Teoria da Computação
Introdução e Conceitos Básicos

Alessandra Hauck & Tiago Leite

FATECS

Introdução

- **Linguagens formais**
  - Desenvolvida em 1950
  - Estudar e desenvolver teorias relacionadas a linguagem natural
  - Inclinou-se para as linguagens artificias
  - Linguagens ordinárias da ciência da computação
- A partir de então desenvolveu-se bastante
- O enfoque maior foi em aplicações de:
  - Análise léxica
  - Análise sintática
- Ex:
  - Linguagens de programação
  - circuitos digitais
  - expressões matemáticas
  - linguagens naturais...

Sintaxe

Sintaxe

Sintaxe é a parte da gramática que estuda a disposição das palavras na frase e a das frases no discurso, bem como a relação lógica das frases entre si.

- Exemplo de **sintaxe correta:**
  - José bebeu água.
  - Maria acabou a prova.
  - As flores são belas.
- Exemplo de **sintaxe errada:**
  - Água bebeu José.
  - A prova acabou Maria.
  - As flores é bela. (erro de concordância)
  - Os bandidos começaram a correrem. (erro de concordância)
  - Eu vou corre. (erro léxico)

Sintaxe

- Exemplo de **sintaxe correta:**
  - José bebeu água.
  - `if` $(a < 10)$ `then`
  - $A = 45$;
- Exemplo de **sintaxe errada:**
  - `then` $(a < 10)$ `if`
  - $45 = a$;
  - `if` $(10 > a)$ `then` , erro sintático?

Semântica

Semântica

- **Semântica** é a parte da gramática que estuda o significado das palavras na frase e a das frases no discurso
- Um erro semântico pode alterar completamente o sentido da frase
- A frase deve ser analisada como um todo para descobrir o significado de uma palavra
- Exemplo de semântica correta:
  - Eu caminho todos os dias ⇒ caminho: ato de andar
  - O caminho é longo ⇒ caminho: estrada
  - Vou colher flores ⇒ colher: pegar
  - A colher caiu no chão ⇒ colher: objeto
- Exemplo de semântica errada:
  - Recebi um xeque sem fundo ⇒ Esta frase não tem sentido, pois:
    - xeque: jogada de xadrez
    - cheque: papel moeda

Semântica

Exemplo de semântica correta:

- `int soma;` ⇒ valor inteiro
- `float soma;` ⇒ valor real
- `class soma;` ⇒ Tipo Abstrato de Dados (TAD)

Exemplo de semântica errada:

- A <u>seção</u> inicia-se as 20h ⇒ Esta frase não tem sentido, pois:
  - <u>seção</u>: divisão, repartição
  - <u>sessão</u>: reunião, encontro
- `int num;`
  `num = 2.4;` ⇒ Erro semântico ⇒ "*incompatible types*"
- `media = 52.5;` ⇒ Erro semântico ⇒ "*variable not defined*"

Sintaxe $\times$ Semântica

Sintaxe × Semântica

Sintaxe	**Semântica**
Reconhecido **antes** da semântica	Analisado **após** a sintática
Primeiro a receber tratamento adequado	Tratamentos mais elaborados
Tratamentos mais simples	É baseado em interpretações, logo mais subjetivo
Possui construções matemáticas bem e definidas universalmente reconhecidas (Gramáticas de Chomsky)	ASSOCIADO: com uma interpretação do seu significado
LIVRE: sem significado associado	
Manipula símbolos	

Sintaxe e Semântica

- Para resolver um problema real é necessário dar uma interpretação semântica para os símbolos, por ex:
  - `int a;` $\Rightarrow$ o símbolo `int` representa os números inteiros

Dizer que um programa está...

- Sintaticamente "**errado**"
  - Não existe num programa essa expressão!
  - O correto é: o texto escrito não é aceito pela linguagem
- Sintaticamente "**correto**"
  - O texto é aceito pela linguagem
  - ATENÇÃO: pode não ser o programa que o programador esperava escrever!
  - Logo dizemos que e o programa é sintaticamente "válido"

Sintaxe e Semântica

- Programa "correto" ou "errado" $\Rightarrow$ Vai depender se o mesmo modela corretamente:
  - a linguagem regular e
  - o comportamento desejado
- ATENÇÃO: Dentro das linguagens artificias, definir os limites entre sintaxe e semântica podem não ser tão fáceis
- Esta disciplina será centrada na analise sintática

Conceitos Básicos

Introdução

- **Linguagem**
  - Segundo o **dicionário**, uma linguagem é o uso da palavra articulada ou escrita como meio de expressão e comunicação entre pessoas
  - Não é suficiente preciso para definir modelos matemáticos
  - Então faremos algumas definições formais para nosso estudo
- Para definir linguagem formal:
  - Precisamos de um alfabeto; e
  - uma cadeia de caracteres, ou palavras

Alfabeto

Alfabeto (Σ)

- **Alfabeto (Σ):** Conjunto finito de símbolos ou caracteres
- Notação: Σ
- Símbolos ou caracteres: Entidade abstrata básica
  - **Ex:** letras, dígitos...
- Exemplos de alfabetos:
  - $\{a, b, c\}$
  - $\emptyset \Rightarrow$ conjunto vazio
- Não são alfabetos:
  - $\mathbb{N} \Rightarrow$ conjunto dos naturais
  - $\{a, aab, bbb, aba, \cdots\}$

Alfabeto (Σ)

- **Alfabeto em Linguagem de programação** $\Rightarrow$ Conjunto de todos os símbolos utilizados na linguagem de programação. Ex.:
  - letras
  - dígitos
  - Caracteres especiais: $>$, $<$, $<=$, $*$ ...
  - Espaços ou brancos
- **Alfabeto binário** $\Rightarrow$ Domínio de valores de um bit
  - Podemos usar $\{a, b\}$ ou $\{0, 1\}$ ou $\{V, F\}$

Palavra

Palavra (w)

- Palavra: Sequência <u>finita</u> de símbolos ou caracteres justapostos
- Também chamada de:
  - Cadeia de caracteres
  - Sentença
- Notação: w
- Exemplos de palavras no alfabeto $\{a, b\}$:
  - ab, bb, aaa
  - ε ou $\lambda \Rightarrow$ cadeia vazia
- Não são palavras no alfabeto $\{a, b\}$:
  - $ab...$
  - $aaa...$
  - abc

Palavra (w)

Elementos de uma palavra:

- Prefixo: É qualquer sequência inicial de símbolos de uma palavra
- Sufixo: É qualquer sequência final de símbolos de uma palavra
- Subpalavra: É qualquer sequência de símbolos contíguos de uma palavra
- Comprimento: $|w| \Rightarrow$ número de caracteres de uma palavra
- Ex.: $|\varepsilon| = 0$
- Ex.: Sobre o alfabeto $\{a, b, c\}$ e a palavra $w = abcb$, Temos:
  - $|w| = 4$
  - Prefixos de w: ε, a, abc, $abcb$
  - Sufixos de w: ε, b, bcb, $abcb$
- **Todos sufixo e prefixo é uma subpalavra!!!**

Palavra (w)

Exemplo de palavra em Linguagem de programação:

- Um bloco de programa
- Uma palavra-chave, por exemplo `if`, `while`...
- Um programa inteiro

Palavra - Propriedade: Concatenação $(\cdot)$

Concatenação $(\cdot)$:

- Operação binária sobre um conjunto de palavras
- Associa duas palavras , ou seja, é a justaposição da primeira palavra com a segunda
- **Propriedades**:
  - Elemento neutro (ε): $\varepsilon \cdot w = w = w \cdot \varepsilon$
  - Associativa: $v \cdot (w \cdot t) = (v \cdot w) \cdot t$
    ou $\quad v(wt) = (vw)t$
- Ex.: Para o alfabeto $\Sigma = \{a, b\}$ e as palavras $w = bb$ e $w_1 = baaab$, temos:
  - $w_1 \cdot w = baaab \cdot bb = baaabbb$
  - $w_1 \cdot \varepsilon = baaab \cdot \varepsilon = baaab$

Palavra - Concatenação Sucessiva

Exemplo de Concatenação Sucessiva:

- $b^4 = b \cdot b \cdot b \cdot b = bbbb$
- $a^0 = \varepsilon$
- $a^3b^5 = a^3 \cdot b^5 = a \cdot a \cdot a \cdot b \cdot b \cdot b \cdot b \cdot b = aaabbbbb$
- Se Σ é um alfabeto, então:
  - Σ^* é conjunto de todas as palavras possíveis sobre Σ
  - $\Sigma^+ = \Sigma^* - \varepsilon$

Linguagem Formal

Linguagem Formal (L)

Linguagem Formal é um subconjunto de Σ^*. Ou seja, um subconjunto de todas "palavras" possíveis dentro de um alfabeto

Notação: L

Exemplos:

- $\emptyset$ e $\{\varepsilon\}$ $\Rightarrow$ são linguagens sobre qualquer alfabeto
- Σ^* e Σ^+ $\Rightarrow$ são linguagens sobre qualquer alfabeto
- Conjunto de palíndromos sobre $\Sigma = \{a, b\}$:
  - $\{\varepsilon, a, b, aba, bab, aabbaa, \cdots\}$
- Linguagem de programação: Conjunto de todos programas (palavras) de uma linguagem de programação. Ex.:
  - Conjunto de palavras chave de uma linguagem de programação: `if`, `while`, `do` , `int`, `integer`...

Linguagem Formal

Exemplos de palavras-chave de linguagem de programação:

- $C++$: {`for`, `if`, `while`, `do`, `int`, `new` ... }
- *Delphi*: {`for`, `if`, `while`, `do`, `integer`, `begin`, `end` ... }
- *Java*: {`for`, `if`, `while`, `do`, `int`, `forech` ... }

Veja que:

- Cada linguagem tem um conjunto de palavras que aceita
- Ou seja, esse conjunto de palavras aceitas é o que chamamos de linguagem

Pergunta...

- A palavra "**`begin`**" é aceita pela linguagem $C++$?
- A palavra "**`how`**" é aceita pela linguagem português?
- A palavra "`teste := ` $2 * x$`;`" é aceita pela linguagem Java?

Gramática

Gramática (G)

- Gramática (G): Conjunto **finito de regras**
- Palavras são geradas quando essas regras são aplicada sucessivas vezes!
- Definição de Linguagem: é o conjunto de todas as palavras geradas por uma gramática
- **Obs.:** A gramática também é usada para definir semântica (ou aeja, na análise semântica)

Definição Formal de **Gramática**

Gramatica de Chomsky, ou gramática irrestrita, ou apenas gramática:

$$G = (V, T, P, S)$$

onde:

- V $\rightarrow$ variáveis ou não-terminais (SEMPRE letras MAIÚSCULAS)
- T $\rightarrow$ terminais (letras minúsculas, dígitos ou caracteres)
- $P\rightarrow$ regras de produções ou produções
  - **Par de relação, regra de produção ou produção:**

    $(V \cup T)^{+} \rightarrow (V \cup T)^{+} \Rightarrow$ relação finita
- S $\rightarrow$ variável inicial (elemento diferente de V)

Representação de uma regra de produção:

$$\alpha \to \beta$$

- Se α tem mais de uma produção:

$$\begin{array}{lcc} \alpha \to \beta_1 & & \\ \alpha \to \beta_2 & \Rightarrow & \alpha \to \beta_1 \mid \beta_2 \mid \beta_3 \\ \alpha \to \beta_3 & & \text{(forma abreviada)} \end{array}$$

Gramática - Derivação

Derivação:

- É o processo de aplicar as regras de produções sucessivas vezes, começando pela variável inicial (S)
- Permite gerar as palavras da linguagem

Na prática derivar é:

- Substituir uma <u>subpalavra</u>, segunda uma regra de produção
- **Ex.:** Para as regra de produções abaixo:

  $$\alpha \to \beta_1 \mid \beta_2 \mid \beta_3$$

  Podemos substituir α por:

  - β_1
  - β_2
  - β_3

Gramática - Exemplo 1

Seja a gramática:

$$G = (V, T, P, S)$$

- $V = \{N, D\}$
- $T = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- $P = \{$

  $N \rightarrow D$

  $N \rightarrow DN$

  $D \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

  $\}$
- $S = \{N\}$

Gramática - Exemplo 1

Seja a gramática:

$$G = (V, T, P, S)$$

- $V = \{N, D\}$
- $T = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- $P = \{$

  1 $N \rightarrow D$

  2 $N \rightarrow DN$

  3 $D \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

  $\}$
- $S = \{N\}$

Para facilitar vamos numerar as produções

Vamos derivar o número 243

Gramática - Exemplo 1

Derivação do número 243:

P = {
  N → D 1
  N → DN 2
  D → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 3
}

1º $N \to DN$ aplicando a regra 2 em N

Gramática - Exemplo 1

Derivação do número 243:

P = {
  N → D (1)
  N → DN (2)
  D → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 (3)
}

1º	$N \rightarrow DN$	aplicando a regra (2) em N
2º	$DN \rightarrow 2N$	aplicando a regra (3) em D

Derivação do número 243:

P = {
  N → D ①
  N → DN ②
  D → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 ③
}

1º	$N \rightarrow DN$	aplicando a regra ② em N
2º	$DN \rightarrow 2N$	aplicando a regra ③ em D
3º	$2N \rightarrow 2DN$	aplicando a regra ② em N

Derivação do número 243:

P = {
  N → D (1)
  N → DN (2)
  D → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 (3)
}

1º	$N \rightarrow DN$	aplicando a regra (2) em N
2º	$DN \rightarrow 2N$	aplicando a regra (3) em D
3º	$2N \rightarrow 2DN$	aplicando a regra (2) em N
4º	$2DN \rightarrow 24N$	aplicando a regra (3) em D

Gramática - Exemplo 1

Derivação do número 243:

```
P = {
    N → D                         1
    N → DN                        2
    D → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9    3
}
```

1º	$N \rightarrow DN$	aplicando a regra 2 em N
2º	$DN \rightarrow 2N$	aplicando a regra 3 em D
3º	$2N \rightarrow 2DN$	aplicando a regra 2 em N
4º	$2DN \rightarrow 24N$	aplicando a regra 3 em D
5º	$24N \rightarrow 24D$	aplicando a regra 1 em N

Gramática - Exemplo 1

Derivação do número 243:

P = {
  N → D ①
  N → DN ②
  D → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 ③
}

1º	$N \rightarrow DN$	aplicando a regra ② em N
2º	$DN \rightarrow 2N$	aplicando a regra ③ em D
3º	$2N \rightarrow 2DN$	aplicando a regra ② em N
4º	$2DN \rightarrow 24N$	aplicando a regra ③ em D
5º	$24N \rightarrow 24D$	aplicando a regra ① em N
6º	$24D \rightarrow 243$	aplicando a regra ③ em D

Gramática - Exemplo 1

Derivação do número 243:

P = {
N → D ①
N → DN ②
D → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 ③
}

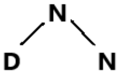
N

Aplicando a regra ②

Gramática - Exemplo 1

Derivação do número 243:

P = {
  N → D ①
  N → DN ②
  D → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 ③
}

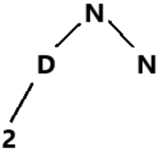


Aplicando a regra ③ em **D**

Gramática - Exemplo 1

Derivação do número 243:

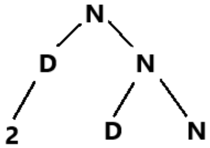
P = {
N → D (1)
N → DN (2)
D → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 (3)
}



Aplicando a regra ② em **N**

Derivação do número 243:

P = {
N → D 1
N → DN 2
D → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 3
}

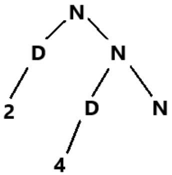


Aplicando a regra 3 em **D**

Gramática - Exemplo 1

Derivação do número 243:

P = {
N → D (1)
N → DN (2)
D → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 (3)
}

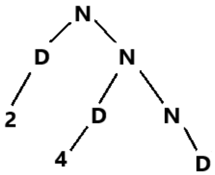


Aplicando a regra ① em **N**

Gramática - Exemplo 1

Derivação do número 243:

$$
\begin{aligned}
&P = \{ \\
&\quad N \rightarrow D \quad 1 \\
&\quad N \rightarrow DN \quad 2 \\
&\quad D \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \quad 3 \\
&\}
\end{aligned}
$$

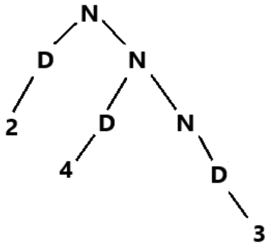


Aplicando a regra 3 em **D**

Gramática - Exemplo 1

Derivação do número 243:

P = {
N → D (1)
N → DN (2)
D → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 (3)
}



Gramática - Exemplo 2

$$G = (V, T, P, S)$$

- V = {N, D}
- T = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
- P = {

  $N \rightarrow D$

  $N \rightarrow DN$

  $D \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

  }
- S = {N}

Faça as seguintes derivações

- 1000
- 02
- 5890
- 3,45

Gramática - Exemplo 2

Podemos concluir que:

- A gramática anterior produz a linguagem dos números naturais
- Denotamos LINGUAGEM GERADA por:

$$G\text{: } L(G) \text{ ou } \mathrm{GERA}(G)$$

- Para que duas linguagens G_1 e G_2 sejam iguais ou equivalentes: o conjunto das palavras aceitas devem ser iguais, ou seja:

$$L(G_1) = L(G_2)$$

- Convencionaremos que:
  - Letras MAIÚSCULS para Variáveis $\Rightarrow A, B, C, S, ..., T$
  - Letras minúsculas para terminais $\Rightarrow a, b, c, s, ..., t$
  - Para palavras de símbolos terminais $\Rightarrow u, v, w, x, y, z$

Gramática - Exercício 1

- G = (V, T, P, S)
- V = {S, X, Y, A, B, F}
- T = {a,b}
- P = {

  S → XY

  X → XaA | XbB | F

  Aa → aA

  Ab → bA

  AY → Ya

  Ba → aB

  Bb → bB

  BY → Yb

  Fa → aF

  Fb → bF

  FY → ε

  }
- S

- Esta gramática gera a linguagem:

  {ww | w é palavra de {a,b}* }

- Isto é:
  - ab<u>ab</u>
  - abb<u>abb</u>
  - aaa<u>aaa</u>
- Exercicio, derive:
- ba<u>ba</u>
- bbba<u>bbba</u>

Gramática - Exercício 2

- G = (V, T, P, S)
- V = {A, B, C}
- T = {a,b}
- P = {

  S → aA | bB

  A → bB | aC

  B → aA | bC

  C → a | b | aC | bC | ε

  }
- S

- Esta gramática gera a linguagem:

  {w | w tem pelo menos ***aa*** ou ***bb*** como subpalavra}

- Testar se as palavras são aceitas
  - abab
  - aab
  - abaab
  - abb
  - babbabab

Gramática - Exercício 3

$$G = (V, T, P, S)$$

$V = \{S, A, B\}$
$T = \{a, b\}$
$P = \{$

- $S \rightarrow AB$
- $A \rightarrow aA \mid \varepsilon$
- $B \rightarrow bB \mid \varepsilon$

$\}$
$S = \{S\}$

Esta gramática gera a linguagem:

$$\{a^n b^n \mid n \geq 0\}$$

Ou seja:

- ab
- $aabb$
- $aaabbb$

Derive as seguintes sentenças:

a) $aabb$
b) $aaabbb$
c) ab

Gramática - Exercício 4

$$G = (V, T, P, S)$$

$V = \{S, A\}$
$T = \{0, 1\}$
$P = \{$

- $S \rightarrow 0S1 \mid A$
- $A \rightarrow \varepsilon$

$\}$
$S = \{S\}$

Esta gramática gera a linguagem:

$$\{0^n 1^n \mid n \geq 0\}$$

Ou seja:

- 01
- 0011
- 000111

Derive as seguintes sentenças:

a. 0011
b. 000111
c. 01

$$G = (V, T, P, S)$$

$V = \{S, A, B\}$
$T = \{a, b\}$
$P = \{$

- $S \rightarrow aA \mid bB$
- $A \rightarrow aS \mid bAA \mid \varepsilon$
- $B \rightarrow bS \mid aBB \mid \varepsilon$

$\}$
$S = \{S\}$

Esta gramática gera a linguagem de palavras com quantidades de a e b balanceadas. Ou seja:

- aba
- $bbaabb$
- $aabb$

Derive as seguintes sentenças:

a. $abab$
b. $aabbaa$
c. $bbabb$

$$G = (V, T, P, S)$$

$V = \{S, X, Y, A, B, F\}$
$T = \{a, b\}$
$P = \{$

- $S \to XY$
- $X \to XaA \mid XbB \mid F$
- $Aa \to aA$
- $Ab \to bA$
- $AY \to Ya$
- $Ba \to aB$
- $Bb \to bB$
- $BY \to Yb$
- $Fa \to aF$
- $Fb \to bF$
- $FY \to \varepsilon$

$\}$
$S = \{S\}$
Esta gramática gera a linguagem:

$$\{ww \mid w \text{ é palavra de } \{a, b\}^*\}$$

Ex.: $abab$, $abbabb$, $aaaaaa$, etc.
Derive as seguintes sentenças:

a. $baba$
b. $bbbabbba$