Universidade Federal do Rio Grande do Norte
Instituto Metrópole Digital
Bacharelado em Tecnologia da Informação
Fundamentos Matemáticos da Computação II

Estudo dirigido do conteúdo da Unidade 2

Autor: Yuri Alessandro Martins

Natal/RN
10 de maio de 2016

Sumário

1 Teoria dos Conjuntos

1.1 Os Axiomas de Zermelo-Frankel

1.1.1 Extensionalidade

Para quaisquer conjuntos A, B:

$$A = B \iff (\forall x)(x \in A \iff x \in B)$$

1.1.2 Emptyset

Garante que existe um conjunto vazio ($\emptyset$).

$$\exists x \forall y \ (y \ \notin \ x)$$

1.1.3 Pairset

Para todo a e b, existe o conjunto $\{a, b\}$.

$$\forall a \forall b \ \exists w \forall x (x \in w \iff x = a \ \vee \ x = b)$$

1.1.4 Separation

Para cada condição $P(x)$,

$$\forall a \exists w \forall x (x \in w \iff x \in a \ \wedge \ P(x))$$

Um problema de usar somente esses últimos três axiomas é que só somos capazes de formar conjuntos com cardinalidade ≤ 2.

- ZF4 (1.1.4) é um <u>axiom-scheme</u>. Isto é, possui infinitos axiomas dentro dele, já que para cada P(x) estamos formando um novo axioma.

- Usando os axiomas anteriores, é possível representarmos algumas coisas como conjuntos:

  - (x, y) $\triangleq$ { {x}, {x, y}}
  - A $\setminus$ B $\triangleq$ { x $\in$ A | x $\notin$ B }
  - $A \cap B \triangleq \{x \in A \mid x \in B\}$

1.1.5 Powerset $\wp$

Para cada conjunto a, existe um conjunto b, onde os elementos de b são subconjuntos de a.

$$\forall a \exists p \forall w (w \in p \iff \forall x (x \in a \implies x \in b))$$

Esse é o conjunto $\wp$(a).

Aqui $x \in a$ é uma abreviação de $(\forall t)[t \in x \implies t \in a]$. O Axioma da Extensionalidade (1.1.1) implica que para cada a, apenas um conjunto b pode satisfazer a defininção do Powerset; Nós podemos chamar **Conjunto Potência** de a e denotá-lo como:
$\wp(a) \triangleq \{x \mid Set(x) \& x \in a\}$

Algumas propriedades interesantes:
$\wp(\emptyset) = \{\emptyset\}$
$\wp(\{\emptyset\}) = \{\emptyset \, , \{\emptyset\}\}$

Exercício: Para cada conjunto A, existe um conjunto B cujo membros são exatamente singletons dos membros de A:

$$x \in B \iff (\exists t \in A)[x = \{t\}]$$

1.1.6 Unionset

Corresponde ao conjunto $\cup a$.

$$\forall a \exists u \forall x (x \in u \iff (\exists e \in a)[x \in e])$$

Ex: $\cup \emptyset = \cup \{\emptyset\} = \emptyset$

- $a \cup b = \cup \{a, b\}$
  Usando os axiomas ZF2 (1.1.2) e ZF5 (1.1.6)

$$t \in A \cup B \iff (\exists X \in \{A, B\})[t \in X]$$
$$t \in A \cup B \iff t \in A \vee t \in B$$

- $a \times b \triangleq \{w \in S \mid \exists x \exists y (w = (x, y) \wedge x \in a \wedge x \in b)\}$
  Onde $S = \wp(\wp(a \cup b))$
- sigletonset $\triangleq \{x \in \wp a \mid (\exists t \in a)[x = \{t\}]\}$
- $\cap a \triangleq \{x \in \cup a \mid (\forall e \in a)[x \in e]\}$

1.1.7 Infinity Axiom

$\exists I(\emptyset \in I \land \forall x(x \in I \implies \{x\} \in I))$ ou
$\exists I(\emptyset \in I \land \forall x(x \in I \implies x \cup \{x\} \in I))$

Esse axioma é garantido pois

$$\begin{aligned} \{x\} &\neq x \\ x \cup \{x\} &\neq x \end{aligned}$$

Com ele, somos capazes de montar o seguinte conjunto infinito:
I $= \{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}, \cdots\}$

1.2 Relações, Funções e Funções parciais na ZFC

1.2.1 Definindo um par ordenado

A operação de par do Kuratowski:
$(x, y) = \{\{x\}, \{x, y\}\}$, como vimos anteriormente na seção 1.1.4

PROOF AND MORE DETAILS...

1.2.2 Relações

Def: Sejam A, B conjuntos, R é uma relação entre A e B se $R \subseteq A \times B$.
Dessa forma,

- $f(a) = b \rightsquigarrow (a, b) \in f$

- União Disjunta: $A \uplus B = (\{0, a\} \times A) \cup (\{1, b\} \times B)$

Sendo R uma relação sobre o conjunto $\mathbb{N}(R \subseteq A \times A), R$ pode ser:

$$\begin{aligned} & xRx : Reflexiva \rightsquigarrow \text{``} =,\ \leq,\ \geq,\ \subseteq'' \\ xRy \implies & yRx : Simetrica \rightsquigarrow \text{``} ='' \\ xRy \land yRz \implies & xRz : Transitiva \rightsquigarrow \text{``} =,\ \leq,\ \geq,\ <,\ >,\ \subseteq'' \end{aligned}$$

Ainda existem outras propriedas como essas, como a Antireflexiva ou Antisimétrica.

1.2.3 Relações de Equivalência

Uma relação sobre um conjunto A é chamada **relação de equivalência** se ela for reflexiva, simétrica e transitiva.

O conjunto de todos os elementos que são relacionados a um elemento a de A é chamado de classe de equivalência de a. Isso implica que:

$$\cup[a] = A$$

- $[a] \cap [b] = \emptyset quando [a] \neq [b]$

Uma partição de um conjunto S é uma coleção de subconjuntos disjuntos não vazios de S. A união de todas as partições resulta, portanto, em S. Em outras palavras, os subconjuntos A_i formam partições de S se e somente se

$$\begin{aligned} A_i &\neq \emptyset \\ A_i \cap A_j &= \emptyset, quando\ i \neq j \\ \cap A_i &= S \end{aligned}$$

Podemos definir classes de equivalência como:

$$\begin{aligned} [x/\backsim] &\triangleq \{a \in A \mid x \backsim a\} \\ [A/\backsim] &\triangleq \{c \in \wp(A) \mid \exists x\ C = [x/\backsim]\} \end{aligned}$$

Seja $x, y \in A$, e $\backsim$ uma relação de equivalência no A:

$$\begin{aligned} [x/\backsim] = [y/\backsim] &\iff x \backsim y \\ [x/\backsim] = [y/\backsim] &\iff \begin{cases} [x/\backsim] & \text{se } x \backsim y \\ \emptyset & \text{se não} \end{cases} \\ \cup\{[x/\backsim] \mid x \in a\} &= A \end{aligned}$$

1.2.4 Funções parciais na ZFC

O conceito de funções parciais remete a ideia de uma função em que nem todos os x possuem uma $f(x)$. B

$$\begin{aligned} &f : \mathbb{N} \to \mathbb{N} && \text{Domínio da função} \\ &f(x) = \sqrt[2]{x} && x = 3 \text{ não possui uma saída bem definida nesse domínio.} \end{aligned}$$

1.3 Currying

Dada uma f do tipo $f : (X \times X) \to Z$, então a técnica de **currying** a torna $(f) : X \to (Y \to Z)$. Isto é, currying torna um paramêtro do tipo X e retorna uma função do tipo $Y \to Z$.

Achar um $\phi((x, y) \to A \rightarrowtail\!\!\!\!\rightarrow (x \to (y \to A)))$
$\phi(F) = G$, onde G é definida pela,
$G(x)(y) = g$, ode g é definida pela,
$g(y), f(x, y)$

1.4 Cardinais

Seja A um conjunto. O que é $|A|$?

c1. $A =_c |A|$

c2. $A =_c B \iff |A| = |B|$

c3. para todo conjunto de conjuntos $\in$,
$\{|x| \mid x \in \in\}$ é conjunto.

Aqui estaremos definindo funções cardinais fracas (**weak**). Isso porquê seria necessário provar o c2, algo extremamanete complicado agora. Portanto, podemos o resumir como:

c2. $A =_c B \iff |A| =_c |B|$

Sejam κ, λ, μ números cardinais:

- $\kappa + \lambda \triangleq_c \kappa \uplus \lambda$
- $\kappa.\lambda \triangleq_c \kappa \times \lambda$
- $\kappa^\lambda \triangleq_c (\kappa \to \lambda)$

1.5 Os Axiomas de Peano

Structed set: $(\mathbb{N}; 0; S)$, onde $0 \in \mathbb{N}$ e $S : \mathbb{N} \to \mathbb{N}$

$$
\begin{aligned}
0 &\in \mathbb{N} \\
S &: \mathbb{N} \to \mathbb{N} \\
S &: \mathbb{N} \rightarrowtail \mathbb{N} \\
(\forall x \in \mathbb{N})&[S_n \neq 0] \\
(\forall x \subseteq \mathbb{N})&[[0 \in X \land (\forall n \in \mathbb{N})[n \in X \implies S_n \in X]] \implies X = \mathbb{N}]
\end{aligned}
$$

O axioma de peano 5 é o que nos permite realizar indução matemática. Observe:

$$
\begin{aligned}
\forall x \subseteq \mathbb{N} &\ \textit{corresponde a}\ \textbf{base}. \\
(\forall n \in \mathbb{N})[n \in X \implies S_n \in X] &\ \textit{corresponde ao}\ \textbf{passo indutivo}. \\
n \in X &\ \textit{corresponde a}\ \textbf{hipotése indutiva}.
\end{aligned}
$$

1.6 Teorema da Recursão

Theorem. *Sejam:* $(\mathbb{N}, 0, S)$ *um sistema de naturais conjunto* E.
$a \in E$
$h : E \to E$
Então existe $f : \mathbb{N} \to E$
tal que: $f(0) = aef(S_n) = h(f(n))$.

1.7 Os Naturais na ZFC

Para estabelecermos os Naturais na ZFC, temos que garantir duas coisas:

- Existência de $\mathbb{N}$
- Singularidade de $\mathbb{N}$

Para tal, iremos precisar do Teorema da Recursão (1.6).

1.7.1 Existência de $\mathbb{N}$

Seja $J =$ todos os conjuntos X tal que satisfaz o ZF7 (1.1.7)
$J = \{X \in \wp I \mid \emptyset \in X \land (\forall x \in X)[S \in X]\}$

Seja $\mathbb{N} \triangleq \cap J$

Seja $0_1 = \emptyset$

[1]Visiste 2 para λ-Calculus e entender melhor esse ponto

Seja $S_1 = \lambda x.\{x\}$ [1]

Seja $0_2 = \emptyset$
Seja $S_2 = \lambda x.x \cup \{x\}$ [2]

Seja $\mathbb{N} \triangleq \cap J$

Encaixando com os Axiomas de Peano:

1. $(\forall x \in J)[\emptyset \in X]$, então $\emptyset \in \cap J$ e $\emptyset \in \mathbb{N}$
2. Também, pela própria definição de J
3. $a \neq b \iff S_a \neq S_b$ ou $a \neq b \iff \{a\} \neq \{b\}$
4. $\forall x\{x\} \neq \emptyset$
5. Seja $X \subseteq \mathbb{N}$, tal que
   0 $\in$ X
   $(\forall x \in X)[S_X \in X]$
   Seja $n \in \mathbb{N}$
   $\exists p : x = \{p\}$
   $->$ Mesmo que $\{p\} \in \cap J = \mathbb{N}$
   $->$ Mesmo que $n \in X$ e $x \geq \mathbb{N}$

Basicamente, podemos descrever $\mathbb{N}$ de duas maneiras, agora:

0 $\emptyset = \emptyset$
1 $\{\emptyset\} = \{0\}$
2 $\{\{\emptyset\}\} = \{1\}$
3 $\{\{\{\emptyset\}\}\} = \{2\}$
$\vdots$
S = λx.$\{x\}$

0 $\emptyset = \emptyset$
1 $\{\emptyset\} = \{0\}$
2 $\{\emptyset, \{\emptyset\}\} = \{0, 1\}$
3 $\{\emptyset, \{\emptyset\} \{\{\emptyset\}\}\}$
$\vdots$
S = $\lambda x.x \cup \{x\}$

1.7.2 Singularidade de $\mathbb{N}$

"$\mathbb{N}$ is unique up to isomorphism:"
$\pi : (\mathbb{N}_1; 0_1; S_1) \rightarrowtail\!\!\!\!\rightarrow (\mathbb{N}_1; 0_2; S_2)$.

$$\pi(0_1) = 0_2$$

$$\pi(S_1 n_1) = S_2 \pi(n_1)$$

Se traçarmos um paralelo com o Teorema da Recursão (1.6), para tentarmos provar a singularidade de $\mathbb{N}$, podemos realizar as seguintes associações:

[2] Veja nota 1

- $\mathbb{N}$: $\mathbb{N}_1$
- E: $\mathbb{N}_2$
- a: 0_2
- h: S_2

Demonstração. $\pi : \mathbb{N}_1 \implies \mathbb{N}_2$
$\pi[\mathbb{N}_1] = \mathbb{N}_2$

- $0_2 \in \pi[\mathbb{N}_1]$
  –> Como $\pi(0_1) \implies S_2 n_2 \in \pi[\mathbb{N}_1]$
- $n_2 \in \pi[\mathbb{N}_1]$
  –> Suponha que $n_2 \in \pi[\mathbb{N}_1]$ –> H.I
  –> $(\exists n_1 \in \mathbb{N}_1)[\pi(n_1) = n_2]$
  –> $\pi(S_1 n_1) = S_2(\pi(n_2))$ que $= S_2 n_2$

MAIS COISA AQUI DEPOIS...[?]

□

1.8 String Recursion

- Dado [ ] $\in [\mathbb{N}]$
- Se $n \in \mathbb{N}$, e $L \in [\mathbb{N}]$, então $(n : L) \in [\mathbb{N}]$

Exemplo: 2:3:4:[ ] = [2,3,4]

Alguns exemplos de funções recursivas que podemos definir utilizando String Recursion:

$$iszero : [\mathbb{N}] \to \mathbb{B}$$
$$iszero\ 0 = true$$
$$iszero\ S_n = false$$

$$empty : [\mathbb{N}] \to \mathbb{B}$$
$$empty\ [\ \ ] = true$$
$$empty\ (x : x_s) = false$$

$$
\begin{aligned}
++ : [\mathbb{N}] \rightarrow [\mathbb{N}] &\rightarrow [\mathbb{N}] \\
[\ \ ]\ ++\ y_s &= y_s \\
(x : x_s)\ ++\ y_s &= x : (x_s ++ y_s)
\end{aligned}
$$

$$
\begin{aligned}
Ex : [1,2] ++ [6,7,8,9] &= 1 : 2 : [\ \ ] ++ [6,7,8,9] \\
&= 1 : (2 : [\ \ ] ++ [6,7,8,9]) \\
&= 1 : (2 : ([\ \ ] ++ [6,7,8,9])) \\
&= 1 : 2 : [6,7,8,9] \\
&= [1,2,6,7,8,9]
\end{aligned}
$$

$$
\begin{aligned}
reverse : [\mathbb{N}] &\rightarrow [\mathbb{N}] \\
reverse[\ \ ] &= [\ \ ] \\
reverse[x] &= [x] \\
revese(x : x_s) &= reversexs ++ [x]
\end{aligned}
$$

$$
\begin{aligned}
\sqsubseteq : [\mathbb{N}] &\rightarrow [\mathbb{N}] \rightarrow \mathbb{B} \\
(x : x_s) \sqsubseteq [\ \ ] &= false \\
'[\ \ ] \sqsubseteq y_s &= true \\
(x : x_s) \sqsubseteq (y : y_s) &= (x = y) \wedge xs \sqsubseteq ys
\end{aligned}
$$

$$
\begin{aligned}
Ex : [2,3,4,5] \sqsubseteq [2,3,5,7] &= (2 = 2) \wedge ([3,4,5] \sqsubseteq [3,5,7]) \\
&= (3 = 3) \wedge ([4,5] \sqsubseteq [5,7]) \\
&= (4 = 5) \wedge ([5] \sqsubseteq [7]) \\
&= \text{FALSE}
\end{aligned}
$$

$$
\begin{aligned}
\in : \mathbb{N} &\rightarrow [\mathbb{N}] \rightarrow \mathbb{B} \\
n \in x[\ \ ] &= false \\
x \in (x : x_s) &= (n = x) \vee (n \in x_s)
\end{aligned}
$$

$$
\begin{aligned}
find : \mathbb{N} \to [\mathbb{N}] \to \mathbb{B}& \\
findn[\ \ ] &= 0 \\
findn(n : nx) &= 0 \\
findn(x : x_s) &= 1 + find\ n\ x_s
\end{aligned}
$$

$$
\begin{aligned}
sum : [\mathbb{N}] \to \mathbb{N}& \\
sum[\ \ ] &= 0 \\
sum(x : xs) &= x + sumxs
\end{aligned}
$$

$$
\begin{aligned}
\oplus : [\mathbb{N}] \to [\mathbb{N}] \to [\mathbb{N}]& \\
'[\ \ ] \oplus y_s &= y_s \\
x_s \oplus [\ \ ] &= x_s \\
(x : x_s) \oplus (y : y_s) &= (x + y) : (x_s \oplus y_s)
\end{aligned}
$$

$$
\begin{aligned}
circle : (\mathbb{N} \to \mathbb{N} \to \mathbb{N}) \to [\mathbb{N}] \to [\mathbb{N}] \to [\mathbb{N}]& \\
circlef[\ \ ]y_s &= [\ \ ] \\
fxs[\ \ ] &= [\ \ ] \\
f(x : x_s)(y : y_s) &= [fxy] : (circle\ f\ x_s\ y_s)
\end{aligned}
$$

2 λ-Calculus

2.1 O conjunto de λ-termos

Sendo $\Lambda = \lambda$-termos;

$$
\begin{aligned}
X &\in \Lambda \\
s, t \in \Lambda &\implies (s\ t) \in \Lambda \\
x \in var, t \in \Lambda &\implies \lambda X.t \in \Lambda
\end{aligned}
$$

2.2 Conversões α, β e η

2.2.1 Conversão α

Determina que a escolha da variável ligada, na abstração lambda, não importa (normalmente):

$\lambda x.x =_\alpha \lambda y.y$

$\lambda x.\lambda x.x =_\alpha \lambda y.\lambda x.x$ Note que isso não poderá ser transformado em $\lambda y.\lambda x.y$

Primeiro, quando alfa-conversão atua em uma abstração, as únicas ocorrências de variáveis que podem ser renomeados são aqueles que são vinculados a esta mesma abstração. No segundo exemplo, portanto:

$\lambda x.\lambda x.x \neq_\alpha \lambda y.\lambda x.y$ Este último tem um significado diferente do original.

Em segundo lugar, uma conversão α não é possível se isto irá resultar em uma variável sendo capturada por uma abstração diferente. Por exemplo, se substituirmos x com y em $\lambda x.\lambda y.x$, nós obteríamos $\lambda y.\lambda y.y$, que tem um significado diferente da expressão anterior.

2.2.2 Redução β

Redução β é a ideia de aplicar uma função. Por exemplo, se temos $f(x) = x * 2$, para $x = 2$ aplicamos o valor a função que irá ficar como $f(2) = 2 * 2$. Essa é basicamente a ideia da redução β.

$(\lambda x.x * 2)\ 2 =_\beta 2 * 2$

2.2.3 Redução η

Eta-conversão expressa a ideia de extensionalidade, que neste contexto é que duas funções são as mesmas se e somente se eles dão o mesmo resultado para todos os argumentos.[3]

2.3 Booleanos naturais no Λ

$$\lambda y.x = \lambda x.(\lambda y.x) := true := fst$$
$$\lambda x.y = \lambda x.(\lambda y.y) := false := snd$$

[3]Sujeito a severas mudanças no futuro. Visite 3 para saber mais sobre.

2.4 Combinators I, K, B, S

Combinadores[4]

$$
\begin{aligned}
I &= \lambda x.x \\
B &= \lambda x \lambda y.\lambda z.x(yz) && \text{“Composition”} \\
S &= \lambda x \lambda y.\lambda z.\lambda xz(yz)
\end{aligned}
$$

3 Política de Colaboração

Você é capaz de alterar o conteúdo desse documento, para corrigir erros, melhorar suas explicações ou dar dicas/exemplos adicionais. Esse foi o objetivo desde começo.

Visita a página remota do documento para obter sua versão mais atualizada e/ou colaborar também.

Como base foram utilizados os livros “Notes on Set Theory” [?] e “Classic Set Theory” [?]. Caso você queira continuar usando-os como base para esse documento, sinta-se a vontade.

[4] A melhorar bruscamente