

1 INTRODUÇÃO

1.1 Notas Históricas

1.2 Abordagem

1.3 Conceitos Básicos

1.4 Conclusões

1.5 Exercícios

1 INTRODUÇÃO

- Inicia com uma breve história do surgimento e do desenvolvimento dos conceitos, resultados e formalismos nos quais a Teoria da Computação é baseada.
- **Formalização dos conceitos de programa e de máquina**
 - Existem diferentes computadores, com diferentes arquiteturas, e existem diversos tipos de linguagens de programação,
 - Modelos matemáticos simples.
 - Programas e máquinas são tratados como entidades distintas, mas complementares e necessárias para a definição de computação.

1.1 Notas Históricas

- ◆ Ciência da Computação é o conhecimento sistematizado relativo à computação.
- ◆ Na antiga Grécia (século III A.C., no desenho de algoritmos por Euclides)
- ◆ Na Babilônia (complexidade\reducibilidade problemas).
- ◆ Interesse atual possui duas ênfases:
 - **ênfase teórica** - idéias fundamentais e modelos computacionais: da biologia (modelos para redes de neurônios), da eletrônica (teoria do chaveamento), da matemática (lógica), da lingüística (gramáticas para linguagens naturais).
 - **ênfase prática** - projeto de sistemas computacionais aplicando a teoria à prática.

No início do século XX, diversas pesquisas foram desenvolvidas com o objetivo de definir um modelo computacional suficientemente genérico, capaz de implementar qualquer *Função Computável*.

1901, Lista de Problemas não resolvidos; Equações Diofantinas: Equações de Coeficientes Inteiros tem raízes inteiras?

1928, David Hilbert, *Entscheidungsproblem*, Problema de Decisão - o qual consistia em encontrar um procedimento para demonstrar se uma dada fórmula no cálculo de proposições de primeira ordem era válida ou não. (Fórmula do cálculo de predicados no qual a quantificação é restrita às variáveis individuais)

1931, Kurt Gödel - *Incompleteness Theorem* (teorema da não completude), onde demonstrou que tal problema (mecanização do processo de prova de teoremas) não tinha solução.

- A classe de funções usada por Gödel foi a das funções primitivas recursivas (definidas por Dedekind em 1888).
- Gödel foi (aparentemente) o primeiro a identificar um formalismo para definir a noção de *procedimento efetivo*.

1936, Alonso Church, usou dois formalismos para mostrar que o problema de decisão não tem solução: λ -calculus e funções recursivas primitivas (Kleene, 1936).

Procedimentos efetivos - “São caracterizações tão gerais da noção do efetivamente computável quanto consistentes com o entendimento intuitivo usual” (Church).

1936, Alan Turing propôs, em um formalismo para a representação de procedimentos efetivos. Esse foi o primeiro trabalho a identificar programas escritos para uma “máquina computacional autômata”, como noções intuitivas de efetividade.

Desde então, muitos outros formalismos foram propostos, os quais possuem o mesmo poder computacional que as *funções recursivas* (ou *Cálculo Lambda*):

- Máquina de Turing (1936);
- Sistema Canônico de Post (1943);
- Algoritmo de Markov e a Linguagem *Snobol* (1954);
- Máquinas de Registradores (1963);
- RASP (*Random Access Stored Programs* - 1964).

Programa é definido como sendo um procedimento efetivo, pode ser descrito usando qualquer dos formalismos equivalentes. Ou seja, qualquer destes formalismos permite descrever todos os procedimentos possíveis que podem ser executados em um computador.

1.2 Abordagem

A abordagem desse livro combina a abordagem histórica com abordagens próximas dos sistemas computacionais modernos.

Parte dessa abordagem é inspirada, entre outros, no trabalho de Richard Bird em *Programs and Machines- An Introduction to the Theory of Computation* ([BIR76]).

1.3 Conceitos Básicos

Linguagem é um conceito fundamental no estudo da Teoria da Computação, pois se trata de uma forma precisa de expressar problemas, permitindo um desenvolvimento formal adequado ao estudo da computabilidade.

No dicionário de Aurélio, linguagem é definida como:

"o uso da palavra articulada ou escrita como meio de expressão e comunicação entre pessoas".

Esta definição não é suficientemente precisa para permitir o desenvolvimento matemático de uma teoria sobre linguagens.

As definições que seguem são construídas usando como base a noção de **Símbolo** ou **Caractere**, que é uma entidade abstrata básica, não sendo definida formalmente. **Letras e dígitos são exemplos de símbolos frequentemente usados.**

Definição 1.1 **Alfabeto.**

Um **Alfabeto** é um conjunto finito de *símbolos* ou *caracteres*.

- um conjunto infinito *não* é um alfabeto;
- o conjunto vazio *é* um alfabeto.

Exemplo 1.1

Os seguintes conjuntos são exemplos de alfabetos:

- $\{a, b, c\}$;
- conjunto vazio.

Os seguintes conjuntos não são exemplos de alfabetos:

- conjunto dos números naturais;
- $\{a, b, aa, ab, ba, bb, aaa, \dots\}$.

Definição 1.2**Cadeia de Símbolos, Palavra.**

Uma *cadeia de símbolos* sobre um conjunto Σ é uma sequência de zero ou mais símbolos de Σ justapostos.

- *cadeia de símbolos finita* é denominada de **Palavra**.
- ε denota a *cadeia vazia* ou palavra vazia

Definição 1.3**Comprimento ou Tamanho de uma Palavra.**

O *comprimento* ou *tamanho* de uma palavra w , representado por $|w|$, é o número de símbolos que compõem a palavra.

Definição 1.4**Prefixo, Sufixo, Subpalavra.**

Um *prefixo* (ou *sufixo*) de uma palavra é qualquer sequência inicial (ou final) de símbolos da palavra.

Uma *subpalavra* de uma palavra é qualquer sequência de símbolos contígua da palavra.

Exemplo 1.2 Palavra, Prefixo, Sufixo, Tamanho.

- abcb é uma palavra sobre o alfabeto $\{a, b, c\}$;
- $|abcb| = 4$ e $|\varepsilon| = 0$;
- Relativamente à palavra abcb, tem-se que:
 $\varepsilon, a, ab, abc, abcb$ são os prefixos;
 $\varepsilon, b, cb, bcb, abcb$ são os respectivos sufixos.
- Qualquer prefixo ou sufixo de uma palavra é uma sub-palavra.

Definição 1.5**Conjunto de todas as palavras**

Seja Σ um alfabeto, então o *conjunto de todas as palavras* Σ^* é definido indutivamente:

- $\varepsilon \in \Sigma^*$,
- Se $x \in \Sigma$, então $x \in \Sigma^*$
- Se u e v são palavras de Σ^* , então $uv \in \Sigma^*$

Σ^* , o conjunto de todas as palavras possíveis sobre Σ ;

Σ^+ , denota $\Sigma^* - \{ \varepsilon \}$.

Exemplo 1.6

Se $\Sigma = \{a, b\}$, então:

$$\Sigma^+ = \{a, b, aa, ab, ba, bb, aaa, \dots\}$$

$$\Sigma^* = \{\varepsilon, a, b, aa, ab, ba, bb, aaa, \dots\}$$

Definição 1.6**Linguagem Formal**

Uma **linguagem formal** ou simplesmente **linguagem** L é um conjunto de palavras sobre um alfabeto Σ . $L \subseteq \Sigma^*$

Exemplo 1.7**Linguagem Formal.**

Suponha o alfabeto $\Sigma = \{a, b\}$. Então:

- a) O conjunto vazio e o conjunto formado pela palavra vazia são linguagens sobre Σ . Obviamente, $\{\} \neq \{\varepsilon\}$.
- b) O conjunto de palíndromos (palavras que têm a mesma leitura da esquerda para a direita e vice-versa) sobre Σ é um exemplo de linguagem infinita.

$\varepsilon, a, b, aa, bb, aaa, aba, bab, bbb, aaaa, \dots$

Definição 1.7**Concatenação de Palavras.**

A **concatenação de palavras** é uma **operação** binária, definida sobre uma linguagem L , a qual associa a cada par de palavras uma palavra formada pela justaposição da primeira com a segunda. A operação satisfaz as seguintes propriedades:

- é **associativa**; $v(wt) = (vw)t = vwt$
- possui **elemento neutro** à esquerda e à direita, o qual é a palavra vazia. $\varepsilon w = w = w\varepsilon$
- não necessariamente é **fechada** em L ; **a concatenação de duas palavras de uma linguagem não necessariamente resulta em uma palavra da linguagem.**

Exemplo 1.10**Concatenação de Palavras.**

Suponha o alfabeto $\Sigma = \{a, b\}$.

Então, para as palavras $v = baaaa$ e $w = bb$, tem-se que:

- $vw = baaaabb$
- $v\varepsilon = v = baaaa$

Definição 1.8**Concatenação Sucessiva de uma Palavra.**

A *concatenação sucessiva de uma palavra* (com ela mesma) ou simplesmente *concatenação sucessiva*, representada na forma de um expoente (suponha w uma palavra):

w^n onde n é o número de concatenações sucessivas, é definida indutivamente a partir da operação de concatenação binária, como segue:

$$w^0 = \varepsilon;$$

$$w^n = w^{n-1}w, \text{ para } n > 0.$$

Exemplo 1.11**Concatenação Sucessiva.**

Sejam w uma palavra e a um símbolo. Então:

$$w^3 = www$$

$$w^1 = w$$

$$a^5 = aaaaa$$

$$a^n = aaa...a \quad (\text{o símbolo } a \text{ repetido } n \text{ vezes})$$

1.4 Conclusões

A origem do estudo da computação é milenar e se desenvolveu em diversas épocas e culturas, objetivando responder questões como:

- O que é uma solução computável?
- Quais são os limites do que pode ser computado?
- Existem problemas sem solução computacional?

Foi a partir do século XX que importantes contribuições ocorreram:

- de Hilbert (Problema de Decisão),
 - de Church (Cálculo Lambda e hipótese de Church)
 - de Turing (Máquina de Turing e Problema da Parada).
-
- O estudo da Teoria da Computação independe do estudo do Computador (hardware e software) como se conhece hoje.
 - Foram introduzidos alguns dos principais conceitos sobre os quais todo o estudo é construído, como alfabeto, palavra e linguagem formal.

1.5 Exercícios

Exercício 1.1

Elabore uma *linha de tempo* sobre o desenvolvimento do conceito de função computável.

Exercício 1.2

Em que se consistia o problema de Hilbert– *Entscheidungsproblem* e por que ele é sem solução?

Exercício 1.3

Qual a importância da *Tese de Church* e porque ela não é demonstrável?

Exercício 1.4

Marque os conjuntos que são alfabetos:

- a) Conjunto dos números naturais; []
- b) Conjunto dos números primos; []
- c) Conjunto das letras do alfabeto brasileiro; []
- d) Conjunto dos algarismos arábicos; []
- e) Conjunto dos algarismos romanos; []
- f) Conjunto { a, b, c, d }; []
- g) Conjunto das partes de { a, b, c }; []
- h) Conjunto das vogais; []
- i) Conjunto das letras gregas. []

Exercício 1.5

Dê os possíveis prefixos e sufixos de cada uma das seguintes palavras:

- a) *teoria*
- b) *universidade*
- c) *aaa*
- d) *abccba*
- e) *abcabc*

Exercício 1.6

Exemplifique, comprovando ou negando as seguintes propriedades algébricas da operação de concatenação de palavras:

- a) **Total** (suponha a operação definida sobre uma determinada linguagem)
- b) **Comutatividade**
- c) **Elemento neutro**
- d) **Associatividade**
- e) **Elemento absorvente**, genericamente definida como segue: para uma dada operação binária \oplus sobre um conjunto A , afirma-se que a operação \oplus possui *elemento absorvente* se existe $a \in A$ tal que, para qualquer $x \in A$ vale que: $a \oplus x = x \oplus a = a$
- f) **Elemento inverso**, genericamente definido como segue: para uma dada operação binária \otimes sobre um conjunto A , afirma-se que a operação \otimes possui *elemento inverso* se possui *elemento neutro* ε e, para qualquer $a \in A$ existe $\underline{a} \in A$ tal que: $a \otimes \underline{a} = \underline{a} \otimes a = \varepsilon$

Exercício 1.7

Quando se pode dizer que a estrutura algébrica da operação de concatenação de palavras de uma linguagem é análoga à estrutura da operação de adição nos naturais?

Exercício 1.8

Todos possíveis prefixos da palavra **abccba** são:

- a) { a, ab, abc, abcc, abccb, abccba }
- b) { a, ba, cba, ccba, bccba, abccba }
- c) { a, ab, abc, abccb, abccba }
- d) { a, ba, cba, ccba, abccba }
- e) Nenhuma das alternativas anteriores está correta.

Exercício 1.9

Em relação à palavra **abccba**, marque a alternativa correta:

- a) Os conjuntos de sufixos e de prefixos são o mesmo;
- b) O Alfabeto sobre o qual é gerado essa palavra só pode ser $\Sigma = \{a, b, c\}$;
- c) Existem exatamente três subpalavras que são prefixos e sufixos ao mesmo tempo;
- d) Existem exatamente duas subpalavras que são prefixos e sufixos ao mesmo tempo;
- e) Nenhuma das afirmativas anteriores está correta.