

Para entender melhor o que motiva a criação da teoria dos tipos, essas notas serão iniciadas com o cálculo  $\lambda$  não tipado.

## 0.1 Definições

O cálculo lambda serve como uma abstração em cima do conceito de função. Uma função é algo que pega um *input* e retorna um *output*, por exemplo a função  $f(x) = x^2$  pega um input  $x$  e retorna seu quadrado  $x^2$ . No cálculo lambda, essa função é denotada por  $\lambda x.x^2$ , onde  $\lambda x$  simboliza que essa função espera receber como entrada  $x$ . Quando queremos receber uma resposta específica de uma função, usamos números no lugar das variáveis, como por exemplo  $f(3) = 3^2 = 9$ . No cálculo lambda, isso é feito na forma de  $(\lambda x.x^2)(3)$ .

Esses dois princípios de construção são definidos como:

- **Abstração:** Seja  $M$  uma expressão e  $x$  uma variável, podemos construir uma nova expressão  $\lambda x.M$ . Essa expressão é chamada de Abstração de  $x$  sobre  $M$
- **Aplicação:** Sejam  $M$  e  $N$  duas es expressões, podemos construir uma expressão  $MN$ . Essa expressão é chamada de Aplicação de  $M$  em  $N$ .

Dadas essas operações, precisamos também de uma definição que dê conta do processo de encontrar o resultado após a aplicação em uma função. Esse processo é chamado de  $\beta$ -redução. Ela faz uso da substituição e usa como notação os colchetes.

**Definição 0.1** ( $\beta$ -redução). A  $\beta$ -redução é o processo de reescrita de uma expressão da forma  $(\lambda x.M)N$  em outra expressão  $M[x := N]$ , ou seja, a expressão  $M$  na qual todo  $x$  foi substituído por  $N$ .

## 0.2 Sintáxe do Cálculo Lambda

É interessante definir a sintaxe do cálculo lambda de forma mais formal. Para isso, são utilizados métodos que podem ser familiares para aqueles que já trabalharam com lógica proposicional, lógica de primeira ordem ou teoria de modelos.

**Definição 0.2.** (i) Os *termos lambda* são palavras em cima do seguinte alfabeto:

- variáveis:  $v_0, v_1, \dots$
- abstrator:  $\lambda$
- parentesis:  $(, )$

(ii) O conjunto de  $\lambda$ -termos  $\Lambda$  é definido de forma indutiva da seguinte forma:

- Se  $x$  é uma variável, então  $x \in \Lambda$
- $M \in \Lambda \rightarrow (\lambda x.M) \in \Lambda$
- $M, N \in \Lambda \rightarrow MN \in \Lambda$

Na teoria dos tipos e no cálculo lambda, é utilizada uma forma concisa de definir esses termos chamada de Formalismo de Backus-Naur ou Forma Normal de Backus (BNF, em inglês). Nessa forma, a definição anterior é reduzida à:

$$\Lambda = V | (\Lambda \Lambda) | (\lambda V \Lambda)$$

Onde  $V$  é o conjunto de variáveis  $V = \{x, y, z, \dots\}$

Para expressar igualdade entre dois termos de  $\Lambda$  utilizamos o símbolo  $\equiv$ .

Algumas definições indutivas podem ser formadas a partir da definição dos  $\lambda$ -termos.

**Definição 0.3** (Multiconjunto de subtermos).

1. (Base)  $Sub(x) = \{x\}$ , para todo  $x \in V$
2. (Aplicação)  $Sub(MN) = Sub(M) \cup Sub(N) \cup \{(MN)\}$
3. (Abstração)  $Sub(\lambda x.M) = Sub(M) \cup \{(\lambda x.M)\}$

Observações:

- (i) Um subtermo pode ocorrer múltiplas vezes, por isso é escolhido chamar de multiconjunto

**Lemma 0.1** (Propriedades de  $Sub$ ).

- (Reflexividade) Para todo  $\lambda$ -termo  $M$ , temos que  $M \in Sub(M)$
- (Transitividade) Se  $L \in Sub(M)$  e  $M \in Sub(N)$ , então  $L \in Sub(N)$ .

**Definição 0.4** (Subtermo próprio).  $L$  é um subtermo próprio de  $M$  se  $L$  é subtermo de  $M$  e  $L \neq M$

Exemplos:

1. Seja o termo  $\lambda x.\lambda y.xy$ , vamos calcular seus subtermos:

$$\begin{aligned} Sub(\lambda x.\lambda y.xy) &= \{\lambda x.\lambda y.xy\} \cup Sub(\lambda y.xy) \\ &= \{\lambda x.\lambda y.xy\} \cup \{\lambda y.xy\} \cup Sub(xy) \\ &= \{\lambda x.\lambda y.xy\} \cup \{\lambda y.xy\} \cup Sub(x) \cup Sub(y) \\ &= \{\lambda x.\lambda y.xy, \lambda y.xy, x, y\} \end{aligned}$$

2. Seja o termo  $(y(\lambda x.(xyz)))$ , vamos calcular os seus subtermos:

$$\begin{aligned} Sub(y(\lambda x.(xyz))) &= Sub(y) \cup Sub((\lambda x.(xyz))) \\ &= \{y\} \cup \{(\lambda x.(xyz))\} \cup Sub((xyz)) \\ &= \{y\} \cup \{(\lambda x.(xyz))\} \cup Sub(x) \cup Sub(y) \cup Sub(z) \\ &= \{y\} \cup \{(\lambda x.(xyz))\} \cup \{x\} \cup \{y\} \cup \{z\} = \{y, (\lambda x.(xyz)), x, y, z\} \end{aligned}$$

Outro conjunto importante para a sintaxe do cálculo lambda é o de variáveis livres. Uma variável é dita *ligante* se está do lado do  $\lambda$ . Em um termo  $\lambda x.M$ ,  $x$  é uma variável ligante e toda aparição de  $x$  em  $M$  é chamada de *ligada*. Se existir uma variável em  $M$  que não é ligante, então dizemos que ela é livre. Por exemplo, em  $\lambda x.xy$ , o primeiro  $x$  é ligante, o segundo  $x$  é ligado e  $y$  é livre.

O conjunto de todas as variáveis livres em um termo é denotado por  $FV$  e definido da seguinte forma:

**Definição 0.5** (Multiconjunto de variáveis livres).

1. (Base)  $FV(x) = \{x\}$ , para todo  $x \in V$
2. (Aplicação)  $FV((MN)) = FV(M) \cup FV(N) \cup \{(MN)\}$
3. (Abstração)  $FV((\lambda x.M)) = FV(M) \setminus \{x\}$

Exemplos:

1. Seja o termo  $\lambda x.\lambda y.xyz$ , vamos calcular seus subtermos:

$$\begin{aligned}
 FV(\lambda x.\lambda y.xyz) &= FV(\lambda y.xyz) \setminus \{x\} \\
 &= FV(xyz) \setminus \{y\} \setminus \{x\} \\
 &= FV(x) \cup FV(y) \cup FV(z) \setminus \{y\} \setminus \{x\} \\
 &= \{x, y, z\} \setminus \{y\} \setminus \{x\} \\
 &= \{z\}
 \end{aligned}$$

- 2.

Vamos definir os termos fechados da seguinte forma:

**Definição 0.6.** O  $\lambda$ -termo  $M$  é dito *fechado* se  $FV(M) = \emptyset$ . Um  $\lambda$ -termo fechado também é chamado de *combinador*. O conjunto de todos os  $\lambda$ -termos fechados é chamado de  $\Lambda^0$ .

Os combinadores são muito utilizados na *Lógica Combinatória*, mas vamos explorá-los mais a frente.

### 0.3 Conversão

No cálculo Lambda, é possível renomear variáveis ligantes/ligadas, pois a mudança dos nomes dessas variáveis não muda na sua renomeação. Por exemplo,  $\lambda x.x^2$  e  $\lambda u.u^2$  podem ser utilizadas de forma igual, mesmo que com nomes diferentes. Podemos definir então essa renomeação

**Definição 0.7.** Seja  $M^{x \rightarrow y}$  o resultado da troca de todas as livre-ocorrências de  $x$  em  $M$  por  $y$ . A relação de renomeação é expressa pelo símbolo  $=_\alpha$  e é definida como:  $\lambda x.M =_\alpha \lambda y.M^{x \rightarrow y}$ , dado que  $y \notin FV(M)$  e  $y$  não seja ligante em  $M$ .

Podemos estender essa definição para a definição do renomeamento, chamado de  $\alpha$ -conversão.

**Definição 0.8** ( $\alpha$ -conversão).

1. (Renomeamento)  $\lambda x.M =_\alpha \lambda y.M^{x \rightarrow y}$
2. (Compatibilidade) Sejam  $M, N, L$  termos. Se  $M =_\alpha N$ , então  $ML =_\alpha NL$ ,  $LM =_\alpha LN$  e, para um  $z$  qualquer,  $\lambda z.M = \lambda z.N$
3. (Reflexividade)  $M =_\alpha M$
4. (Simetria) Se  $M =_\alpha N$ , então  $N =_\alpha M$
5. (Transitividade) Se  $L =_\alpha M$  e  $M =_\alpha N$ , então  $L =_\alpha N$

A partir dos pontos (3), (4) e (5) dessa definição, é possível dizer que a  $\alpha$ -conversão é uma relação de equivalência, chamada de  $\alpha$ -equivalência.

## 0.4 Substituição

Podemos definir agora a substituição de um termo por outro da seguinte forma:

**Definição 0.9** (Substituição).

1.  $x[x := N] \equiv N$
2.  $y[y := x] \equiv y$ , se  $x \neq y$
3.  $(PQ)[x := N] \equiv (P[x := N])(Q[x := N])$
4.  $(\lambda y.P)[x := N] \equiv (\lambda z.P^{y \rightarrow z})[x := N]$  se  $\lambda z.P^{y \rightarrow z}$  é  $\alpha$ -equivalente a  $\lambda y.P$  e  $z \notin FV(N)$

A notação  $[x := N]$  é uma meta-notação, pois não está definida na sintaxe do cálculo lambda.

## 0.5 Beta redução