe um contexto Γ e um tipo ρ tal que $\Gamma \vdash M : \rho$? Em caso afirmativo, determinar Γ e ρ.

- Como M possui apenas variáveis ligadas, podemos considerar $\Gamma \equiv \emptyset$, uma vez que o contexto é usado apenas para atribuir tipos para as variáveis livres de M.

Tipabilidade

Passo 1 (sem contexto):

$$(abst)\ \frac{y : \alpha \to \beta \qquad \lambda z : \alpha . yz : \ldots}{\lambda y : \alpha \to \beta . \lambda z : \alpha . yz : \ldots}$$

Tipabilidade

Passo 2 (sem contexto):

$$
(abst)\ \dfrac{y : \alpha \to \beta \qquad (abst)\ \dfrac{z : \alpha \qquad yz : ...}{\lambda z : \alpha . yz : ...}}{\lambda y : \alpha \to \beta . \lambda z : \alpha . yz : ...}
$$

Tipabilidade

Passo 3 (sem contexto):

$$(abst)\,\dfrac{y:\alpha\to\beta \qquad (abst)\,\dfrac{z:\alpha \qquad (appl)\,\dfrac{y:\ldots \qquad z:\ldots}{yz:\ldots}}{\lambda z:\alpha.yz:\ldots}}{\lambda y:\alpha\to\beta.\lambda z:\alpha.yz:\ldots}$$

Tipabilidade

Passo 4 (final, sem contexto):

$$\dfrac{y:\alpha\to\beta \qquad \dfrac{z:\alpha \qquad \dfrac{y:\alpha\to\beta \qquad z:\alpha}{yz:\beta}\,(appl)}{\lambda z:\alpha.yz:\alpha\to\beta}\,(abst)}{\lambda y:\alpha\to\beta.\lambda z:\alpha.yz:(\alpha\to\beta)\to\alpha\to\beta}\,(abst)$$

Tipabilidade

Passo 4 (completo, parte i):

$$
(appl)\,\dfrac{(var)\,\dfrac{y:\alpha\to\beta\in(y:\alpha\to\beta)}{y:\alpha\to\beta\vdash y:\alpha\to\beta}\qquad (var)\,\dfrac{z:\alpha\in(z:\alpha)}{z:\alpha\vdash z:\alpha}}{y:\alpha\to\beta, z:\alpha\vdash yz:\beta}
$$

Tipabilidade

Passo 4 (completo, parte ii):

$$(abst)\ \frac{z : \alpha \vdash z : \alpha \qquad \boxed{y : \alpha \to \beta, z : \alpha \vdash yz : \beta}}{\boxed{y : \alpha \to \beta \vdash \lambda z : \alpha.yz : \alpha \to \beta}}$$

Tipabilidade

Passo 4 (completo, parte iii):

$$\dfrac{(var)\ \dfrac{y:\alpha\to\beta \in (y:\alpha\to\beta)}{y:\alpha\to\beta\vdash y:\alpha\to\beta} \qquad y:\alpha\to\beta\vdash\lambda z:\alpha.yz:\alpha\to\beta}{\emptyset\vdash\lambda y:\alpha\to\beta.\lambda z:\alpha.yz:(\alpha\to\beta)\to\alpha\to\beta}\ (abst)$$

Verificação de Tipos

Exemplo:
Determinar se o seguinte julgamento é legal:

$$x : \alpha \to \alpha, y : (\alpha \to \alpha) \to \beta \vdash (\lambda z : \beta.\lambda u : \gamma.z)(yx) : \gamma \to \beta$$

Verificação de Tipos

Passo 1 (sem contexto):

$$(appl)\ \frac{(\lambda z : \beta.\lambda u : \gamma.z) : ... \qquad (yx) : ...}{(\lambda z : \beta.\lambda u : \gamma.z)(yx) : \gamma \to \beta}$$

Verificação de Tipos

Passo 2 (sem contexto):

$$
(appl)\dfrac{(abst)\dfrac{z:\beta \qquad \lambda u:\gamma.z:...}{(\lambda z:\beta.\lambda u:\gamma.z):...} \qquad (yx):...}{(\lambda z:\beta.\lambda u:\gamma.z)(yx):\gamma\to\beta}
$$

Verificação de Tipos

Passo 3 (sem contexto):

$$
(appl)\,\dfrac{(abst)\,\dfrac{z:\beta \qquad (abst)\,\dfrac{u:\gamma \qquad z:...}{\lambda u:\gamma.z:...}}{(\lambda z:\beta.\lambda u:\gamma.z):...} \qquad (yx):...}{(\lambda z:\beta.\lambda u:\gamma.z)(yx):\gamma\rightarrow\beta}
$$

Verificação de Tipos

Passo 4 (sem contexto):

$$
(appl)\ \dfrac{(abst)\ \dfrac{z:\beta \qquad (abst)\ \dfrac{u:\gamma \qquad z:...}{\lambda u:\gamma.z:...}}{(\lambda z:\beta.\lambda u:\gamma.z):...} \qquad (appl)\ \dfrac{y:... \qquad x:...}{(yx):...}}{(\lambda z:\beta.\lambda u:\gamma.z)(yx):\gamma\rightarrow\beta}
$$

Verificação de Tipos

Passo 5 (final, sem contexto):

$$(appl)\ \frac{y : (\alpha \to \alpha) \to \beta \qquad x : \alpha \to \alpha}{\boxed{(yx) : \beta}}$$

$$(appl)\ \frac{(abst)\ \dfrac{z : \beta \qquad (abst)\ \dfrac{u : \gamma \qquad z : \beta}{\lambda u : \gamma . z : \gamma \to \beta}}{(\lambda z : \beta . \lambda u : \gamma . z) : \beta \to \gamma \to \beta} \qquad \boxed{(yx) : \beta}}{(\lambda z : \beta . \lambda u : \gamma . z)(yx) : \gamma \to \beta}$$

Construção de Termo

Exemplo:
Seja $\sigma \equiv A \to B \to A$. Determinar se existe um termo M tal que $\emptyset \vdash M : \sigma$.

- Um termo que possui determinado tipo é dito um habitante ("inhabitant") daquele tipo;
- Logo,o problema da Construção de Termo é determinar se um tipo é habitado.

Construção de Termo

Passo 1 (sem contexto):

$$(abst)\ \frac{...:A \qquad ...:B\rightarrow A}{...:A\rightarrow B\rightarrow A}$$

Construção de Termo

Passo 2 (sem contexto):

$$(abst)\ \dfrac{...:A \qquad (abst)\ \dfrac{...:B \qquad ...:A}{...:B \to A}}{...:A \to B \to A}$$

Construção de Termo

Passo 3 (final, sem contexto):

$$
(abst)\,\dfrac{x : A \qquad (abst)\,\dfrac{y : B \qquad x : A}{\lambda y : B.x : B \to A}}{\lambda x : A.\lambda y : B.x : A \to B \to A}
$$

Relação entre $\rightarrow$ e $\Rightarrow$

Considere o tipo $A \rightarrow B \rightarrow A$ e a proposição $A \Rightarrow B \Rightarrow A$:

- O termo que possui o tipo $A \rightarrow B \rightarrow A$ $(\lambda x : A.\lambda y : B.x)$ é também uma prova da proposição $A \Rightarrow B \Rightarrow A$, pois a árvore de dedução natural da proposição pode ser facilmente obtida a partir da árvore de inferência de tipos do termo;
- A função que aceita como argumentos uma prova de A e uma prova de B, e retorna como resultado a prova de A, é uma prova de que a proposição $A \Rightarrow B \Rightarrow A$ é uma tautologia;
- Interpretação PAT ("propositions-as-types" ou "proofs-as-terms");
- Correspondência de Curry-Howard;
- O termo lambda tipado (no caso, $\lambda x : A.\lambda y : B.x$) codifica, simultaneamente, uma proposição $(A \Rightarrow B \Rightarrow A)$ e a respectiva prova (inferida a partir da árvore de derivação do termo em questão).

Relação entre $\rightarrow$ e $\Rightarrow$

Senão, vejamos novamente o exemplo anterior.
A partir do termo tipado (e apenas dele) $\lambda x : A.\lambda y : B.x$ é possível inferir:

- O tipo do mesmo:

$$A \rightarrow B \rightarrow A$$

- A árvore de inferência de tipos:

$$(abst)\ \dfrac{x : A \qquad (abst)\ \dfrac{y : B \qquad x : A}{\lambda y : B.x : B \rightarrow A}}{\lambda x : A.\lambda y : B.x : A \rightarrow B \rightarrow A}$$

Relação entre $\rightarrow$ e $\Rightarrow$

- A substituição de $\rightarrow$ por $\Rightarrow$ resulta na proposição que o termo prova:

$$A \Rightarrow B \Rightarrow A$$

- A eliminação dos termos da árvore resulta na prova da proposição (em termos da Dedução Natural):

$$(abst)\ \dfrac{A \qquad (abst)\ \dfrac{B \qquad A}{B \Rightarrow A}}{A \Rightarrow B \Rightarrow A}$$

Definições

1. Domínio Se $\Gamma \equiv x_1 : \sigma_1, ..., x_n : \sigma_n$, então o domínio$(\Gamma)$, ou simplesmente $\mathtt{dom}(\Gamma)$, é a lista $x_1, ..., x_n$;
2. Subcontexto Γ' é um subcontexto de um contexto Γ, ou $\Gamma' \subseteq \Gamma$, se todas as declarações que ocorrem em Γ' também ocorrem em Γ, na mesma ordem;
3. Permutação Γ' é uma permutação de um contexto Γ se todas as declarações de Γ' ocorrem em Γ e vice-versa;
4. Projeção Se Γ é um contexto e ϕ é um conjunto de variáveis, então a projeção de Γ sobre ϕ , ou $\Gamma \upharpoonright \phi$, é o subcontexto Γ' de Γ com $\mathtt{dom}(\Gamma')=\mathtt{dom}(\Gamma) \cap \phi$.

Definições

Exemplo.
Seja $\Gamma \equiv y : \sigma, x_1 : \rho_1, x_2 : \rho_2, z : \tau, x_3 : \rho_3$.
Então:

1. Domínio $\texttt{dom}(\emptyset) = ()$; $\texttt{dom}(\Gamma) = (y, x_1, x_2, z, x_3)$;
2. Subcontexto $\emptyset \subseteq (x_1 : \rho_1, z : \tau) \subseteq \Gamma$;
3. Permutação $x_2 : \rho_2, x_1 : \rho_1, z : \tau, x_3 : \rho_3, y : \sigma$ é uma permutação de Γ;
4. Projeção $\Gamma \upharpoonright \{z, u, x_1\} = x_1 : \rho_1, z : \tau$.

Lemas e Teoremas

- A seguir são relacionados uma série de resultados fundamentais sobre o STLC;
- Os enunciados são intuitivos, na sua maioria, e as provas não serão apresentadas;
- As provas podem ser encontradas em Nederpelt and Geuvers 2014.

Lema das Variáveis Livres

"Free Variables Lemma"

$$\text{Se } \Gamma \vdash L : \sigma, \text{ então } FV(L) \subseteq \texttt{dom}(\Gamma)$$

Alargamento, Estreitamento e Permutação

1. Alargamento ("Thinning")
   Sejam Γ' e Γ'' dois contextos tais que $\Gamma' \subseteq \Gamma''$. Se $\Gamma' \vdash M : \sigma$, então $\Gamma'' \vdash M : \sigma$;
2. Estreitamento ("Condensing")
   Se $\Gamma \vdash M : \sigma$, então $\Gamma \restriction FV(M) \vdash M : \sigma$;
3. Permutação ("Permutation")
   Se $\Gamma \vdash M : \sigma$, e Γ' é uma permutação de Γ, então $\Gamma' \vdash M : \sigma$.

Lema da Geração

"Generation Lemma"

1. Se $\Gamma \vdash x : \sigma$, então $x : \sigma \in \Gamma$;
2. Se $\Gamma \vdash MN : \tau$, então existe um tipo σ tal que $\Gamma \vdash M : \sigma \to \tau$ e $\Gamma \vdash N : \sigma$;
3. Se $\Gamma \vdash \lambda x : \sigma.M : \rho$, então existe um tipo τ tal que $\Gamma, x : \sigma \vdash M : \tau$ e $\rho \equiv \sigma \to \tau$.

Lema do Subtermo

"Subterm Lemma"

Se o termo M é legal, então todo subtermo de M também é legal.

Unicidade de Tipos

"Uniqueness of Types"

Se $\Gamma \vdash M : \sigma$ e $\Gamma \vdash M : \tau$, então $\sigma \equiv \tau$.

Decidibilidade

No STLC os seguintes problemas são decidíveis:

1. Tipabilidade ("well-typedness") $? \vdash M : ?$
   - Atribuição de Tipos ("type assignment") $\Gamma \vdash M : ?$
2. Verificação de Tipos ("type checking") $\Gamma \overset{?}{\vdash} M : \sigma$
3. Construção de Termo ("term finding") $\Gamma \vdash ? : \sigma$

Substituição no Cálculo Lambda Não-Tipado

(1a) $x[x := N] \equiv N$;

(1b) $y[x := N] \equiv y$, se $x \not\equiv y$;

(2) $(PQ)[x := N] \equiv (P[x := N])(Q[x := N])$;

(3) $(\lambda y.P)[x := N] \equiv \lambda z.(P^{y \to z}[x := N])$, se $\lambda z.P^{y \to z}$ for um α-variante de $\lambda y.P$ tal que $z \notin FV(N)$

Observação:
O termo $\lambda z.P^{y \to z}$ é uma conversão-α do termo $\lambda y.P$ em que a variável y é substituída pela variável z. Ou seja, $\lambda z.P^{y \to z} \equiv_\alpha \lambda y.P$.

Substituição no Cálculo Lambda Não-Tipado

Observe-se a similaridade entre a definição de *Hindley & Seldin* com a definição de *Nederpelt & Geuvers*:

1. (1a) corresponde ao caso (a);
2. (1b) corresponde ao caso (b);
3. (2) corresponde ao caso (c);
4. (3) corresponde simultaneamente aos casos (d), (e), (f) e (g):
   - $x \equiv y$:
     $(\lambda y.P)[y := N] \equiv \lambda z.(P^{y \to z}[y := N]) \equiv \lambda z.P^{y \to z}$
   - $x \not\equiv y$ e $x \notin FV(P)$:
     $(\lambda y.P)[x := N] \equiv \lambda z.(P^{y \to z}[x := N]) \equiv \lambda z.P^{y \to z}$
   - $x \not\equiv y$, $x \in FV(P)$ e $y \notin FV(N)$:
     $(\lambda y.P)[x := N] \equiv \lambda z.(P^{y \to z}[x := N])$
   - $x \not\equiv y$, $x \in FV(P)$ e $y \in FV(N)$:
     $(\lambda y.P)[x := N] \equiv \lambda z.(P^{y \to z}[x := N])$

Ocorre sempre uma conversão-α, mesmo que isto não seja necessário.

Substituição no Cálculo Lambda Tipado

Basta introduzir uma pequena modificação na regra (3), para especificar o tipo da variável:

(1a) $x[x := N] \equiv N$;

(1b) $y[x := N] \equiv y$, se $x \not\equiv y$;

(2) $(PQ)[x := N] \equiv (P[x := N])(Q[x := N])$;

(3) $(\lambda y : \sigma.P)[x := N] \equiv \lambda z : \sigma.(P^{y \to z}[x := N])$, se $\lambda z : \sigma.P^{y \to z}$ for um α-variante de $\lambda y : \sigma.P$ tal que $z \notin FV(N)$

Lema da Substituição

"Substitution Lemma"

Suponha que:

$$\Gamma', x : \sigma, \Gamma'' \vdash M : \tau$$

e também que:

$$\Gamma' \vdash N : \sigma$$

Então,

$$\Gamma', \Gamma'' \vdash M[x := N] : \tau$$

A substituição de uma variável por um termo do mesmo tipo preserva o tipo do termo original onde foi feita a substituição.

Redução-β

Um único passo ($\rightarrow_\beta$):

(1) (base) $(\lambda x : \sigma.M)N \rightarrow_\beta M[x := N]$

(2) (compatibilidade) se $M \rightarrow_\beta N$, então, para qualquer termo L:

- $ML \rightarrow_\beta NL$;
- $LM \rightarrow_\beta LN$;
- $\lambda x : \tau.M \rightarrow_\beta \lambda x : \tau.N$

Redução-β

Zero ou mais passos ($\twoheadrightarrow_\beta$):
$M \twoheadrightarrow_\beta N$ se e somente se existir $n \geq 0$ e termos M_0 e M_n tais que:

- $M_0 \equiv M$,
- $M_n \equiv N$,
- Para todo $0 \leq i < n, M_i \rightarrow_\beta M_{i+1}$.

Em outras palavras,

$$M \equiv M_0 \rightarrow_\beta M_1 \rightarrow_\beta M_2 \rightarrow_\beta \ldots \rightarrow_\beta M_{n-1} \rightarrow_\beta M_n \equiv N$$

Note que $\twoheadrightarrow_\beta$ estende $\rightarrow_\beta$: se $M \rightarrow_\beta N$, então $M \twoheadrightarrow_\beta N$; o contrário não é necessariamente verdadeiro.

Propriedades da Redução-β

- Reflexiva: para todo M, $M \twoheadrightarrow_\beta M$;
- Transitiva: para todo L, M e N, se $L \twoheadrightarrow_\beta M$ e $M \twoheadrightarrow_\beta N$, então $L \twoheadrightarrow_\beta N$.

Igualdade-β

$M =_\beta N$ se e somente se existir $n \geq 0$ e termos M_0 e M_n tais que:

- $M_0 \equiv M$,
- $M_n \equiv N$,
- Para todo $0 \leq i < n, M_i \rightarrow_\beta M_{i+1}$ ou $M_{i+1} \rightarrow_\beta M_i$.

Note que $=_\beta$ estende $\twoheadrightarrow_\beta$ em ambas as direções: se $M \twoheadrightarrow_\beta N$, ou $N \twoheadrightarrow_\beta M$, então $M =_\beta N$.

Propriedades da Igualdade-β

É uma relação de equivalência:

- Reflexiva: para todo M, $M =_\beta M$;
- Simétrica: para todo M e N, se $M =_\beta N$, então $N =_\beta M$;
- Transitiva: para todo L, M e N, se $L =_\beta M$ e $M =_\beta N$, então $L =_\beta N$.

Outros resultados:

- Se $M \twoheadrightarrow_\beta L_1$ e $M \twoheadrightarrow_\beta L_2$, então $L_1 =_\beta L_2$;
- Se $L_1 \twoheadrightarrow_\beta N$ e $L_2 \twoheadrightarrow_\beta N$, então $L_1 =_\beta L_2$.

Teorema de Church-Rosser

É válido no STLC da mesma forma que no Cálculo Lambda Não-Tipado. O motivo é que os tipos não são levados em consideração na definição da redução-β.
"Church-Rosser Theorem" (também conhecido como Teorema da Confluência)

- Seja M um termo lambda e suponha que $M \twoheadrightarrow_\beta N_1$ e $M \twoheadrightarrow_\beta N_2$;
- Então, existe um termo N_3 tal que $N_1 \twoheadrightarrow_\beta N_3$ e $N_2 \twoheadrightarrow_\beta N_3$.

O resultado da computação, se existir (no caso do Cálculo Lambda Não-Tipado), independe da ordem em que as reduções são feitas. No caso do STLC, como o resultado sempre existe, indica que a ordem da computação não é importante.

Teorema de Church-Rosser

Corolário:

- Sejam M e N dois termos lambda e suponha que $M =_\beta N$;
- Então, existe um termo L tal que $M \twoheadrightarrow_\beta L$ e $N \twoheadrightarrow_\beta L$.

Lema da Redução do Sujeito

"Subject Reduction Lemma"

Suponha que:

$$\Gamma \vdash L : \rho$$

e também que:

$$L \twoheadrightarrow_\beta L'$$

Então,

$$\Gamma \vdash L' : \rho$$

A redução-β não modifica o tipo do termo.

Teorema da Normalização Forte

"Strong Normalization Theorem"

Todo termo lambda tipado legal M é normalizável fortemente. Em outras palavras, toda computação é finita.

Conseqüências

As principais características do Cálculo Lambda Tipado (STLC) podem ser provadas a partir dos resultados apresentados anteriormente:

- Nenhum termo contém auto-aplicação;
- Todo termo possui forma normal-β (toda computação é finita);
- Nem todo termo possui um ponto fixo.

Exercício 2.1

Para cada um dos termos abaixo, determinar se o mesmo pode ser tipado com um tipo simples. Em caso afirmativo, qual o tipo do termo e das variáveis envolvidas? Em caso negativo, provar a sua resposta.

(a) xxy

(b) xyy

(c) xyx

(d) $x(xy)$

(e) $x(yx)$

Exercício 2.1

Solução

(a) xxy

$xxy \equiv (xx)y$

$$(appl)\ \dfrac{(appl)\ \dfrac{x : \_ \qquad x : \_}{xx : \_} \qquad y : \_}{(xx)y : \_}$$

Como o termo contém uma auto-aplicação (xx), ele não pode ser tipado.

Exercício 2.1

Solução

(b) xyy

$$\cfrac{(appl)\ \cfrac{x : \_ \qquad y : \_}{xy : \_} \qquad y : \_}{xyy : \_}\ (appl)$$

Exercício 2.1

Solução

(b) xyy

$$
(appl)\,\dfrac{(appl)\,\dfrac{x:\alpha\rightarrow(\alpha\rightarrow\beta)\qquad y:\alpha}{xy:\alpha\rightarrow\beta}\qquad y:\alpha}{xyy:\beta}
$$

Exercício 2.1

Solução

(c) xyx

$$
(appl)\,\dfrac{(appl)\,\dfrac{x : \_ \qquad y : \_}{xy : \_} \qquad x : \_}{xyx : \_}
$$

Exercício 2.1

Solução

(c) xyx

$$
\dfrac{(appl)\ \dfrac{\boxed{x : \rho \to (\alpha \to \beta)} \qquad y : \rho}{xy : \alpha \to \beta} \qquad \boxed{x : \alpha}}{xyx : \beta}\ (appl)
$$

Como a variável x não pode ter dois tipos diferentes, segue que o termo não é tipável.

Exercício 2.1

Solução

(d) $x(xy)$

$$(appl)\,\dfrac{x : \_ \qquad (appl)\,\dfrac{x : \_ \qquad y : \_}{xy : \_}}{x(xy) : \_}$$

Exercício 2.1

Solução

(d) $x(xy)$

$$\dfrac{\boxed{x:\alpha\to\beta} \qquad (appl)\,\dfrac{\boxed{x:\rho\to\alpha} \qquad y:\rho}{xy:\alpha}}{x(xy):\beta}\,(appl)$$

Basta agora fazer $\beta \equiv \rho \equiv \alpha$:

$$\dfrac{x:\alpha\to\alpha \qquad (appl)\,\dfrac{x:\alpha\to\alpha \qquad y:\alpha}{xy:\alpha}}{x(xy):\alpha}\,(appl)$$

Exercício 2.1

Solução

(e) $x(yx)$

$$\dfrac{x : \_ \qquad (appl)\,\dfrac{y : \_ \qquad x : \_}{yx : \_}}{x(yx) : \_}\,(appl)$$

Exercício 2.1

Solução

(e) $x(yx)$

$$\dfrac{\boxed{x : \alpha \to \beta} \qquad (appl)\,\dfrac{y : \rho \to \alpha \qquad \boxed{x : \rho}}{yx : \alpha}}{x(yx) : \beta}\,(appl)$$

Basta agora fazer $\rho \equiv \alpha \to \beta$:

$$\dfrac{x : \alpha \to \beta \qquad (appl)\,\dfrac{y : (\alpha \to \beta) \to \alpha \qquad x : \alpha \to \beta}{yx : \alpha}}{x(yx) : \beta}\,(appl)$$

Exercício 2.2

Para cada um dos termos abaixo, determinar o tipo do mesmo.

(a) $\overline{0} \equiv \lambda xy.y$

(b) $\overline{1} \equiv \lambda xy.xy$

(c) $\overline{2} \equiv \lambda xy.x(xy)$

Exercício 2.2

Solução

(a) $\overline{0} \equiv \lambda xy.y$

$$
(abst)\,\dfrac{(abst)\,\dfrac{y : \alpha \vdash y : \alpha}{x : \beta \vdash (\lambda y : \alpha.y) : \alpha \to \alpha}}{\emptyset \vdash (\lambda x : \beta.\lambda y : \alpha.y) : \beta \to \alpha \to \alpha}
$$

Exercício 2.2

Solução

(b) $\overline{1} \equiv \lambda xy.xy$

$$
\dfrac{\dfrac{\dfrac{x : \alpha \to \beta \qquad y : \alpha}{y : \alpha \vdash xy : \beta}\,(appl)}{x : \alpha \to \beta \vdash (\lambda y : \alpha.xy) : \alpha \to \beta}\,(abst)}{\emptyset \vdash (\lambda x : \alpha \to \beta.\lambda y : \alpha.xy) : (\alpha \to \beta) \to \alpha \to \beta}\,(abst)
$$

Exercício 2.2

Solução

(c) $\overline{2} \equiv \lambda xy.x(xy)$

$$
(abst)\,\dfrac{(abst)\,\dfrac{(appl)\,\dfrac{x : \alpha \to \alpha \qquad (appl)\,\dfrac{x : \alpha \to \alpha \qquad y : \alpha}{xy : \alpha}}{y : \alpha \vdash x(xy) : \alpha}}{x : \alpha \to \alpha \vdash (\lambda y : \alpha . x(xy)) : \alpha \to \alpha}}{\emptyset \vdash (\lambda x : \alpha \to \alpha . \lambda y : \alpha . x(xy)) : (\alpha \to \alpha) \to \alpha \to \alpha}
$$

Exercício 2.2

Comentário

Observar que o tipo $(\alpha \to \alpha) \to \alpha \to \alpha$ também pode ser empregado para os termos $\overline{0}$ e $\overline{1}$:

(a) $\overline{0} \equiv \lambda xy.y$: basta fazer $\beta \equiv \alpha \to \alpha$;

(b) $\overline{1} \equiv \lambda xy.xy$: basta fazer $\beta \equiv \alpha$.

- Os termos $\overline{0}, \overline{1}$ e $\overline{2}$ podem ser tipados da mesma forma, com o tipo $(\alpha \to \alpha) \to \alpha \to \alpha$;
- Todos os demais Numerais de Church também podem ser tipados da mesma forma, o que é de se esperar pelo que os mesmos representam (números naturais).

Exercício 2.3

Para cada um dos termos abaixo, determinar o tipo do mesmo.

(a) $\mathsf{K} \equiv \lambda xy.x$

(b) $\mathsf{S} \equiv \lambda xyz.xz(yz)$

Exercício 2.3

Solução

(a) $\mathsf{K} \equiv \lambda xy.x$

$$\dfrac{(abst)\ \dfrac{y : \beta \vdash x : \alpha}{x : \alpha \vdash (\lambda y : \beta.x) : \beta \to \alpha}}{\emptyset \vdash (\lambda x : \alpha.\lambda y : \beta.x) : \alpha \to \beta \to \alpha}\ (abst)$$

Exercício 2.3

Solução

(b) $\mathbf{S} \equiv \lambda xyz.xz(yz)$

$$
(abst)\,\dfrac{(abst)\,\dfrac{(abst)\,\dfrac{(appl)\,\dfrac{(appl)\,\dfrac{x : \alpha \to (\beta \to \sigma) \qquad z : \alpha}{xz : \beta \to \sigma} \qquad (appl)\,\dfrac{y : \alpha \to \beta \qquad z : \alpha}{yz : \beta}}{z : \alpha \vdash xz(yz) : \sigma}}{y : \alpha \to \beta \vdash M) : \alpha \to \sigma}}{x : \alpha \to (\beta \to \sigma) \vdash (\lambda y : \alpha \to \beta.M) : (\alpha \to \beta) \to \alpha \to \sigma}}{\emptyset \vdash (\lambda x : \alpha \to (\beta \to \sigma).\lambda y : \alpha \to \beta.M) : (\alpha \to \beta \to \sigma) \to (\alpha \to \beta) \to \alpha \to \sigma}
$$

com $M \equiv \lambda z : \alpha.xz(yz)$.

Exercício 2.4

Faça a tipagem de cada um dos seguintes termos e das respectivas variáveis:

(a) $\lambda xyz.x(yz)$

(b) $\lambda xyz.y(xz)x$

Exercício 2.4

Solução

(a) $\lambda xyz.x(yz)$

$$
(abst)\,\dfrac{(abst)\,\dfrac{(abst)\,\dfrac{(appl)\,\dfrac{x : \_ \qquad (appl)\,\dfrac{y : \_ \qquad z : \_}{yz : \_}}{z : \_ \ :\vdash x(yz) : \_}}{y : \_ \vdash (\lambda z : \_.x(yz)) : \_}}{x : \_ \vdash (\lambda y : \_.\lambda z : \_.x(yz)) : \_}}{\emptyset \vdash (\lambda x : \_.\lambda y : \_.\lambda z : \_.x(yz)) : \_}
$$

Exercício 2.4

Solução

(a) $\lambda xyz.x(yz)$

$$
\dfrac{\dfrac{\dfrac{x:\beta\to\gamma \qquad (appl)\ \dfrac{y:\alpha\to\beta \qquad z:\alpha}{yz:\beta}}{z:\alpha\vdash x(yz):\gamma}\,(appl)}{y:\alpha\to\beta\vdash(\lambda z:\alpha.x(yz)):\alpha\to\gamma}\,(abst)}{\dfrac{x:\beta\to\gamma\vdash(\lambda y:\alpha\to\beta.\lambda z:\alpha.x(yz)):(\alpha\to\beta)\to\alpha\to\gamma}{\emptyset\vdash(\lambda x:\beta\to\gamma.\lambda y:\alpha\to\beta.\lambda z:\alpha.x(yz)):(\beta\to\gamma)\to(\alpha\to\beta)\to\alpha\to\gamma}\,(abst)}\,(abst)
$$

Exercício 2.4

Solução

(b) $\lambda xyz.y(xz)x$

$$
(abst)\dfrac{(abst)\dfrac{(abst)\dfrac{(appl)\dfrac{(appl)\dfrac{y : \_ \qquad (appl)\dfrac{x : \_ \qquad z : \_}{xz : \_}}{y(xz) : \_} \qquad x : \_}{z : \_ \vdash y(xz)x : \_}}{y : \_ \vdash (\lambda z : \_.y(xz)x) : \_}}{x : \_ \vdash (\lambda y : \_.\lambda z : \_.y(xz)x) : \_}}{\emptyset \vdash (\lambda x : \_.\lambda y : \_.\lambda z : \_.y(xz)x) : \_}
$$

Exercício 2.4

Solução

(b) $\lambda xyz.y(xz)x$

$$
(abst)\,\dfrac{(abst)\,\dfrac{(abst)\,\dfrac{(appl)\,\dfrac{(appl)\,\dfrac{y : \beta \to (\alpha \to \beta) \to \gamma \qquad (appl)\,\dfrac{x : \alpha \to \beta \qquad z : \alpha}{xz : \beta}}{y(xz) : (\alpha \to \beta) \to \gamma} \qquad x : \alpha \to \beta}{z : \alpha \vdash y(xz)x : \gamma}}{y : \beta \to (\alpha \to \beta) \to \gamma \vdash (\lambda z : \alpha.y(xz)x) : \alpha \to \gamma}}{x : \alpha \to \beta \vdash (\lambda y : \beta \to (\alpha \to \beta) \to \gamma.\lambda z : \alpha.y(xz)x) : (\beta \to (\alpha \to \beta) \to \gamma) \to \alpha \to \gamma}}{\emptyset \vdash (\lambda x : \alpha \to \beta.\lambda y : \beta \to (\alpha \to \beta) \to \gamma.\lambda z : \alpha.y(xz)x) : \tau}
$$

com $\tau \equiv (\alpha \to \beta) \to (\beta \to (\alpha \to \beta) \to \gamma) \to \alpha \to \gamma$.

Exercício 2.6

Obtenha um contexto e um tipo para o seguinte termo:

$$\lambda x : ((\alpha \to \beta) \to \alpha).x(\lambda z : \alpha.y)$$

Exercício 2.6

Solução

$$
(abst)\ \dfrac{(appl)\ \dfrac{x : (\alpha \to \beta) \to \alpha \qquad (abst)\ \dfrac{z : \alpha \vdash y : \_}{\lambda z : \alpha . y : \alpha \to \_}}{x : (\alpha \to \beta) \to \alpha \vdash x(\lambda z : \alpha . y) : \_}}{\lambda x : (\alpha \to \beta) \to \alpha . x(\lambda z : \alpha . y) : \_}
$$

Exercício 2.6

Solução

$$
\dfrac{\dfrac{x:(\alpha\to\beta)\to\alpha \qquad (abst)\,\dfrac{z:\alpha\vdash y:\beta}{\lambda z:\alpha.y:\alpha\to\beta}}{x:(\alpha\to\beta)\to\alpha\vdash x(\lambda z:\alpha.y):\alpha}\,(appl)}{\lambda x:(\alpha\to\beta)\to\alpha.x(\lambda z:\alpha.y):(\alpha\to\beta\to\alpha)\to\alpha}\,(abst)
$$

Como a variável y é livre, então o tipo da mesma precisa ser definido pelo contexto:

$$
y:\beta\vdash\lambda x:(\alpha\to\beta)\to\alpha.x(\lambda z:\alpha.y):(\alpha\to\beta\to\alpha)\to\alpha
$$

Exercício 2.8

Obtenha tipos para as variáveis x, y e z de tal forma que:

$$\lambda xy.y(\lambda z.yx) : (\gamma \to \beta) \to ((\gamma \to \beta) \to \beta) \to \beta$$

Exercício 2.8

Solução

Não é difícil inferir que:

- $x : \gamma \rightarrow \beta$ e
- $y : (\gamma \rightarrow \beta) \rightarrow \beta$

Resta, portanto, determinar o tipo de z.

Exercício 2.8

Solução

$$
(abst)\;\dfrac{(abst)\;\dfrac{(appl)\;\dfrac{y:(\gamma\to\beta)\to\beta \qquad (abst)\;\dfrac{(appl)\;\dfrac{y:(\gamma\to\beta)\to\beta \qquad x:\gamma\to\beta}{z:\_ \vdash yx:\beta}}{\lambda z:\_.yx:\_\to\beta}}{y:(\gamma\to\beta)\to\beta \vdash y(\lambda z:\_.yx):\beta}}{x:\gamma\to\beta \vdash \lambda y:(\gamma\to\beta)\to\beta.y(\lambda z:\_.yx):((\gamma\to\beta)\to\beta)\to\beta}}{\emptyset \vdash \lambda x:\gamma\to\beta.\lambda y:(\gamma\to\beta)\to\beta.y(\lambda z:\_.yx):(\gamma\to\beta)\to((\gamma\to\beta)\to\beta)\to\beta}
$$

Exercício 2.8

Solução

$$
(abst)\ \dfrac{(abst)\ \dfrac{(appl)\ \dfrac{y : (\gamma \to \beta) \to \beta \qquad (abst)\ \dfrac{(appl)\ \dfrac{y : (\gamma \to \beta) \to \beta \qquad x : \gamma \to \beta}{z : \gamma \vdash yx : \beta}}{\lambda z : \gamma . yx : \gamma \to \beta}}{y : (\gamma \to \beta) \to \beta \vdash y(\lambda z : \gamma . yx) : \beta}}{x : \gamma \to \beta \vdash \lambda y : (\gamma \to \beta) \to \beta . y(\lambda z : \gamma . yx) : ((\gamma \to \beta) \to \beta) \to \beta}}{\emptyset \vdash \lambda x : \gamma \to \beta . \lambda y : (\gamma \to \beta) \to \beta . y(\lambda z : \gamma . yx) : (\gamma \to \beta) \to ((\gamma \to \beta) \to \beta) \to \beta}
$$

Exercício 2.9

Prove que os seguintes julgamentos são legais:

(a) $x : \delta \to \delta \to \alpha, y : \gamma \to \alpha, z : \alpha \to \beta \vdash \lambda u : \delta.\lambda v : \gamma.z(yv) : \delta \to \gamma \to \beta$

(b) $x : \delta \to \delta \to \alpha, y : \gamma \to \alpha, z : \alpha \to \beta \vdash \lambda u : \delta.\lambda v : \gamma.z(xuu) : \delta \to \gamma \to \beta$

Exercício 2.9

Solução

(a) $x : \delta \to \delta \to \alpha, y : \gamma \to \alpha, z : \alpha \to \beta \vdash \lambda u : \delta.\lambda v : \gamma.z(yv) : \delta \to \gamma \to \beta$

$$
(abst)\ \dfrac{(abst)\ \dfrac{(appl)\ \dfrac{z : \_ \qquad (appl)\ \dfrac{y : \_ \qquad v : \_}{yv : \_}}{v : \_ \vdash z(yv) : \_}}{u : \_ \vdash \lambda v : \_.z(yv) : \_}}{\vdash \lambda u : \_.\lambda v : \_.z(yv) : \_}
$$

Exercício 2.9

Solução

(a) $x : \delta \to \delta \to \alpha, y : \gamma \to \alpha, z : \alpha \to \beta \vdash \lambda u : \delta.\lambda v : \gamma.z(yv) : \delta \to \gamma \to \beta$

$$
(abst)\,\dfrac{(abst)\,\dfrac{(appl)\,\dfrac{z : \alpha \to \beta \qquad (appl)\,\dfrac{y : \gamma \to \alpha \qquad v : \gamma}{yv : \alpha}}{v : \gamma \vdash z(yv) : \beta}}{u : \delta \vdash \lambda v : \gamma.z(yv) : \gamma \to \beta}}{y : \gamma \to \alpha, z : \alpha \to \beta \vdash \lambda u : \delta.\lambda v : \gamma.z(yv) : \delta \to \gamma \to \beta}
$$

Exercício 2.9

Solução

(b) $x : \delta \to \delta \to \alpha, y : \gamma \to \alpha, z : \alpha \to \beta \vdash \lambda u : \delta.\lambda v : \gamma.z(xuu) : \delta \to \gamma \to \beta$

$$(abst)\,\dfrac{(abst)\,\dfrac{(appl)\,\dfrac{z : \_ \qquad (appl)\,\dfrac{(appl)\,\dfrac{x : \_ \qquad u : \_}{xu : \_} \qquad u : \_}{xuu : \_}}{v : \_ \vdash z(xuu) : \_}}{u : \_ \vdash \lambda v : \_.z(xuu) : \_}}{\vdash \lambda u : \_.\lambda v : \_.z(xuu) : \_}$$

Exercício 2.9

Solução

(b) $x : \delta \to \delta \to \alpha, y : \gamma \to \alpha, z : \alpha \to \beta \vdash \lambda u : \delta.\lambda v : \gamma.z(xuu) : \delta \to \gamma \to \beta$

$$
(abst)\;\dfrac{(abst)\;\dfrac{(appl)\;\dfrac{z : \alpha \to \beta \qquad (appl)\;\dfrac{(appl)\;\dfrac{x : \delta \to \delta \to \alpha \qquad u : \delta}{xu : \delta \to \alpha} \qquad u : \delta}{xuu : \alpha}}{v : \gamma \vdash z(xuu) : \beta}}{u : \delta \vdash \lambda v : \gamma.z(xuu) : \gamma \to \beta}}{x : \delta \to \delta \to \alpha, z : \alpha \to \beta \vdash \lambda u : \delta.\lambda v : \gamma.z(xuu) : \delta \to \gamma \to \beta}
$$

Exercício 2.10

Prove que os seguintes termos são legais:

(a) $xz(yz)$

(b) $\lambda x : (\alpha \to \beta) \to \beta . x(yz)$

(c) $\lambda y : \alpha . \lambda z : \beta \to \gamma . z(xyy)$

(d) $\lambda x : \alpha \to \beta . y(xz)z$

Exercício 2.10

Solução

(a) $xz(yz)$

$$(appl)\ \dfrac{(appl)\ \dfrac{x : \_ \qquad z : \_}{(xz) : \_} \qquad (appl)\ \dfrac{y : \_ \qquad z : \_}{(yz) : \_}}{(xz)(yz) : \_}$$

Exercício 2.10

Solução

(a) $xz(yz)$

$$(appl)\,\dfrac{(appl)\,\dfrac{x : \alpha \to (\beta \to \gamma) \qquad z : \alpha}{(xz) : \beta \to \gamma} \qquad (appl)\,\dfrac{y : \alpha \to \beta \qquad z : \alpha}{(yz) : \beta}}{(xz)(yz) : \gamma}$$

Exercício 2.10

Solução

(b) $\lambda x : (\alpha \to \beta) \to \beta . x(yz)$

$$
\dfrac{(appl)\ \dfrac{x : (\alpha \to \beta) \to \beta \qquad (appl)\ \dfrac{y : \_ \qquad z : \_}{(yz) : \_}}{x : (\alpha \to \beta) \to \beta \vdash x(yz) : \_}}{y : \_ , z : \_ \vdash \lambda x : (\alpha \to \beta) \to \beta . x(yz) : \_}\ (abst)
$$

Exercício 2.10

Solução

(b) $\lambda x : (\alpha \to \beta) \to \beta . x(yz)$

$$(abst)\ \dfrac{(appl)\ \dfrac{x : (\alpha \to \beta) \to \beta \qquad (appl)\ \dfrac{y : \gamma \to \alpha \to \beta \qquad z : \gamma}{(yz) : \alpha \to \beta}}{x : (\alpha \to \beta) \to \beta \vdash x(yz) : \beta}}{y : \gamma \to \alpha \to \beta, z : \gamma \vdash \lambda x : (\alpha \to \beta) \to \beta . x(yz) : ((\alpha \to \beta) \to \beta) \to \beta}$$

Exercício 2.10

Solução

(c) $\lambda y:\alpha.\lambda z:\beta\rightarrow\gamma.z(xyy)$

$$
(abst)\;\dfrac{(abst)\;\dfrac{(appl)\;\dfrac{z:\beta\rightarrow\gamma \qquad (appl)\;\dfrac{(appl)\;\dfrac{x:\_ \qquad y:\alpha}{xy:\_} \qquad y:\alpha}{xyy:\_}}{z:\beta\rightarrow\gamma\vdash z(xyy):\_}}{y:\alpha\vdash\lambda z:\beta\rightarrow\gamma.z(xyy):\_}}{x:\_\vdash\lambda y:\alpha.\lambda z:\beta\rightarrow\gamma.z(xyy):\_}
$$

Exercício 2.10

Solução

(c) $\lambda y : \alpha . \lambda z : \beta \to \gamma . z(xyy)$

$$
(abst)\ \dfrac{(abst)\ \dfrac{(appl)\ \dfrac{z : \beta \to \gamma \qquad (appl)\ \dfrac{(appl)\ \dfrac{x : \alpha \to \alpha \to \beta \qquad y : \alpha}{xy : \alpha \to \beta} \qquad y : \alpha}{xyy : \beta}}{z : \beta \to \gamma \vdash z(xyy) : \gamma}}{y : \alpha \vdash \lambda z : \beta \to \gamma . z(xyy) : (\beta \to \gamma) \to \gamma}}{x : \alpha \to \alpha \to \beta \vdash \lambda y : \alpha . \lambda z : \beta \to \gamma . z(xyy) : \alpha \to (\beta \to \gamma) \to \gamma}
$$

Exercício 2.10

Solução

(d) $\lambda x : \alpha \to \beta . y(xz)z$

$$
(abst)\ \dfrac{(appl)\ \dfrac{(appl)\ \dfrac{y : \_ \qquad (appl)\ \dfrac{x : \alpha \to \beta \qquad z : \_}{xz : \_}}{y(xz) : \_} \qquad z : \_}{x : \alpha \to \beta \vdash y(xz)z : \_}}{y : \_, z : \_ \vdash \lambda x : \alpha \to \beta . y(xz)z : \_}
$$

Exercício 2.10

Solução

(d) $\lambda x : \alpha \to \beta . y(xz)z$

$$
(abst)\,\dfrac{(appl)\,\dfrac{(appl)\,\dfrac{y : \beta \to \alpha \to \gamma \qquad (appl)\,\dfrac{x : \alpha \to \beta \qquad z : \alpha}{xz : \beta}}{y(xz) : \alpha \to \gamma} \qquad z : \alpha}{x : \alpha \to \beta \vdash y(xz)z : \gamma}}{y : \beta \to \alpha \to \gamma, z : \alpha \vdash \lambda x : \alpha \to \beta . y(xz)z : (\alpha \to \beta) \to \gamma}
$$

Exercício 2.11

Obtenha um termo para cada um dos seguintes tipos, considerando o contexto vazio:

(a) $(\alpha \to \alpha \to \gamma) \to \alpha \to \beta \to \gamma$

(b) $((\alpha \to \gamma) \to \alpha) \to (\alpha \to \gamma) \to \beta \to \gamma$

Exercício 2.11

Solução

(a) $(\alpha \to \alpha \to \gamma) \to \alpha \to \beta \to \gamma$

$$
\dfrac{\dfrac{\dfrac{z : \beta \vdash \_ : \gamma}{y : \alpha \vdash \lambda z : \beta.\_ : \beta \to \gamma}\ (abst)}{x : \alpha \to \alpha \to \gamma \vdash \lambda y : \alpha.\lambda z : \beta.\_ : \alpha \to \beta \to \gamma}\ (abst)}{\lambda x : \alpha \to \alpha \to \gamma.\lambda y : \alpha.\lambda z : \beta.\_ : (\alpha \to \alpha \to \gamma) \to \alpha \to \beta \to \gamma}\ (abst)
$$

Exercício 2.11

Solução

(a) $(\alpha \to \alpha \to \gamma) \to \alpha \to \beta \to \gamma$

$$(abst)\,\dfrac{(abst)\,\dfrac{(abst)\,\dfrac{(appl)\,\dfrac{(appl)\,\dfrac{x : \alpha \to \alpha \to \gamma \qquad y : \alpha}{xy : \alpha \to \gamma} \qquad y : \alpha}{z : \beta \vdash xyy : \gamma}}{y : \alpha \vdash \lambda z : \beta . xyy : \beta \to \gamma}}{x : \alpha \to \alpha \to \gamma \vdash \lambda y : \alpha . \lambda z : \beta . xyy : \alpha \to \beta \to \gamma}}{\lambda x : \alpha \to \alpha \to \gamma . \lambda y : \alpha . \lambda z : \beta . xyy : (\alpha \to \alpha \to \gamma) \to \alpha \to \beta \to \gamma}$$

Exercício 2.11

Solução

(b) $((\alpha \to \gamma) \to \alpha) \to (\alpha \to \gamma) \to \beta \to \gamma$

$$(abst)\dfrac{(abst)\dfrac{(abst)\dfrac{z : \beta \vdash \_ : \gamma}{y : \alpha \to \gamma \vdash \lambda z : \beta.\_ : (\alpha \to \gamma) \to \beta \to \gamma}}{x : (\alpha \to \gamma) \to \alpha \vdash \lambda y : \alpha \to \gamma.\lambda z : \beta.\_ : (\alpha \to \gamma) \to \beta \to \gamma}}{\lambda x : (\alpha \to \gamma) \to \alpha.\lambda y : \alpha \to \gamma.\lambda z : \beta.\_ : ((\alpha \to \gamma) \to \alpha) \to (\alpha \to \gamma) \to \beta \to \gamma}$$

Exercício 2.11

Solução

(b) $((\alpha \to \gamma) \to \alpha) \to (\alpha \to \gamma) \to \beta \to \gamma$

$$
(abst)\ \dfrac{(abst)\ \dfrac{(abst)\ \dfrac{(appl)\ \dfrac{y : \alpha \to \gamma \qquad (appl)\ \dfrac{x : (\alpha \to \gamma) \to \alpha \qquad y : \alpha \to \gamma}{xy : \alpha}}{z : \beta \vdash y(xy) : \gamma}}{y : \alpha \to \gamma \vdash \lambda z : \beta . y(xy) : \beta \to \gamma}}{x : (\alpha \to \gamma) \to \alpha \vdash \lambda y : \alpha \to \gamma . \lambda z : \beta . y(xy) : (\alpha \to \gamma) \to \beta \to \gamma}}{\lambda x : (\alpha \to \gamma) \to \alpha . \lambda y : \alpha \to \gamma . \lambda z : \beta . y(xy) : ((\alpha \to \gamma) \to \alpha) \to (\alpha \to \gamma) \to \beta \to \gamma}
$$

Observação

Nos exercícios 2.12, 2.13 (c) e 2.14 (a seguir) é usada a seguinte estratégia:

- Se existe um termo M do tipo β,
- Então sempre existe um termo N do tipo $\alpha \rightarrow \beta$, para qualquer tipo α.

Basta fazer $N \equiv \lambda m : \alpha.M$. Note que $N : \alpha \rightarrow \beta$.
Em outras palavras,

- Se o tipo β é habitado,
- Então o tipo $\alpha \rightarrow \beta$ também é habitado, para qualquer α.

Novamente, perceba-se a similaridade com o operador lógico implicação ($\Rightarrow$): se a conclusão é verdadeira, então a implicação também é verdadeira, não importando a hipótese.

Exercício 2.12

Construa um termo de cada um dos seguintes tipos:

(a) $((\alpha \to \beta) \to \alpha) \to (\alpha \to \alpha \to \beta) \to \alpha$

(b) $((\alpha \to \beta) \to \alpha) \to (\alpha \to \alpha \to \beta) \to \beta$

Exercício 2.12

Solução

(a) $\tau \equiv ((\alpha \to \beta) \to \alpha) \to (\alpha \to \alpha \to \beta) \to \alpha$

Sejam:

- $x : (\alpha \to \beta) \to \alpha$ e
- $y : \alpha \to \alpha \to \beta$.

Então:

$$\lambda x : (\alpha \to \beta) \to \alpha.\lambda y : \alpha \to \alpha \to \beta.x(\lambda m : \alpha.ymm) : \tau$$

Note que:

- $ymm : \beta$,
- $\lambda m : \alpha.ymm : \alpha \to \beta$ e
- $x(\lambda m : \alpha.ymm) : \alpha$.

Exercício 2.12

Solução

(b) $((\alpha \rightarrow \beta) \rightarrow \alpha) \rightarrow (\alpha \rightarrow \alpha \rightarrow \beta) \rightarrow \beta$

Sejam:

- $x : (\alpha \rightarrow \beta) \rightarrow \alpha$ e
- $y : \alpha \rightarrow \alpha \rightarrow \beta$.

Então:

$$\lambda x : (\alpha \rightarrow \beta) \rightarrow \alpha.\lambda y : \alpha \rightarrow \alpha \rightarrow \beta.y(x(\lambda m : \alpha.ymm))(x(\lambda m : \alpha.ymm)) : \tau$$

Note que:

- $ymm : \beta$,
- $\lambda m : \alpha.ymm : \alpha \rightarrow \beta$,
- $x(\lambda m : \alpha.ymm) : \alpha$,
- $y(x(\lambda m : \alpha.ymm)) : \alpha \rightarrow \beta$ e
- $y(x(\lambda m : \alpha.ymm))(x(\lambda m : \alpha.ymm)) : \beta$.

Exercício 2.13

Obtenha um termo do tipo τ no contexto Γ para de cada um dos seguintes casos:

(a) $\tau \equiv (\alpha \to \beta) \to \alpha \to \gamma, \Gamma \equiv x : \alpha \to \beta \to \gamma$

(b) $\tau \equiv \alpha \to (\alpha \to \beta) \to \gamma, \Gamma \equiv x : \alpha \to \beta \to \alpha \to \gamma$

(c) $\tau \equiv (\alpha \to \gamma) \to (\beta \to \alpha) \to \gamma, \Gamma \equiv x : (\beta \to \gamma) \to \gamma$

Exercício 2.13

Solução

(a) $\tau \equiv (\alpha \to \beta) \to \alpha \to \gamma, \Gamma \equiv x : \alpha \to \beta \to \gamma$

Sejam:

- $y : \alpha \to \beta$ e
- $z : \alpha$.

Então:

$$\Gamma \vdash \lambda y : \alpha \to \beta.\lambda z : \alpha.xy : \tau$$

Exercício 2.13

Solução

(b) $\tau \equiv \alpha \to (\alpha \to \beta) \to \gamma, \Gamma \equiv x : \alpha \to \beta \to \alpha \to \gamma$

Sejam:

- $y : \alpha$ e
- $z : \alpha \to \beta$.

Então:

$$\Gamma \vdash \lambda y : \alpha.\lambda z : \alpha \to \beta.(xy)(zy)y : \tau$$

Exercício 2.13

Solução

(c) $\tau \equiv (\alpha \to \gamma) \to (\beta \to \alpha) \to \gamma, \Gamma \equiv x : (\beta \to \gamma) \to \gamma$

Sejam:

- $y : \alpha$,
- $z : \gamma$ e
- $w : \beta \to \alpha$.

Então:

$$\Gamma \vdash \lambda y : \alpha.\lambda z : \gamma.\lambda w : \beta \to \alpha.x(\lambda m : \beta.z) : \tau$$

Exercício 2.14

Obtenha um termo do tipo:

$$\tau \equiv \alpha \rightarrow \beta \rightarrow \gamma$$

no contexto:

$$\Gamma \equiv x : (\gamma \rightarrow \beta) \rightarrow \alpha \rightarrow \gamma$$

Exercício 2.14
Solução

- $\tau \equiv \alpha \rightarrow \beta \rightarrow \gamma, \Gamma \equiv x : (\gamma \rightarrow \beta) \rightarrow \alpha \rightarrow \gamma$

Sejam:

- $y : \alpha$,
- $z : \beta$.

Então:

$$\Gamma \vdash \lambda y : \alpha . \lambda z : \beta . x(\lambda m : \gamma . z) y : \tau$$