

Teoria da Computação

Aula Teórica 3: “Tudo é um Conjunto”

António Ravara

Departamento de Informática

13 de Março de 2019

Teoria de Conjuntos?!

A ideia (matemática)

“Tudo é um conjunto”, ou mais precisamente, *tudo pode ser modelado por um conjunto*.

O que é?

Noções que definem a entidade “conjunto”, as suas propriedades e as operações sobre tais entidades.

Para que serve?

- ▶ Para dar fundações à Matemática
- ▶ Para modelar conceitos de (outras) disciplinas científicas
- ▶ Juntamente com a Lógica, é o “cálculo” da Informática (à semelhança da Análise Matemática para, e.g., a Física): sistemas e tipos de dados são modeláveis com conjuntos.

Teoria de Conjuntos de Cantor

A Teoria de Conjuntos que vimos foi definida por Cantor, usando a Lógica de Primeira Ordem, seguindo Frege.

Lógica da Teoria de Conjuntos

Considere-se uma *Assinatura de Primeira Ordem* que inclua:

- ▶ a constante $\emptyset$;
- ▶ a função unária sobre conjuntos $\setminus$;
- ▶ as funções binárias sobre conjuntos $\cup$, $\cap$, $\times$;
- ▶ os predicados $=$, $\in$, $\subset$

A Lógica da Teoria de Conjuntos é a linguagem que permite definir conjuntos por compreensão.

Lógica da Teoria de Conjuntos

A Lógica da Teoria de Conjuntos é a linguagem que permite definir conjuntos por compreensão.

Definição por compreensão

Se P for uma fórmula da Lógica da Teoria de Conjuntos,

$$S \stackrel{\text{def}}{=} \{x \mid P(x)\}$$

é um conjunto.

Note-se que a variável x está implicitamente universalmente quantificada, sendo o seu âmbito definido pelas chavetas.

Tudo é um conjunto?!

A Teoria é *inconsistente*, como provou Russel (com o paradoxo do Bibliotecário).

O problema: o conjunto de todos os conjuntos...

Seja R o conjunto de todos os conjuntos que não se contêm a si mesmos:

$$R \stackrel{\text{def}}{=} \{X \mid X \notin X\}$$

A propriedade parece fazer sentido – o conjunto vazio, por exemplo, não é um membro de si mesmo...

Há obviamente inúmeros exemplos de conjuntos que não são membros de si mesmos, logo

Proposição

O conjunto R , se existe, não é vazio.

O paradoxo de Russel

$$R \stackrel{\text{def}}{=} \{X \mid X \notin X\}$$

Consegue-se verificar que $R \in R$?

- ▶ Por definição de R , se $R \in R$ então tem-se que $R \notin R$, o que é absurdo.
- ▶ Assumindo antes que $R \notin R$, concluímos que não podemos ter que $R \notin R$, ou seja tem que se verificar que $R \in R$, o que é igualmente absurdo!

A inconsistência

Acabámos de provar que:

$R \in R$ se, e só se, $R \notin R$

Como a propriedade é obviamente falsa (inconsistente!), a Teoria de Conjuntos não pode permitir definir conjuntos como R .

- ▶ O problema está na auto-referência.
- ▶ Expressões como $\{X \mid X \notin X\}$ não podem ser aceites (“legais”) na linguagem da Teoria de Conjuntos.
- ▶ O problema foi identificado por Russel e resolvido por Zermelo e Fraenkel.

Teoria de Conjuntos ZFC (de Zermelo-Fraenkel-Cantor)

Como conseguir que a compreensão fique bem definida?

O princípio da Separação

Dada uma propriedade P da Lógica da Teoria de Conjuntos,

$$\{x \in S \mid P(x)\}$$

é um conjunto bem definido, se S é um conjunto bem definido.

Exemplos e contra-exemplos

- ▶ O conjunto dos naturais pares $\{n \in NAT \mid n \% 2 = 0\}$ está bem definido.
- ▶ O conjunto dos conjuntos não vazios $\{S \mid S \neq \emptyset\}$ não está bem definido.

Uma operação “mal definida”

Complemento de um conjunto

O complemento “absoluto” de um conjunto S , que se escreve $\overline{A}$, é o conjunto de todos os elementos que não estão em S .

- ▶ Esta “definição” não respeita o princípio da Separação,
- ▶ logo o complemento “absoluto” de um conjunto não é definível em ZFC.

Modelação de Sistemas

ZFC permite modelar entidades como valores naturais ou booleanos, relações, funções, bases de dados e até algoritmos.

Uma lista telefónica

Pode ser entendida como uma relação:

$$PHONELIST \stackrel{\text{def}}{=} FIRSTNAME \times LASTNAME \times PHONENUM$$

- ▶ $FIRSTNAME \stackrel{\text{def}}{=} STRING$
- ▶ $LASTNAME \stackrel{\text{def}}{=} STRING$
- ▶ $PHONENUM \stackrel{\text{def}}{=} ALG^9$
sendo $ALG \stackrel{\text{def}}{=} \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

Note-se que um valor em *STRING* é uma sequência de caracteres. Em *STRING* estão sequências de todos os comprimentos possíveis.

Um valor na lista telefónica é um triplo:

$$(Ivo, Ives, 212212212) \in PHONELIST$$

Modelação de Tipos de Dados

Uma lista telefónica é um conjunto de triplos.

- ▶ Mas esta lista pertence a que conjunto?
- ▶ Ou seja, como definir o conjunto das listas telefónicas?

Listas telefónicas

$LISTAS_TEL \stackrel{\text{def}}{=} \wp(PHONELIST)$

sendo

$PHONELIST \stackrel{\text{def}}{=} FIRSTNAME \times LASTNAME \times PHONENUM$

Powerset

Dados conjuntos A e S , tem-se que

$$A \in \wp(S) \Leftrightarrow A \subseteq S$$

Funções em ZFC

No contexto da Teoria de Conjuntos, uma função é um conjunto de pares ordenados.

Notação

Seja (x, y) um elemento de dado conjunto que define uma função. Escreve-se frequentemente o par ordenado na forma $x \mapsto y$, de forma a sublinhar que x é aplicado em y .

Extensão da função

Conjunto (de pares ordenados) que define a função.

Funções em ZFC

Define-se uma função f como um conjunto

$$f \stackrel{\text{def}}{=} \{x \mapsto y \in D \times C \mid P(x, y)\}$$

- ▶ sendo $P(x, y)$ uma fórmula da Lógica da Teoria de Conjuntos, estabelece uma relação entre o argumento x e o resultado y ,
- ▶ onde D é o domínio e C o contra-domínio.

Em alternativa

$$f \subseteq D \times C \quad \text{ou} \quad f \in D \rightarrow C$$

$$f \stackrel{\text{def}}{=} \{x \mapsto y \mid P(x, y)\}$$

Um exemplo

Negação: função $NOT \in BOOL \rightarrow BOOL$

$$NOT \stackrel{\text{def}}{=} \{ TRUE \mapsto FALSE, FALSE \mapsto TRUE \}$$

Tem-se então que:

- ▶ $NOT(TRUE) = FALSE$
- ▶ $NOT(FALSE) = TRUE$

Um exercício de modelação: uma lâmpada

Considere que tem dois estados (ON e OFF) e funções para a ligar (*turn_on*), desligar (*turn_off*) e verificar se está ligada (*status*).

- ▶ O conjunto de estados: $SLAMP = \{0, 1\}$
- ▶ A função $turn_on \in SLAMP \rightarrow SLAMP$

$$turn_on \stackrel{\text{def}}{=} \{0 \mapsto 1, 1 \mapsto 1\}$$

- ▶ A função $status \in SLAMP \rightarrow BOOL$

$$status \stackrel{\text{def}}{=} \{0 \mapsto FALSE, 1 \mapsto TRUE\}$$

- ▶ A estrutura que modela o sistema:

$$LAMP \stackrel{\text{def}}{=} (SLAMP, turn_on, turn_off, status)$$

Funções de selecção

Como “destruir” pares?

Para seleccionar elementos num par, usam-se as funções de projecção.

Dado um produto cartesiano $A \times B$, definem-se as funções

$$\pi_1 \stackrel{\text{def}}{=} \{((a, b) \mapsto a) \in (A \times B) \times A \mid (a, b) \in A \times B\}$$

$$\pi_2 \stackrel{\text{def}}{=} \{((a, b) \mapsto b) \in (A \times B) \times B \mid (a, b) \in A \times B\}$$

Logo, $\pi_1 \in (A \times B) \rightarrow A$ e $\pi_2 \in (A \times B) \rightarrow B$.

Por exemplo, $\pi_1((\text{“Sofia”}, 22255)) = \text{“Sofia”}$ e

$\pi_2((\text{“Sofia”}, 22255)) = 22255$.

Outro exercício de modelação: um Banco

Pretende-se definir uma entidade que é essencialmente
um conjunto de contas bancárias.

Uma conta bancária tem um nome, um número e um saldo.

O sistema: $ACC \stackrel{\text{def}}{=} NAME \times ACCNUM \times AMOUNT$

sendo:

$$NAME \stackrel{\text{def}}{=} STRING$$
$$ACCNUM \stackrel{\text{def}}{=} NAT$$
$$AMOUNT \stackrel{\text{def}}{=} NAT$$

Uma conta

$(Belmiro, 1000, 2000000000) \in ACC$

Operações no Banco

O Banco

$$ACC \stackrel{\text{def}}{=} NAME \times ACCNUM \times AMOUNT$$

permite saber o *saldo* de dada conta, *depositar* e *levantar* montantes arbitrários.

- ▶ A função $balance \in ACC \rightarrow AMOUNT$

$$balance \stackrel{\text{def}}{=} \{c \mapsto m \mid m = \pi_3(c)\}$$

- ▶ A função $deposit \in ACC \times AMOUNT \rightarrow ACC$

$$deposit \stackrel{\text{def}}{=} \{(c, m) \mapsto d \mid d = (\pi_1(c), \pi_2(c), balance(c) + m)\}$$

Operações no Banco

O Banco

$$ACC \stackrel{\text{def}}{=} NAME \times ACCNUM \times AMOUNT$$

A função *withdraw*

E se o valor a retirar é superior ao saldo?

Como $AMOUNT \stackrel{\text{def}}{=} NAT$, dados $n, m \in NAT$, tem-se por definição que $m - n = 0$ se $n > m$.

Logo, fica bem definida

$$withdraw \stackrel{\text{def}}{=} \{(c, m) \mapsto d \mid d = (\pi_1(c), \pi_2(c), balance(c) - m)\}$$

Operações no Banco

O Banco: $ACC \stackrel{\text{def}}{=} NAME \times ACCNUM \times AMOUNT$
onde agora $AMOUNT \stackrel{\text{def}}{=} INT$

A função *withdraw*

Só se permite levantar se o saldo for suficiente; senão aplica-se uma multa.

$$\begin{aligned} withdraw \stackrel{\text{def}}{=} \{ & (c, m) \mapsto d \mid \exists b. d = (\pi_1(c), \pi_2(c), b) \\ & \wedge ((balance(c) - m \geq 0) \rightarrow (b = balance(c) - m)) \\ & \wedge ((balance(c) - m < 0) \rightarrow (b = balance(c) - 5)) \} \end{aligned}$$

Note-se que o antecedente da segunda implicação é a negação da primeira. Logo, uma das implicações será verdadeira porque o antecedente é falso e a outra estabelece o valor de b .