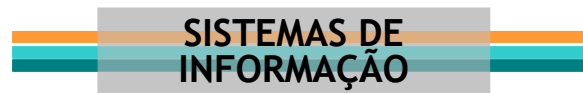


Tema 06

Métodos sintáticos

Professora:
Ariane Machado Lima



Avisos

Próxima aula: PROVA

Atividade de validação cruzada para 10/10

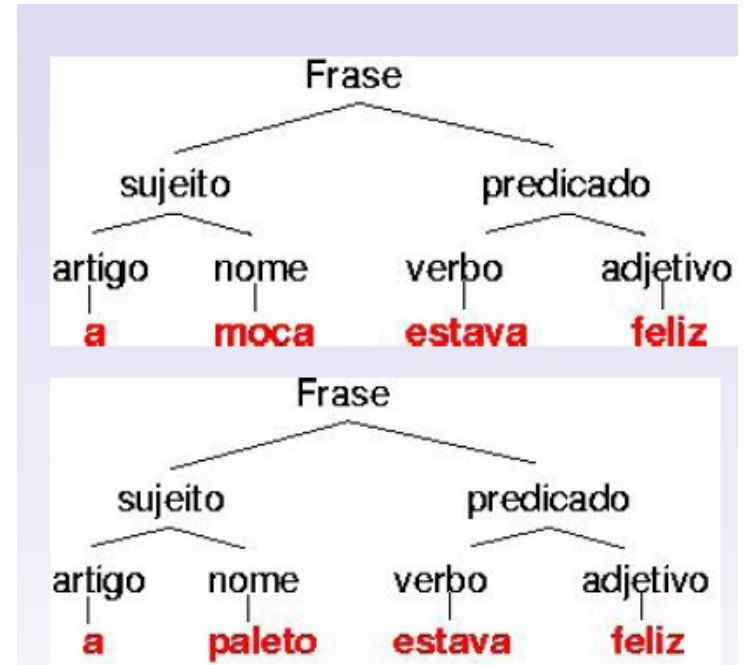
Trabalho final: para 06/12

- Trabalho escrito ou
- APRESENTAÇÕES DIA 06/12



Gramáticas

Frase	→	sujeito	predicado
sujeito	→	artigo	nome
artigo	→	a	
artigo	→	o	
nome	→	paletó	
nome	→	moça	
nome	→	dia	
predicado	→	verbo	adjetivo
verbo	→	é	
verbo	→	estava	
adjectivo	→	feliz	
adjectivo	→	azul	

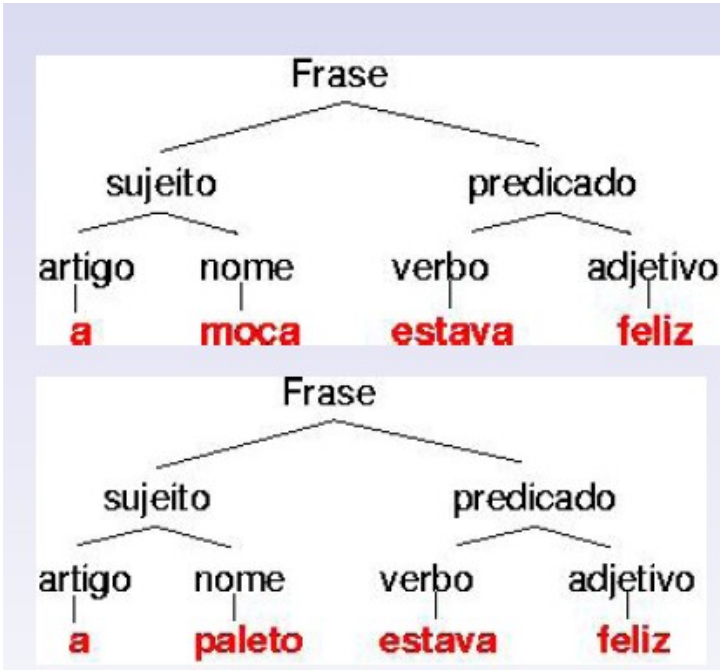


conjunto de produções

Gramáticas

símbolo inicial

Frase	→	sujeito	predicado
sujeito	→	artigo	nome
artigo	→	a	
artigo	→	o	
nome	→	paletó	
nome	→	moça	
nome	→	dia	
predicado	→	verbo	adjetivo
verbo	→	é	
verbo	→	estava	
adjetivo	→	feliz	
adjetivo	→	azul	



símbolos não-terminais

símbolos terminais



Gramáticas

- Definição: uma gramática G é uma quádrupla (V, Σ, S, P) , onde
 - V é o conjunto de símbolos não-terminais (variáveis)
 - Σ é o conjunto de símbolos terminais
 - S é o símbolo inicial
 - P é o conjunto de produções da forma
 $(\Sigma \cup V)^* V (\Sigma \cup V)^* \rightarrow (\Sigma \cup V)^*$



Gramáticas

- Uma **forma sentencial** de uma gramática G é qualquer cadeia obtida pela aplicação recorrente das seguintes regras:
 - S (símbolo inicial de G) é uma forma sentencial
 - Sejam $\alpha\beta$ uma forma sentencial de G e $\rho \rightarrow \gamma$ uma produção de G . Então $\alpha\gamma\beta$ é também uma forma sentencial de G .

$(\alpha, \beta, \gamma \in (\Sigma \cup V)^*$ e $\rho \in (\Sigma \cup V)^* V (\Sigma \cup V)^*$)

Gramáticas

- Uma **forma sentencial** de uma gramática G é qualquer cadeia obtida pela aplicação recorrente das seguintes regras:
 - S (símbolo inicial de G) é uma forma sentencial
 - Sejam $\alpha\rho\beta$ uma forma sentencial de G e $\rho \rightarrow \gamma$ uma produção de G . Então $\alpha\gamma\beta$ é também uma forma sentencial de G .

$(\alpha, \beta, \gamma \in (\Sigma \cup V)^*$ e $\rho \in (\Sigma \cup V)^* V (\Sigma \cup V)^*$)

- **Derivação direta:**
 - $\alpha\rho\beta \Rightarrow \alpha\gamma\beta$



Gramáticas

- **Derivação**: aplicação de zero ou mais derivações diretas
 - $\alpha \Rightarrow^* \mu$
 - isto é, $\alpha \Rightarrow \beta \Rightarrow \dots \Rightarrow \mu$
- Uma cadeia w ($w \in \Sigma^*$) é uma **sentença** de G se $S \Rightarrow^* w$ (S sendo o símbolo inicial)
- Linguagem **gerada** por G :

$$L(G) = \{ w \in \Sigma^* \mid S \Rightarrow^* w \}$$



Gramáticas - Exemplos

- $G = (V, \Sigma, S, P)$, onde

- $V = \{S, A\}$

- $\Sigma = \{0,1,2,3\}$

- $S = S$

- $P = \{$

$S \rightarrow 0S33$

$S \rightarrow A$

$A \rightarrow 12$

$A \rightarrow \varepsilon$

$\}$



Gramáticas - Exemplos

- $G = (V, \Sigma, S, P)$, onde
 - $V = \{S, A\}$
 - $\Sigma = \{0,1,2,3\}$
 - $S = S$
 - $P = \{$
 - $S \rightarrow 0S33$
 - $S \rightarrow A$
 - $A \rightarrow 12$
 - $A \rightarrow \varepsilon$
- Ex de formas sentenciais:
 $S, 0S33, 00S3333, 00A3333$
- $0S33 \Rightarrow 00S3333$
- $0S33 \Rightarrow^* 00A3333$
- $0S33 \Rightarrow^* 0S33$
- Ex de sentenças:
 $00123333, 12, \varepsilon$
-



Gramáticas - Exemplos

- $G = (V, \Sigma, S, P)$, onde
 - $V = \{S, A\}$
 - $\Sigma = \{0, 1, 2, 3\}$
 - $S = S$
 - $P = \{$
 - $S \rightarrow 0S33$
 - $S \rightarrow A$
 - $A \rightarrow 12$
 - $A \rightarrow \varepsilon$
- Ex de formas sentenciais:
 $S, 0S33, 00S3333, 00A3333$
- $0S33 \Rightarrow 00S3333$
- $0S33 \Rightarrow^* 00A3333$
- $0S33 \Rightarrow^* 0S33$
- Ex de sentenças:
 $00123333, 12, \varepsilon$
- $L(G) =$



Gramáticas - Exemplos

- $G = (V, \Sigma, S, P)$, onde
 - $V = \{S, A\}$
 - $\Sigma = \{0, 1, 2, 3\}$
 - $S = S$
 - $P = \{$
 - $S \rightarrow 0S33$
 - $S \rightarrow A$
 - $A \rightarrow 12$
 - $A \rightarrow \varepsilon$ $\}$
- Ex de formas sentenciais:
 $S, 0S33, 00S3333, 00A3333$
 - $0S33 \Rightarrow 00S3333$
 - $0S33 \Rightarrow^* 00A3333$
 - $0S33 \Rightarrow^* 0S33$
- Ex de sentenças:
 $00123333, 12, \varepsilon$
 - $L(G) = \{0^m 1^n 2^n 3^{2m} \mid m \geq 0 \text{ e } n = 0 \text{ ou } n = 1\}$



Gramáticas - Simplificação

• $G = (V, \Sigma, S, P)$, onde

• $V = \{S, A\}$

• $\Sigma = \{0,1,2,3\}$

• $S = S$

• $P = \{$

$S \rightarrow 0S33$

$S \rightarrow A$

$A \rightarrow 12$

$A \rightarrow \varepsilon$

$\}$



• $G = (V, \Sigma, S, P)$,
onde

• $V = \{S, A\}$

• $\Sigma = \{0,1,2,3\}$

• $S = S$

• $P = \{$

$S \rightarrow 0S33 \mid A$

$A \rightarrow 12 \mid \varepsilon$

$\}$



Gramáticas

- Gramáticas são dispositivos **generativos** (geram cadeias)
- Dada uma cadeia w , reconhecer se $w \in L(G)$ é um processo chamado **análise sintática**
- Dependendo do formato das produção, a análise sintática pode ser mais ou menos complexa

Hierarquia de Chomsky

- Hierarquia das linguagens em classes de acordo com a sua complexidade relativa (Noam Chomsky, 1956)
- Cada classe de linguagem pode ser gerada por um tipo de gramática (formato das produções)
- Cada tipo de gramática tem uma complexidade de análise sintática diferente
- Na prática: dada uma linguagem, saber qual o dispositivo mais eficiente para análise sintática



Hierarquia de Chomsky

$$\alpha \rightarrow \beta$$

Linguagens irrestritas
(tipo 0)

$$\alpha \in (VU\Sigma)^*V(VU\Sigma)^* \\ \beta \in (VU\Sigma)^*$$

Linguagens sensíveis ao contexto
(tipo 1)

$$\alpha \in (VU\Sigma)^*V(VU\Sigma)^* \\ \beta \in (VU\Sigma)^* \\ |\alpha| \leq |\beta|$$

Linguagens livres de contexto
(tipo 2)

$$\alpha \in V \\ \beta \in (VU\Sigma)^*$$

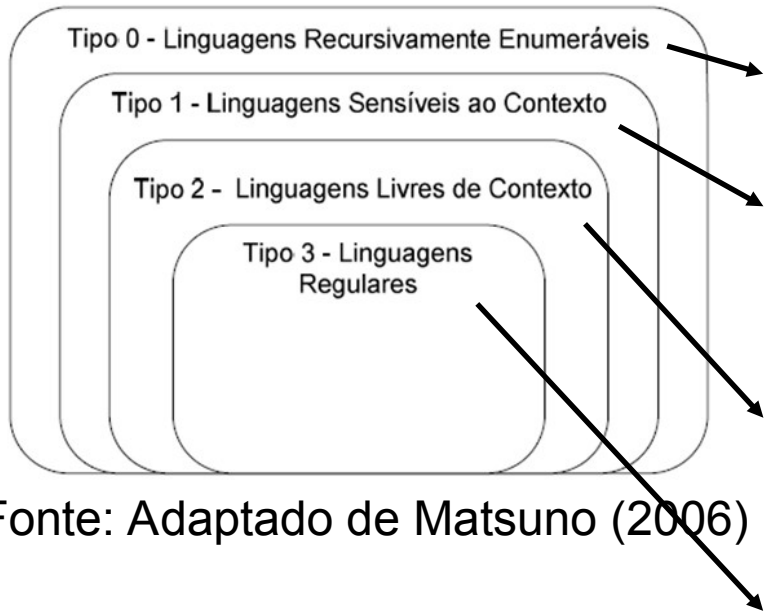
Linguagens regulares
(tipo 3)

$$\alpha \in V$$

$$\beta \in \Sigma_\epsilon, \beta \in V, \beta \in (V\Sigma \text{ ou } \Sigma V)$$



Hierarquia de Chomsky



Fonte: Adaptado de Matsuno (2006)

Linguagem	Autômato	Gramática	Reconhecimento
Recursivamente enumerável	Máquina de Turing com fita infinita 	Irrestrita $Baa \rightarrow A$	Indecidível
Sensível ao contexto	Máquina de Turing com fita finita 	Sensível ao contexto $At \rightarrow aA$	NP-Completo
Livre de contexto	Autômato de pilha 	Livre de contexto $S \rightarrow gSc$	Polinomial
Regular	Autômato finito 	Regular $A \rightarrow cA$	Linear

Fonte: Adaptado de Searls (2002)

Árvore sintática ou árvore de derivação

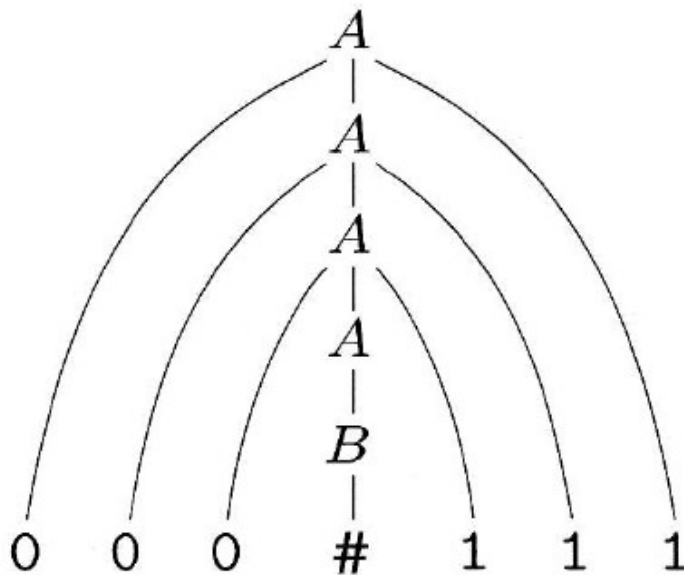
$$A \rightarrow 0A1$$

$$A \rightarrow B$$

$$B \rightarrow \#$$

Por exemplo, a gramática G_1 gera a cadeia 000#111.

$$A \Rightarrow 0A1 \Rightarrow 00A11 \Rightarrow 000A111 \Rightarrow 000B111 \Rightarrow 000\#111$$

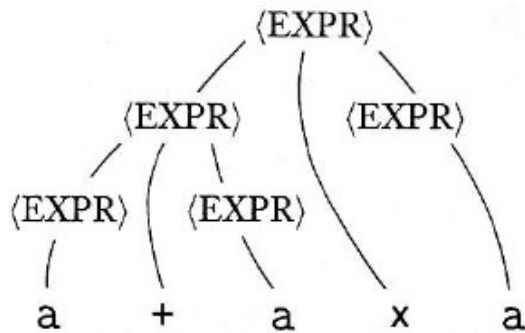


Árvore sintática
ou
Árvore de derivação



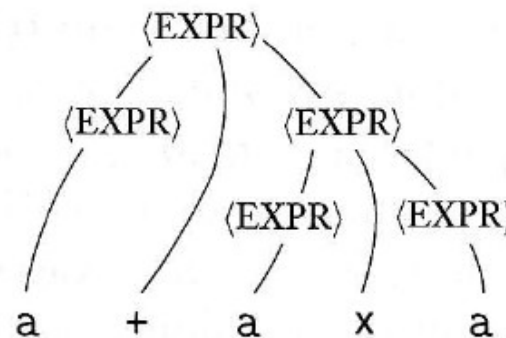
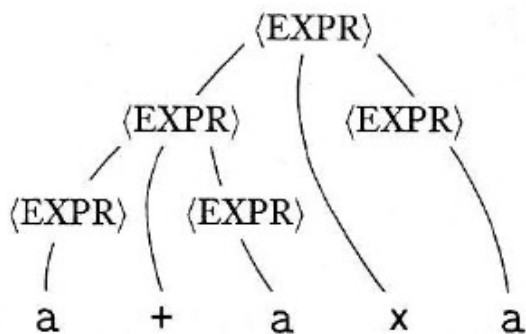
Mais exemplos de árvores sintáticas

$\langle \text{EXPR} \rangle \rightarrow \langle \text{EXPR} \rangle + \langle \text{EXPR} \rangle \mid \langle \text{EXPR} \rangle \times \langle \text{EXPR} \rangle \mid (\langle \text{EXPR} \rangle) \mid a$



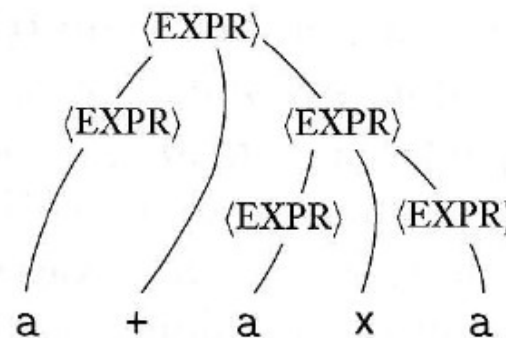
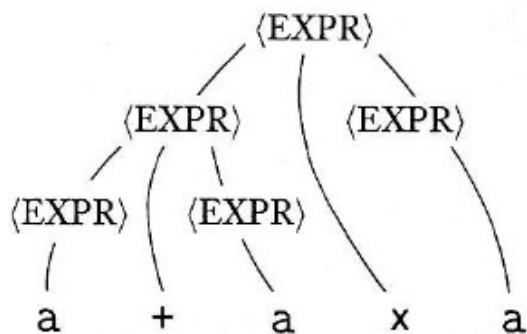
Mais exemplos de árvores sintáticas

$\langle \text{EXPR} \rangle \rightarrow \langle \text{EXPR} \rangle + \langle \text{EXPR} \rangle \mid \langle \text{EXPR} \rangle \times \langle \text{EXPR} \rangle \mid (\langle \text{EXPR} \rangle) \mid a$



Mais exemplos de árvores sintáticas

$\langle \text{EXPR} \rangle \rightarrow \langle \text{EXPR} \rangle + \langle \text{EXPR} \rangle \mid \langle \text{EXPR} \rangle \times \langle \text{EXPR} \rangle \mid (\langle \text{EXPR} \rangle) \mid a$



Duas árvores sintáticas distintas para a mesma cadeia!!!!

Ambiguidade

DEFINIÇÃO 2.7

Uma cadeia w é derivada *ambiguamente* na gramática livre-do-contexto G se ela tem duas ou mais derivações mais à esquerda diferentes. A gramática G é *ambígua* se ela gera alguma cadeia ambiguamente.

Ambiguidade

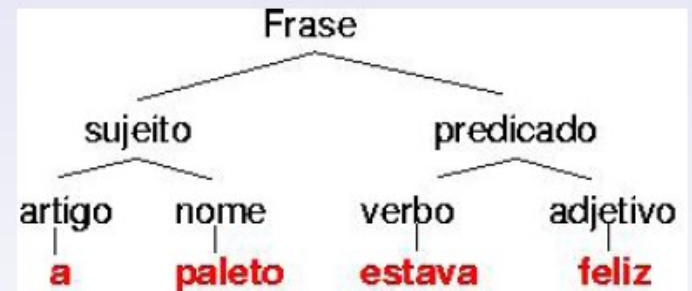
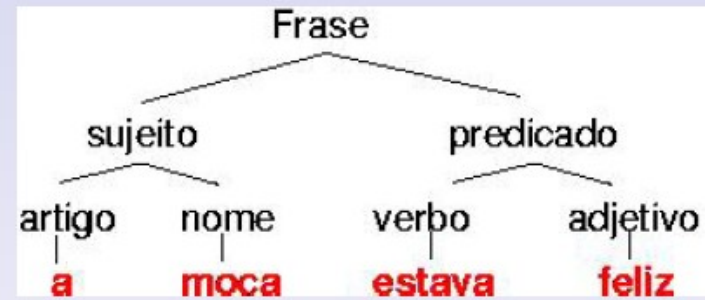
DEFINIÇÃO 2.7

Uma cadeia w é derivada *ambiguamente* na gramática livre-do-contexto G se ela tem duas ou mais derivações mais à esquerda diferentes. A gramática G é *ambígua* se ela gera alguma cadeia ambiguamente.

- Algumas gramáticas ambíguas podem ser convertidas em não-ambíguas
- Algumas linguagens são inerentemente ambíguas (só podem ser descritas por gramáticas ambíguas)
 - Eu vi o menino com uma luneta

Gramáticas Estocásticas

Frase	→	sujeito	predicado	[1.0]
sujeito	→	artigo	nome	[1.0]
artigo	→	a		[0.3]
artigo	→	o		[0.7]
nome	→	paletó		[0.2]
nome	→	moça		[0.3]
nome	→	dia		[0.5]
predicado	→	verbo	adjetivo	[1.0]
verbo	→	é		[0.7]
verbo	→	estava		[0.3]
adjectivo	→	feliz		[0.6]
adjectivo	→	azul		[0.4]

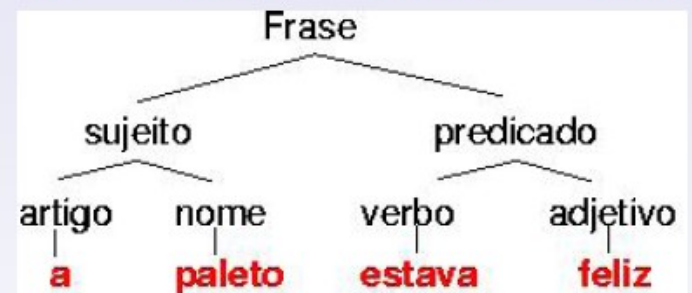
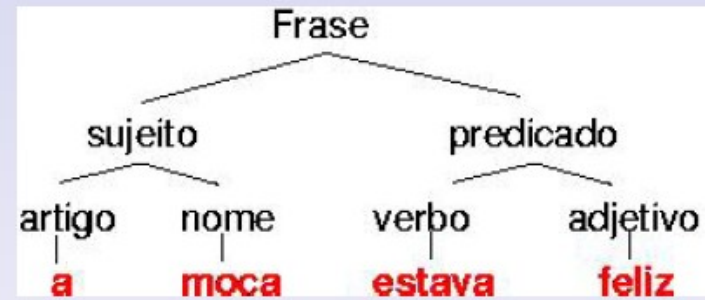


$P(x, t | G)$ = produto das probabilidades das produções usadas na derivação t de x , para uma dada gramática G

$$P(x | G) = \sum_i P(x, t_i | G)$$

Gramáticas Estocásticas

Frase	→	sujeito	predicado	[1.0]
sujeito	→	artigo	nome	[1.0]
artigo	→	a		[0.3]
artigo	→	o		[0.7]
nome	→	paletó		[0.2]
nome	→	moça		[0.3]
nome	→	dia		[0.5]
predicado	→	verbo	adjetivo	[1.0]
verbo	→	é		[0.7]
verbo	→	estava		[0.3]
adjectivo	→	feliz		[0.6]
adjectivo	→	azul		[0.4]



$P(x, t | G)$ = produto das probabilidades das produções usadas na derivação t de x , para uma dada gramática G

$P(x | G) = \sum_i P(x, t_i | G)$ ← **Considera a ambiguidade da linguagem!**

Gramáticas Estocásticas

- Definição: uma gramática estocástica G é uma quintupla (V, Σ, S, P, ρ) , onde
 - V é o conjunto de símbolos não-terminais (variáveis)
 - Σ é o conjunto de símbolos terminais
 - S é o símbolo inicial
 - P é o conjunto de produções da forma
 $(\Sigma \cup V)^* V (\Sigma \cup V)^* \rightarrow (\Sigma \cup V)^*$
 - ρ é o conjunto de distribuições de probabilidades sobre as produções de mesmo lado esquerdo

$$\sum_i \rho(\alpha \rightarrow \beta_i) = 1$$



Gramática

- Reconhecimento
- Geração
- Árvore sintática



Análise sintática

- Analisadores sintáticos: programas que, dados uma gramática G e uma sequência s , solta:
 - Se a sequência s é ou não reconhecida pela gramática G (não estocástico)
 - Qual a probabilidade da sequência s dada G (estocástico)
 - Qual(is) a(s) árvore(s) sintática(s) de s dada G (uma, só a mais provável ou todas)

Gramáticas para Reconhecimento de Padrões **Sintáticos e Estruturais**

- Uma classe pode ser vista como uma linguagem (conjunto de cadeias)
- Cada classe C_i é representada por uma gramática G_i
- Cada gramática atribui a uma cadeia x uma probabilidade $P(x|G_i)$: verossimilhança
- No caso binário (pertence ou não pertence à linguagem) compara-se a um modelo nulo

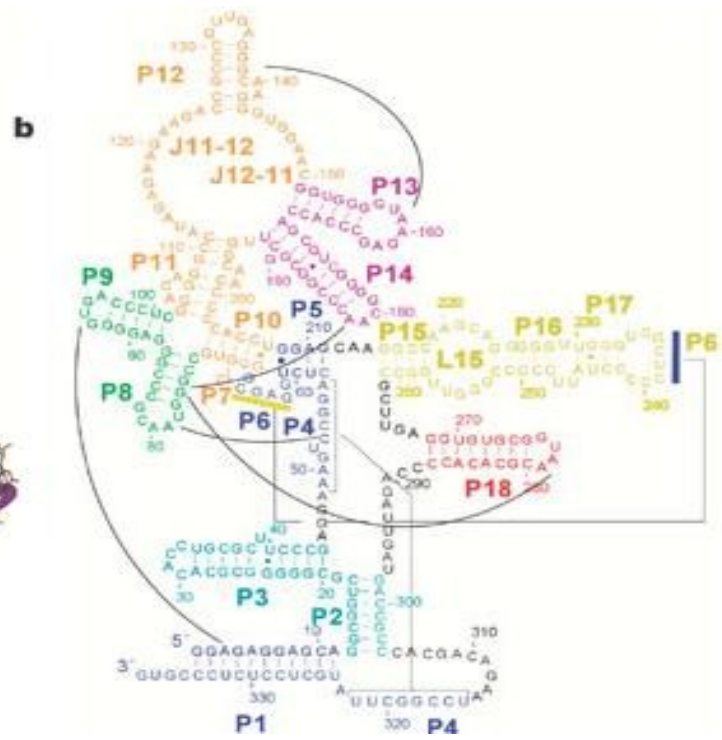
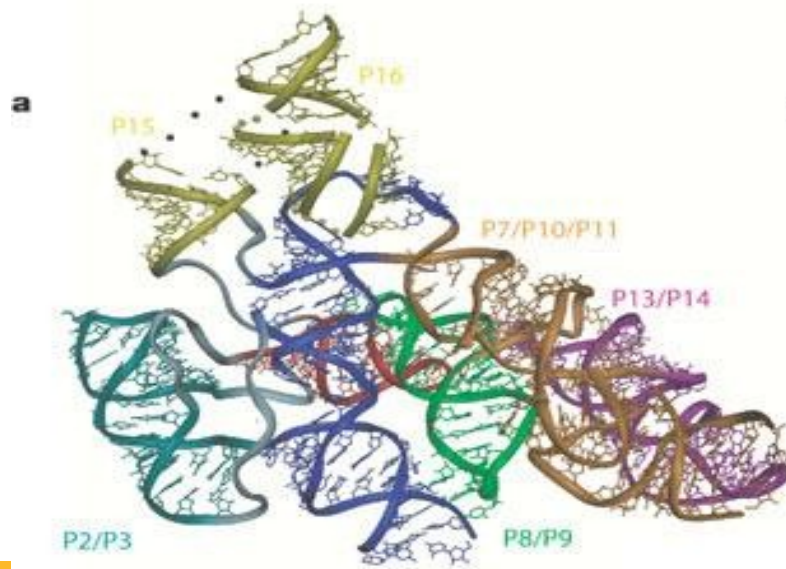
Várias aplicações

- Bioinformática (análise de sequências)
- Processamento de linguagem natural
- Imagens
- Outras...



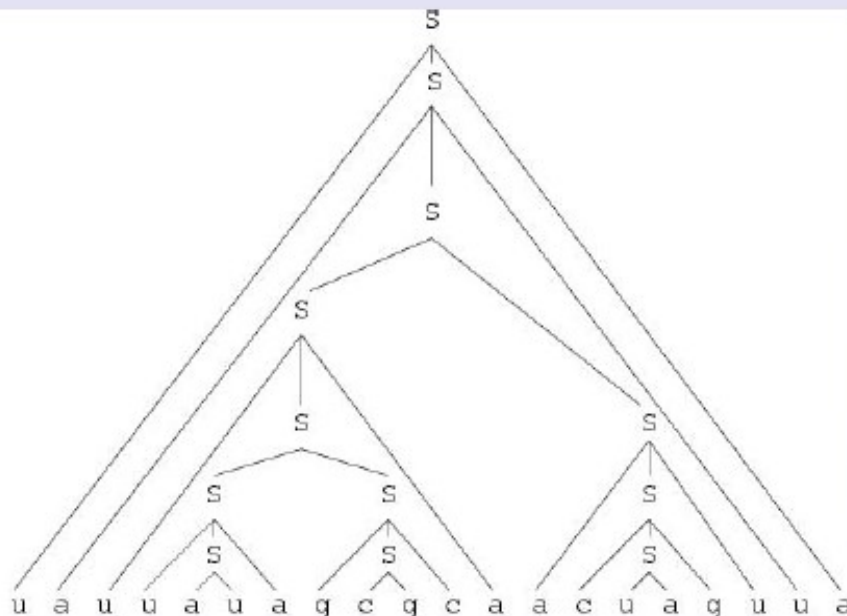
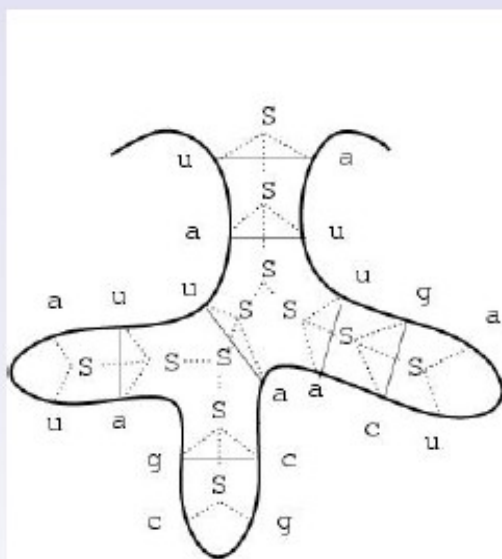
Estrutura secundária de RNAs

- Busca por padrões de sequência E estruturais
- GRAMÁTICAS REGULARES NÃO RESOLVEM...
Não conseguem modelar pareamentos entre bases

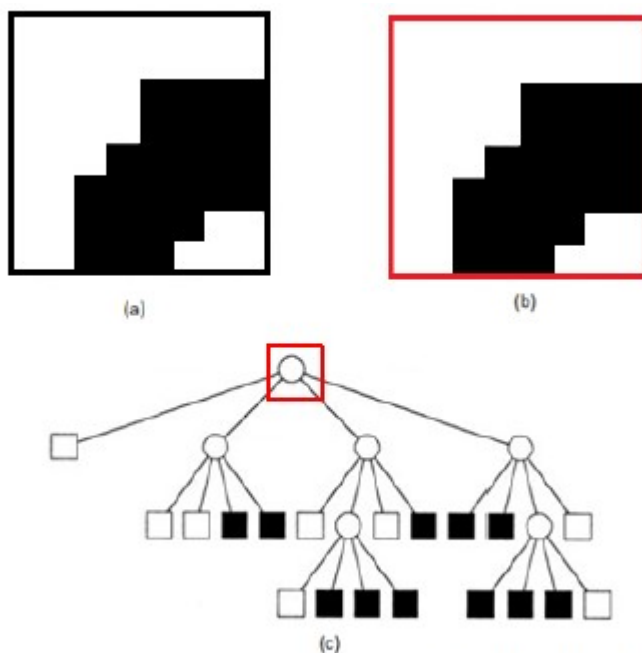


Estrutura secundária e gramáticas

$S \rightarrow$	aSu [0.1]	uSa [0.1]	cSg [0.1]	gSc [0.1]	
$S \rightarrow$	aS [0.1]	uS [0.1]	cS [0.1]	gS [0.1]	
$S \rightarrow$	SS [0.04]	a [0.04]	u [0.04]	c [0.04]	g [0.04]

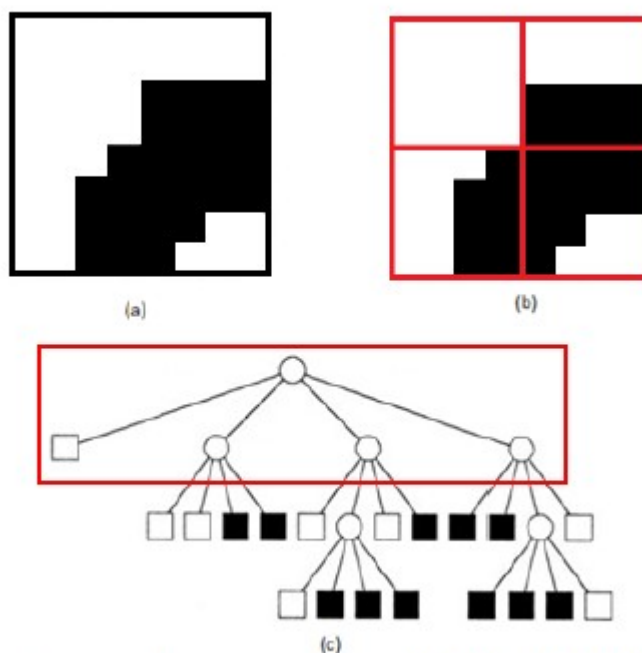


Imagens: representação por quadrees



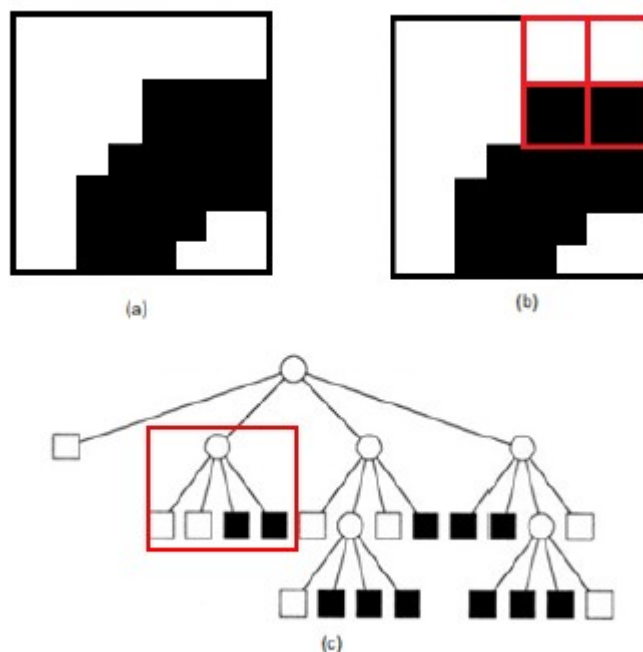
Estrutura de uma quadtree. Baseado em (AIZAWA; NAKAMURA, 1999)

Imagens: representação por quadrees



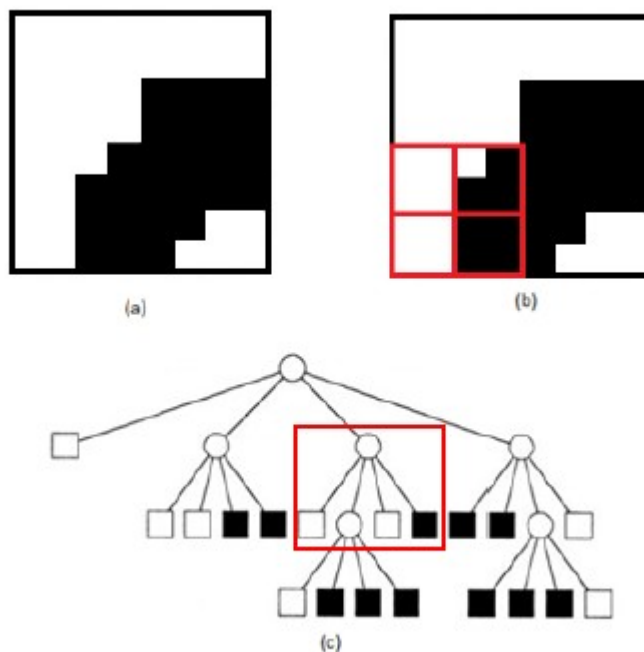
Estrutura de uma quadtree. Baseado em (AIZAWA; NAKAMURA, 1999)

Imagens: representação por quadrees



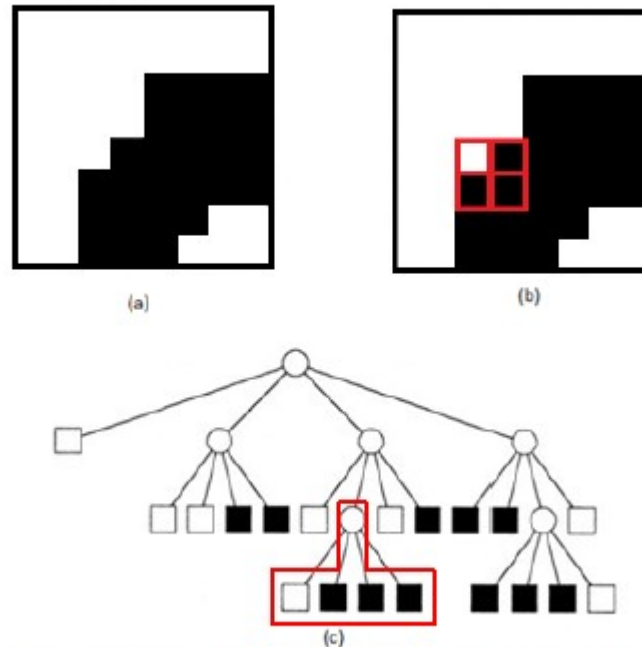
Estrutura de uma quadtree. Baseado em (AIZAWA; NAKAMURA, 1999)

Imagens: representação por quadrees



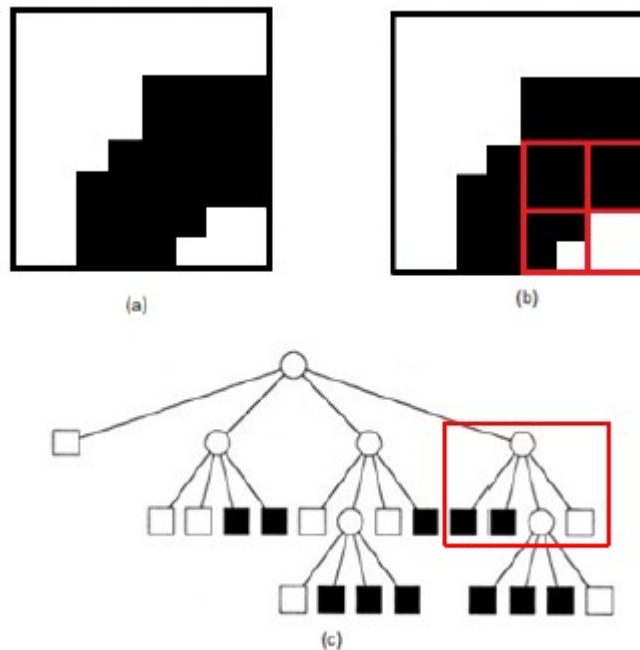
Estrutura de uma quadtree. Baseado em (AIZAWA; NAKAMURA, 1999)

Imagens: representação por quadrees



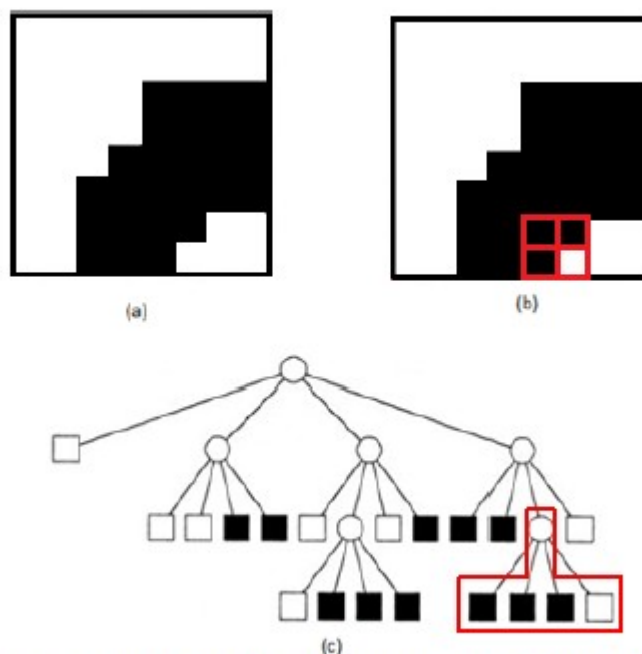
Estrutura de uma quadtree. Baseado em (AIZAWA; NAKAMURA, 1999)

Imagens: representação por quadrees



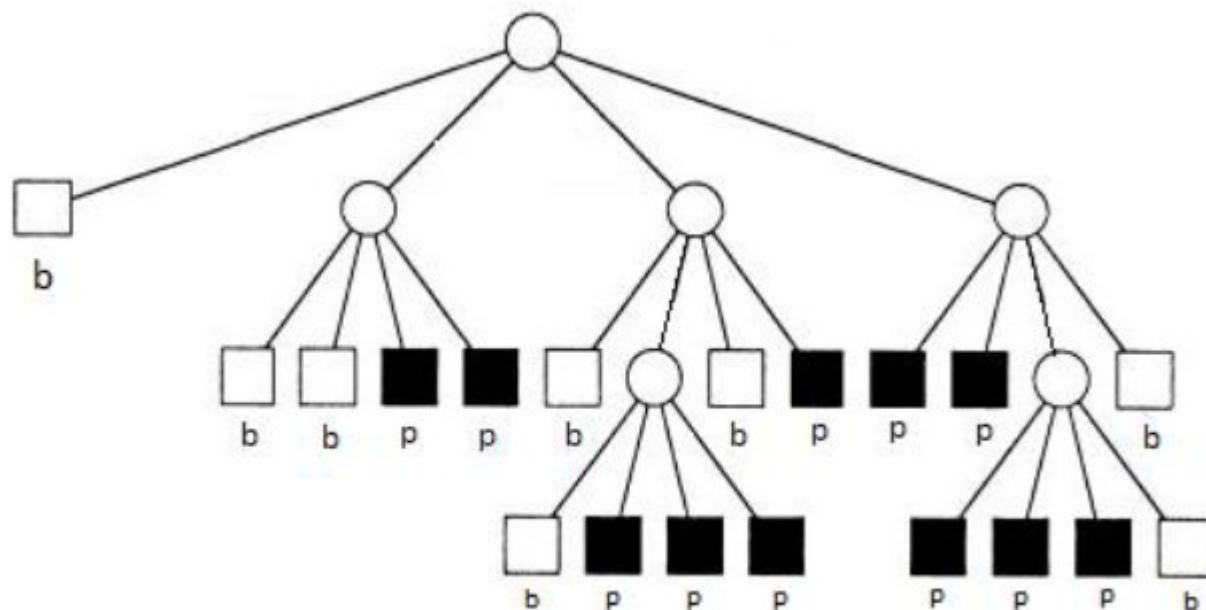
Estrutura de uma quadtree. Baseado em (AIZAWA; NAKAMURA, 1999)

Imagens: representação por quadrees



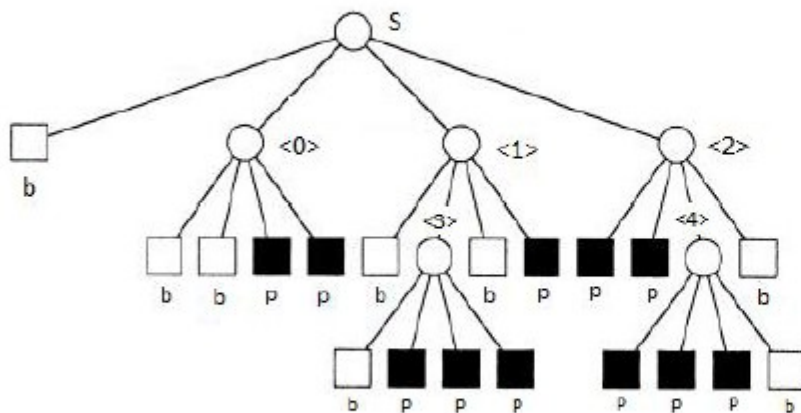
Estrutura de uma quadtree. Baseado em (AIZAWA; NAKAMURA, 1999)

Imagens: representação por quadrees



- String gerada: $(b(bbpp)(b(bppp)bp)(pp(pppb)b));$

Imagens: representação por quadrees



- Gramática gerada $G = (V_N, V_T, R, S)$
 - S representa o símbolo inicial da gramática;
 - V_N é o conjunto de símbolos não-terminais $\{S, \langle 0 \rangle, \langle 1 \rangle, \dots, \langle 4 \rangle\}$;
 - V_T é o conjunto de símbolos terminais $\{b, p, (\cdot)\}$;
 - R é o conjunto de regras de produção:
 - $S \rightarrow (b \langle 0 \rangle \langle 1 \rangle \langle 2 \rangle) [1.0]$
 - $\langle 0 \rangle \rightarrow (b b p p) [1.0]$
 - $\langle 1 \rangle \rightarrow (b \langle 3 \rangle b p) [1.0]$
 - $\langle 2 \rangle \rightarrow (p p \langle 4 \rangle b) [1.0]$
 - $\langle 3 \rangle \rightarrow (b p p p) [1.0]$
 - $\langle 4 \rangle \rightarrow (p p p b) [1.0]$

Grafos AND-OR

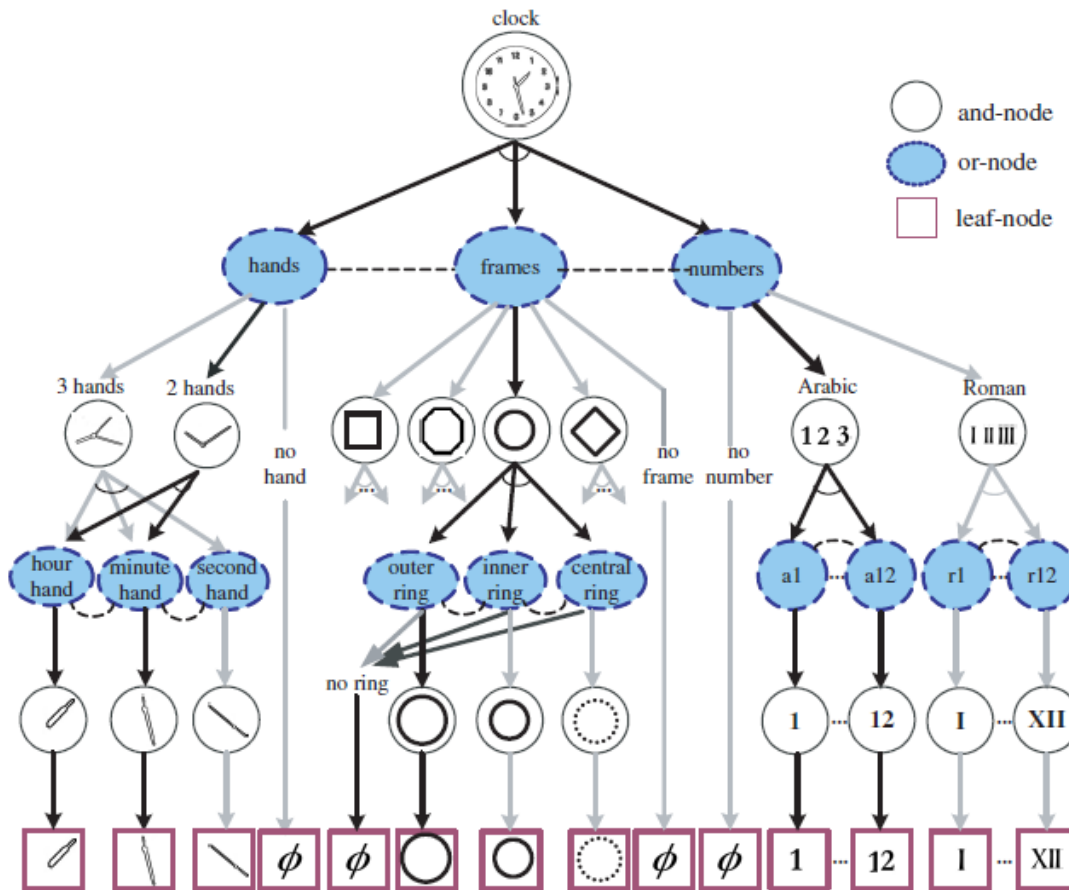


Fig. 6.1 An And-Or graph example for the object category — clock. It has two parse graphs shown in Figure 5.1, one of which is illustrated in dark arrows. Some leaf nodes are omitted from the graph for clarity. From [86].

Foundations and Trends® in
 Computer Graphics and Vision
 Vol. 2, No. 4 (2006) 259–362
 © 2007 S.-C. Zhu and D. Mumford
 DOI: 10.1561/06000000018

Mais exemplos de gramáticas - Detecção de estenoses



ELSEVIER

Artificial Intelligence in Medicine 26 (2002) 145–159

**Artificial
Intelligence
in Medicine**

www.elsevier.com/locate/artmed

Syntactic reasoning and pattern recognition for analysis of coronary artery images

Marek R. Ogiela^{*}, Ryszard Tadeusiewicz

*Institute of Automatics 30 Mickiewicza Avenue, University of Mining and Metallurgy,
PL-30-059, Krakow, Poland*

Received 5 March 2002; received in revised form 23 April 2002; accepted 23 April 2002

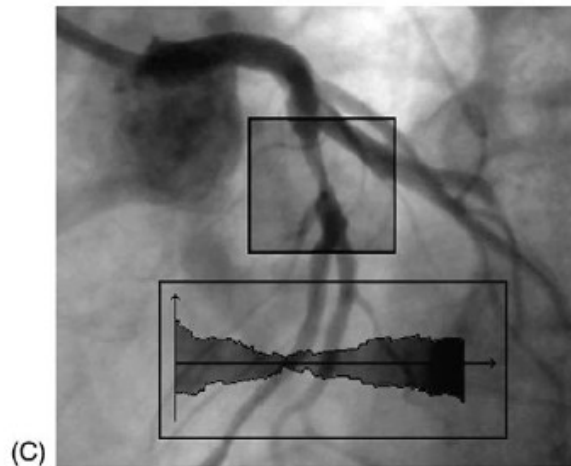
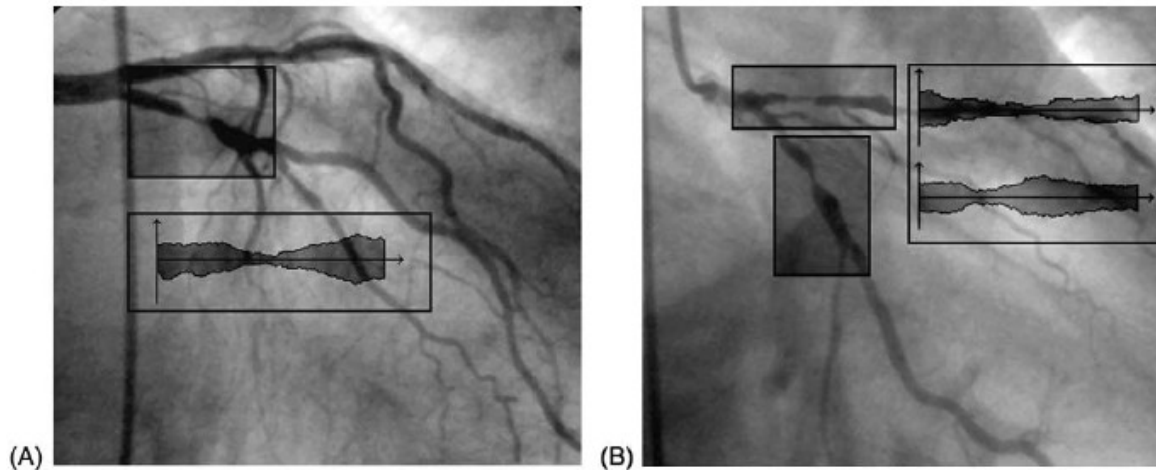


SISTEMAS DE
INFORMAÇÃO



EACH

Mais exemplos de gramáticas - Detecção de estenoses



Mais exemplos de gramáticas - Detecção de estenoses

$$G_{CA} = (V_N, V_T, SP, STP)$$

where V_N is the set of non-terminal symbols, V_T the set of terminal symbols, SP the production set, STS the grammar start symbol.

$$V_N = \{\text{SYMPTOM}, \text{STENOSIS}, H, V, NV\}$$

$$V_T = \{h, v, nv\} \text{ for } h \in (-10^\circ, 10^\circ), v \in (11^\circ, 90^\circ), nv \in (-11^\circ, -90^\circ)$$

STS=SYMPTOM

SP:

1. SYMPTOM \rightarrow STENOSIS
2. STENOSIS \rightarrow NV HV
3. STENOSIS \rightarrow NV VNV H
4. V \rightarrow v|V v
5. NV \rightarrow nv|NV nv
6. H \rightarrow h|H h

SYMPTOM := STENOSIS

$$w_{\text{sym}} := w_{\text{sym}} + w_{\text{nv}}; \quad h_{\text{sym}} := h_{\text{sym}} + h_{\text{nv}}$$

$$w_{\text{sym}} := w_{\text{sym}} + w_v; \quad h_{\text{sym}} := h_{\text{sym}} + h_v$$

$$w_{\text{sym}} := w_{\text{sym}} + w_h; \quad h_{\text{sym}} := h_{\text{sym}} + h_h$$



Revisão de gramáticas em imagens

Using Grammars for Pattern Recognition in Images: A Systematic Review

RICARDO WANDRÉ DIAS PEDRO, FÁTIMA L. S. NUNES,
and ARIANE MACHADO-LIMA, School of Arts, Sciences and Humanities, University of Sao Paulo,
Brazil

ACM Computing Surveys, Vol. 46, No. 2, Article 26, Publication date: November 2013.



Gramáticas para Reconhecimento de Padrões **Sintáticos e Estruturais**

- Uma classe pode ser vista como uma linguagem (conjunto de cadeias)
- Cada classe C_i é representada por uma gramática G_i
- Cada gramática atribui a uma cadeia x uma probabilidade $P(x|G_i)$: verossimilhança
- No caso binário (pertence ou não pertence à linguagem) compara-se a um modelo nulo

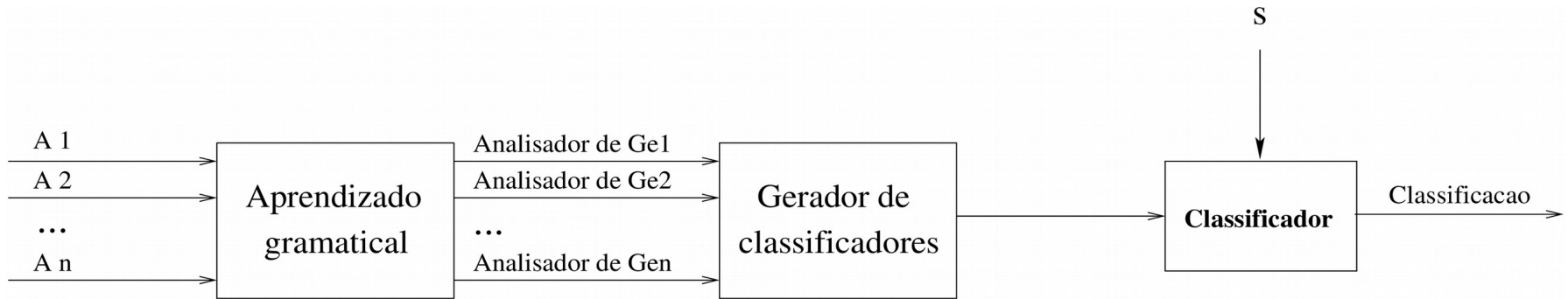
Gramáticas para Reconhecimento de Padrões Sintáticos

- A gramática pode ser desenhada manualmente ou aprendida
- Modelo nulo N: depende da aplicação
 - Pode ser uma distribuição uniforme
- Cálculo de um score:
 - $S(x) = P(x|G) / P(x|N)$ ou
 - $S(x) = \log P(x|G) - \log P(x|N)$

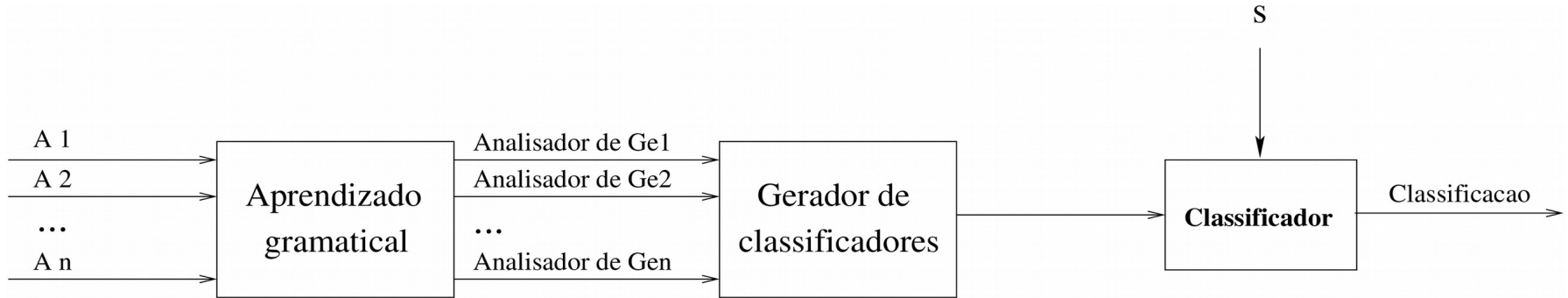
Razão da verossimilhança ou
log-likelihood ratio



GrammarLab



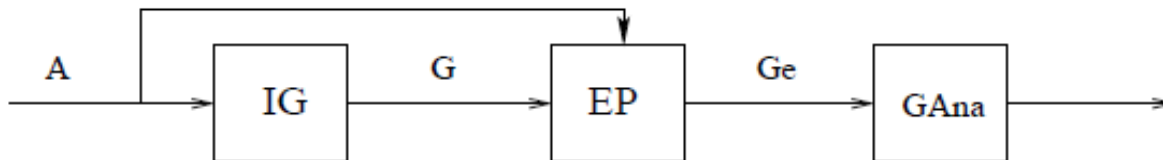
GrammarLab



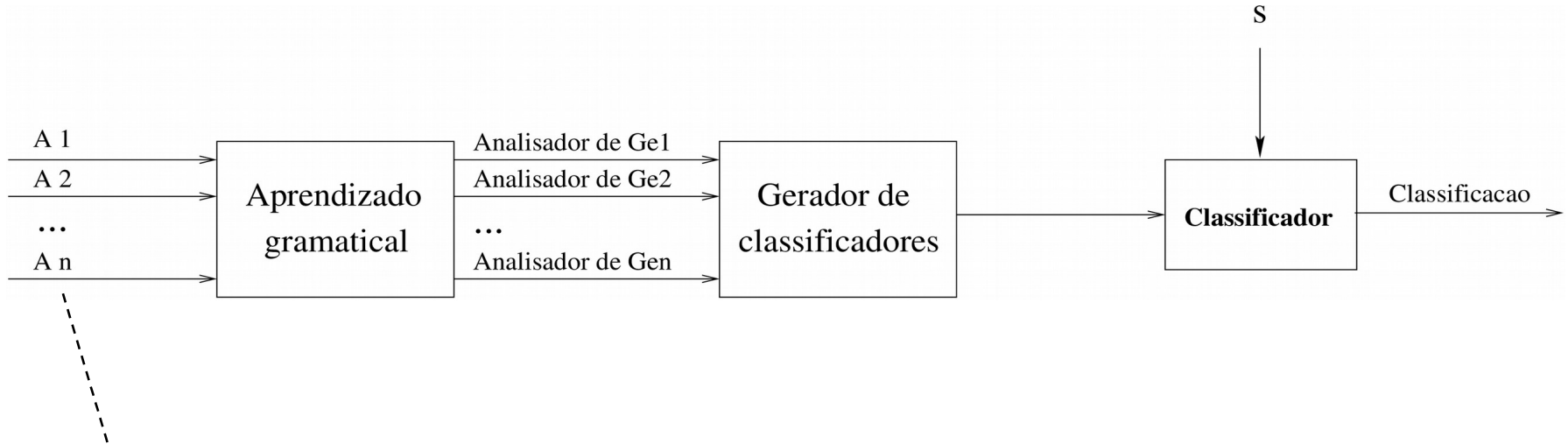
Aprendizado gramatical (supervisionado):

- Aprendizado das produções
- Aprendizado das probabilidades

Passos distintos ou conjugados



GrammarLab



Amostra de treinamento:

- só positiva
- positiva e negativa

Classificação

- No caso determinístico, se a sequência é reconhecida ou não
 - Não paramétrico
- No caso estocástico - classificação bayesiana:
 - comparação com um modelo nulo
 - Paramétrico:
 - Distribuição multinomial sobre as produções com o mesmo lado esquerdo
 - Distribuição de Dirichlet sobre as priori's

Referências

- RAMOS, M. V. M.; NETO, J. J.; VEJA, I. S. **Linguagens Formais: Teoria, Modelagem e Implementação**. Ed. Bookman, 2009.
- SIPSER, M. **Introdução à Teoria da Computação**. Ed. Thomson, 2007
- DUDA, R.; HART, P.; STORK, D. **Pattern Classification and Scene Analysis**. Ed. John Wiley, 2001. Cap 8.6 e 8.7
- DURBIN, R.; EDDY, S. R.; KROGH, A. **Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids**. Cambridge University Press, 2002. Cap 9

Próximas atividades

- Validação cruzada para os classificadores:
 - Naive Bayes (aula de redes bayesianas)
 - Redes Neurais (aula de RN)
 - SVM (aula de SVM)
 - Random Forests (aula de árvores de decisão)
- Ideia: para cada um dos 3, teste de vários parâmetros (específicos de cada um deles), utilizando:
 - todas as características
 - apenas com os componentes principais selecionados
 - apenas com as características selecionadas pelo selecionador 1
 - apenas com as características selecionadas pelo selecionador 2
- Em cada um, calcular os valores médios de revocação (sensibilidade), precisão e erro total.
- Apresentar os resultados em uma tabela (métricas nas colunas, com/sem seleção de características nas linhas) e discuti-los