Gramáticas

Definições

Exemplos de gramáticas

Gramáticas

- Conceito introduzido pela lingüística
  - Objetivo de ensinar o inglês pelo computador e conseguir um tradutor de línguas
- Fracasso da tradução por volta dos anos 50.
- Sucesso na descrição de LP. Algol 60 foi descrito como uma linguagem livre de contexto na notação BNF.
- Atualmente, a tradução voltou a ganhar força dado o sucesso da tradução por métodos estatísticos que aprender a tarefa a partir de grandes conjuntos de dados (temos a WEB a disposição!)

Antes de vermos a noção formal de uma gramática vamos para algumas definições:

Alfabeto ou Vocabulário: Conjunto finito não vazio de símbolos. Símbolo é um elemento qualquer de um alfabeto.

$\delta\varepsilon$

Ex: {A,B,C,….Z} alfabeto latino

{α, β, γ, δ, ε, …, ω} alfabeto grego

{0,1} alfabeto binário

{0,1,2,3,4,5,6,7,8,9} alfabeto de digitos

{a, b}

Cadeia ou palavra: Concatenação de símbolos de um alfabeto. Define-se como cadeia vazia ou nula uma cadeia que não contém nenhum símbolo.

Ex: aab

123094

λ – cadeia nula

Comprimento de cadeia: Número de símbolos de uma cadeia.

Ex: $|aab| = 3$

$|123094|=6$

$|\lambda|=0$

Concatenação de cadeias: Define-se a concatenação z de uma cadeia x com uma cadeia y, como sendo a concatenação dos símbolos de ambas as cadeias, formando a cadeia xy. $|z| = |x| + |y|$

Ex: $x = abaa$; $y = ba \Rightarrow z = abaaba$

$x = ba$; $y = \lambda \Rightarrow z = ba$

Produto de alfabetos: É o produto cartesiano de alfabetos.

Ex: $V1 = \{a,b\}$ $V2 = \{1, 2, 3\} \Rightarrow V1.V2 = V1xV2 = \{a1, a2, a3, b1, b2, b3\}$

Observe que $V1.V2 \neq V2.V1$

Exponenciação de alfabetos: São todas as cadeias de comprimento n sobre V (V^n). $V^0=\{\lambda\}$, $V^1=V$, $V^n=V^{n-1}.V$

Ex: $V = \{0, 1\}$

$V^3 = V^2.V = (V.V).V = \{00, 01, 10, 11\}.\{0, 1\} = \{000, 001, 010, 011, 100, 101, 110, 111\}$

$|V^n| = m^n$ onde $|V| = m$

Fechamento (Clausura) de um Alfabeto: Seja A um alfabeto, então o fechamento de A é definido como $A^* = A^0 \cup A^1 \cup A^2 \cup \ldots \cup A^n \cup \ldots$

Portanto A^* = conjunto das cadeias de qualquer comprimento sobre o alfabeto a.

Ex: $A = \{1\}$
$A^* = \{\lambda, 1, 11, 111, \ldots\}$

Fechamento Positivo de A: $A^+ = A^* - \{\lambda\}$

Linguagem é uma coleção/conjunto de *cadeias* de símbolos sobre um alfabeto/vocabulário. Estas cadeias são denominadas ***sentenças*** **da linguagem**, e são formadas pela justaposição de elementos individuais, os símbolos da linguagem.

Ex: V = {0, 1}

L1 = $\{0^n1^n \mid n >= 1\}$ infinita

L1 = {01, 0011, 000111, ...}

L2 = {ab, bc} finita

(linguagem formada pelas cadeias ab e bc)

L3 = $\{ab^n \cup a^nb \mid n>=0\}$ infinita

(linguagem formada por todas as cadeias que começam com "a" seguido de um número qualquer de "b"'s OU começam com um número qualquer de "a"'s seguidos de um "b", por exemplo ab, abb, aab, aaab, ...)

Finita ou Infinita?

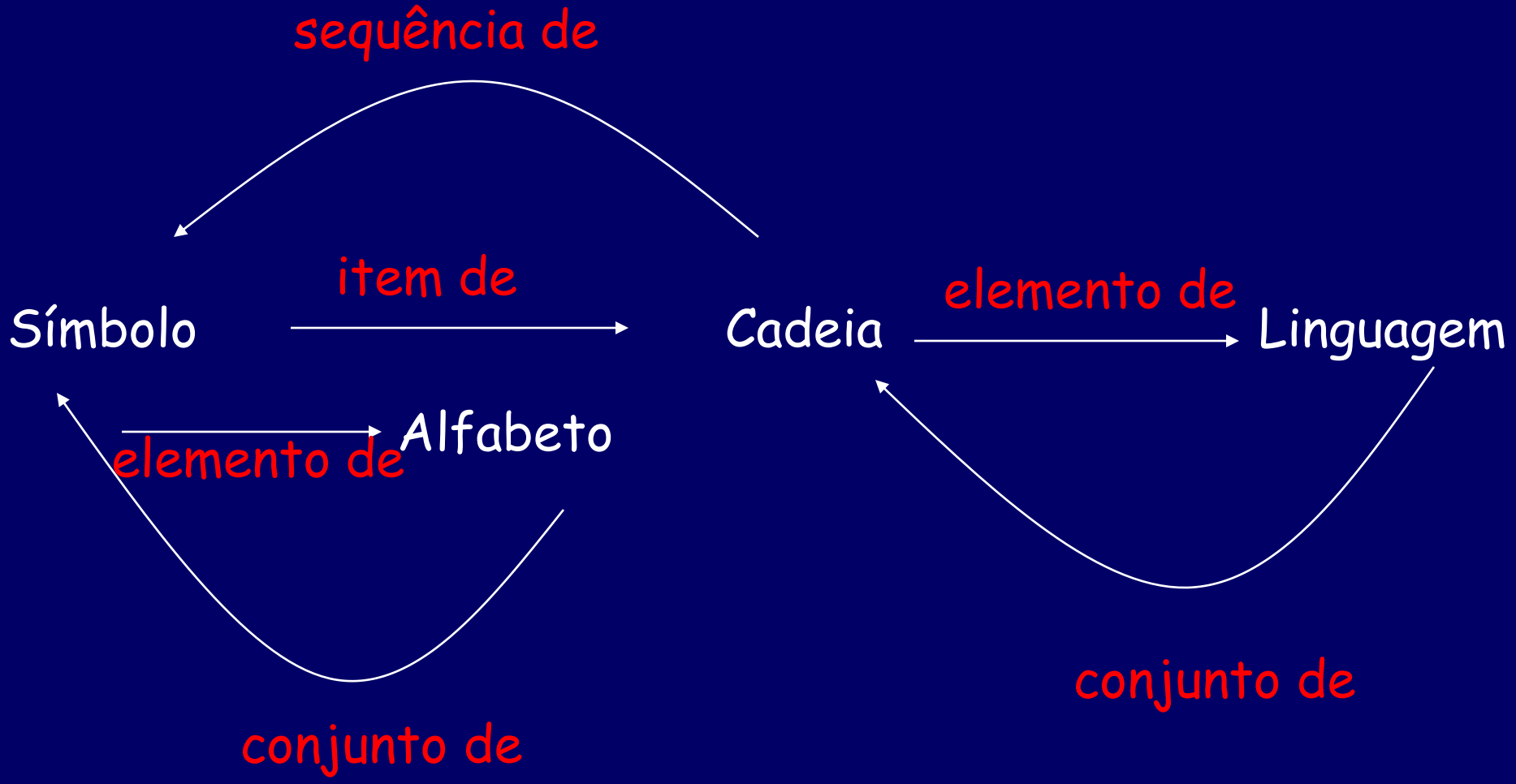
i) L1 = { x | x ∈ {a, b, c}* e x contém ab}

ii) L2 = {w ∈ {0,1}* | w tem pelo menos um 1}

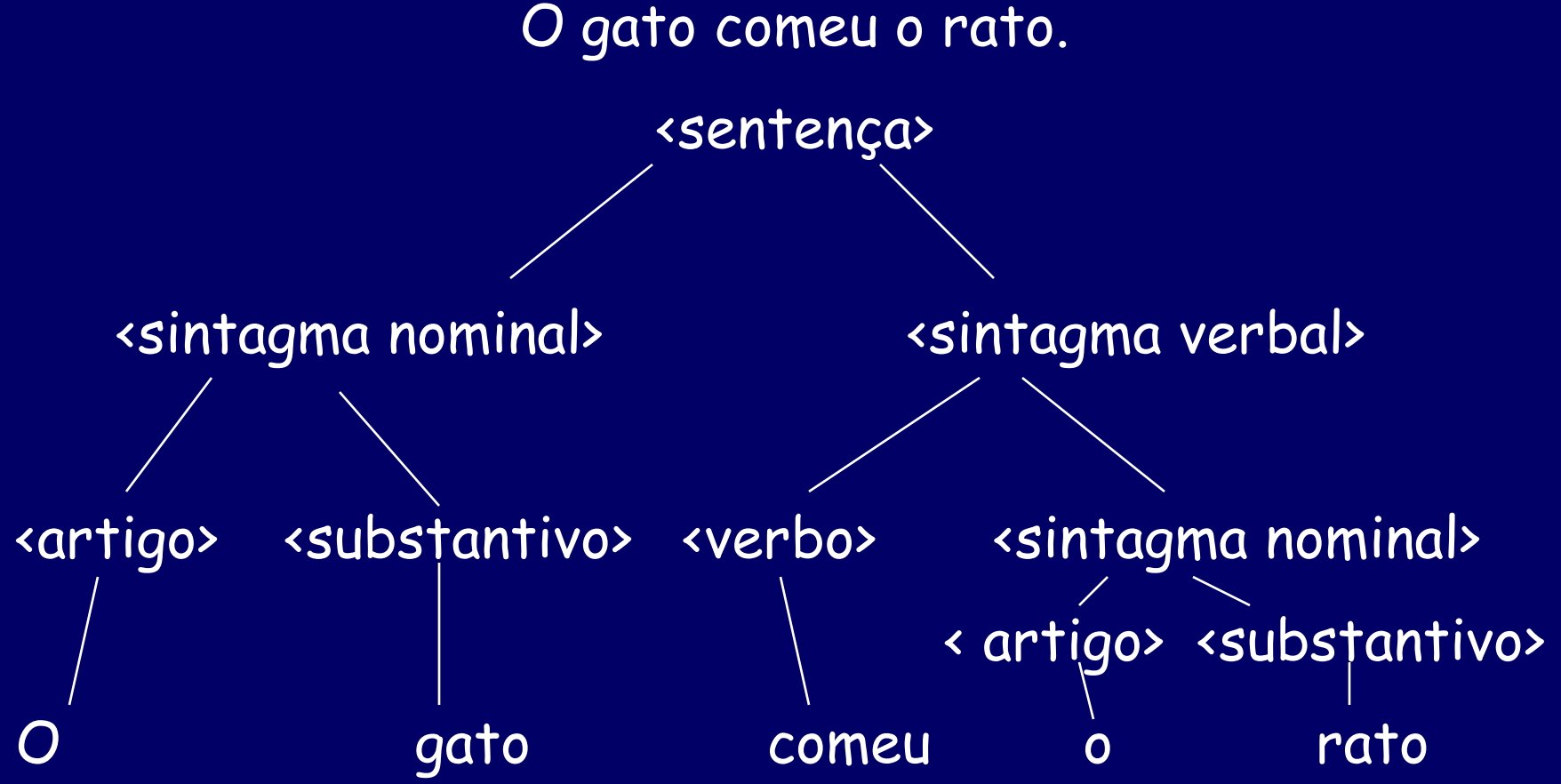
iii) L3 = { x | x é uma cadeia de 0´s e 1´s e |x| >= 2}

iv) L1 = {w ∈ {a,b}* | todo a tem um b imediatamente à sua direita}

v) L2 = {w ∈ {a,b}* | w tem aa como subpalavra}


sequência de
item de
Símbolo
Cadeia
elemento de
Linguagem
Alfabeto
elemento de
conjunto de
conjunto de

Vamos analisar uma sentença do Português colocada no contexto da cadeia alimentar



Regras para analisar a sentença dada

<sentença> -> <sn> <sv>
<sn> -> <artigo> <substantivo>
<sv> -> <verbo> <sn>

<artigo> -> o
<substantivo> -> gato | rato
<verbo> -> comeu

"->" definido por
< > para distinguir sentença, sv, sn das palavras do português
Com essa gramática poderíamos gerar:
1) O gato comeu o rato
2) O rato comeu o gato
que são gramaticalmente corretas mas 2) é inaceitável no contexto dado.

Para formar essa gramática parcial para o português usamos 4 conceitos:

- categorias sintáticas como <sentença>, <sn>, <sv>, <artigo>, <substantivo>, <verbo> são chamados de não terminais ou variáveis
- palavras são chamadas de terminais
- relação entre terminais e não terminais são chamadas de produções
- o não terminal que gera todas as cadeias de terminais (<sentença>) é chamado de símbolo inicial

Formalmente, as gramáticas são caracterizadas como quádruplas ordenadas

$$G = (Vn, Vt, P, S)$$

onde:

Vn representa o vocabulário não terminal da gramática. Este vocabulário corresponde ao conjunto de todos os símbolos dos quais a gramática se vale para definir as leis de formação das sentenças da linguagem.

Vt é o vocabulário terminal, contendo os símbolos que constituem as sentenças da linguagem. Dá-se o nome de terminais aos elementos de Vt.

P representa o conjunto de todas as leis de formação utilizadas pela gramática para definir a linguagem.

Para tanto, cada construção parcial, representada por um não-terminal, é definida como um conjunto de regras de formação relativas à definicão do não-terminal a ela referente. A cada uma destas regras de formação que compõem o conjunto P dá-se o nome de produção da gramática.

Assumimos Vn ∩ Vt = ∅. Convencionamos que Vn U Vt = V
Cada produção P tem a forma:

$$\alpha \rightarrow \beta \qquad \alpha \in V+; \qquad \beta \in V^*$$

S ϵ Vn denota a principal categoria gramatica de *G*; é dito o símbolo inicial ou o axioma da gramática. Indica onde se inicia o processo de geração de sentenças.

Ex.1: *G* = ({S, A, B}, {a, b}, P, S)
P: {S -> AB
A -> a
B -> b}

Notação/Convenções

- Letras do alfabeto latino maiúsculas {A,B,..Z}: variáveis
- Letras do começo do alfabeto latino minúsculas {a,b,c,...}: terminais
- Letras do fim do alfabeto latino minúsculas {t,u,v,x,z}: cadeias de terminais
- Letras gregas minúsculas {α,β,γ,δ,ε, ..., ω}: cadeias de terminais e não terminais

  alfa, beta, gama, delta, épsilon, zeta, eta, teta, iota, kapa, lâmbda, mi, ni, xi, ômicron, pi, rô, sigma, tau, úpsilon, fi, qui, psi, ômega

Exercício

- Tentem definir uma gramática simples, por exemplo, de algum construtor (definição) ou comando de uma linguagem de programação.

Comando de atribuição com variáveis simples em Pascal

G = ({<atribuição>, <variável>, <expressão>, <identificador>}, {:=, letras, digitos, _, operadores, (,)}, P, <atribuição>)

P = {

<atribuição>::= <variável> := <expressão>

<variável> ::= <identificador>

}

<expressão> deve ainda ser definida. Veremos isto mais para frente.

<identificador> ::= é formado por uma letra, seguido de qualquer número de letras, digitos ou underscore (tem fazer esta regra)

Definida uma gramática G, qual é a linguagem gerada por ela?

Precisaremos das relações $\Rightarrow_G$ (deriva diretamente) e $\Rightarrow_G^*$ (deriva) definidas entre as cadeias de V*

Def1. Se $\alpha \rightarrow \beta$ é uma produção de P e γ (gama) e δ (delta) são cadeias quaisquer de V*, então $\gamma\ \alpha\ \delta \Rightarrow_G \gamma\ \beta\ \delta$ (deriva diretamente na gramática G).

Dizemos que a produção $\alpha \rightarrow \beta$ é aplicada à cadeia $\gamma\ \alpha\ \delta$ para obter $\gamma\ \beta\ \delta$. A relação $\Rightarrow_G$ relaciona cadeias exatamente quando a segunda é obtida a partir da primeira pela aplicação de uma única produção.

No Ex.1.: $S \Rightarrow_G AB$; $aB \Rightarrow_G ab$ ou

$S \Rightarrow_G AB \Rightarrow_G aB \Rightarrow_G ab$

Def2. Suponha que $\alpha_1\ \alpha_2\ \alpha_3\ \ldots\ \alpha_m$ são cadeias de V* e $\alpha_1 \Rightarrow_G \alpha_2$, $\alpha_2 \Rightarrow_G \alpha_3$, ..., $\alpha_{m-1} \Rightarrow_G \alpha_m$.
Então dizemos que $\alpha_1 \Rightarrow_G^* \alpha_m$ (deriva).
Aplicamos algum número de produções de P.
Por convenção $\alpha \Rightarrow_G^* \alpha$ para a cadeia α.

No Ex.1.: $S \Rightarrow_G^* ab$;
$S \Rightarrow_G^* aB$;
$AB \Rightarrow_G^* ab$;
$ab \Rightarrow_G^* ab$

Def3. Forma sentencial: uma cadeia α composta de terminais e não terminais se $S \Rightarrow^* \alpha$

No Ex.1: aB, AB, S, ab são formas sentenciais.

Uma forma sentencial, α, é uma sentença de G se $S \Rightarrow^* \alpha$ e $\alpha \in Vt^*$ (são composta de terminais). Ou seja, as cadeias geradas pela gramática são as sentenças de G.

Def4. A Linguagem L gerada por uma gramática G é definida como o conjunto de cadeias geradas por G. Ou seja,

$L(G) = \{x \mid x \in Vt^* \text{ e } S \Rightarrow_G^* x\}$ ou {x | x é sentença de G}

1. A cadeia consiste somente de terminais
2. A cadeia pode ser derivada a partir do símbolo inicial da gramática

Def5. Duas gramáticas G1 e G2 são equivalentes sse L(G1) = L(G2)

Exemplos de Gramáticas

G1 = ({S}, {0,1}, P1, S)
P1: { 1. S -> 0S1
2. S -> 01 }

Qual é a linguagem gerada por G1? Aplicamos o **processo de derivação** para saber L(G1), que é o processo de obtenção de cadeias a partir de uma gramática.

G2 = ({S,B,C}, {a,b,c}, P2, S)
P2: { 1. S -> aSBC 2. S -> aBC
3. CB -> BC 4. aB -> ab
5. bB -> bb 6. bC -> bc
7. cC -> cc}
L(G2) = ?

G1

- A menor cadeia gerada é 01: $S \Rightarrow^2 01$

- Se aplicarmos n-1 vezes a produção 1, seguida da produção 2 teremos:

- $S \Rightarrow 0S1 \Rightarrow 00S11 \Rightarrow 0^3S1^3 \Rightarrow^*$
- $0^{n-1}S1^{n-1} \Rightarrow 0^n1^n$
- Portanto, $L(G1) = \{0^n1^n \mid n >= 1\}$

  ou $S \Rightarrow^* 0^n1^n$

JFLAP : <untitled4>

File Input Convert Help

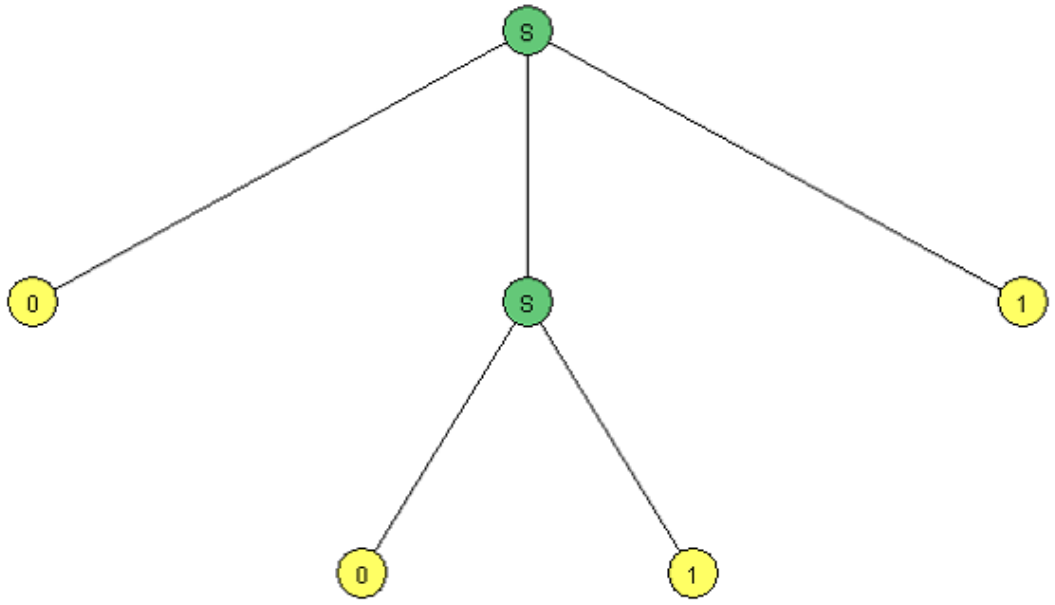
Editor | Brute Parser

Start | Pause | Step | Noninverted Tree

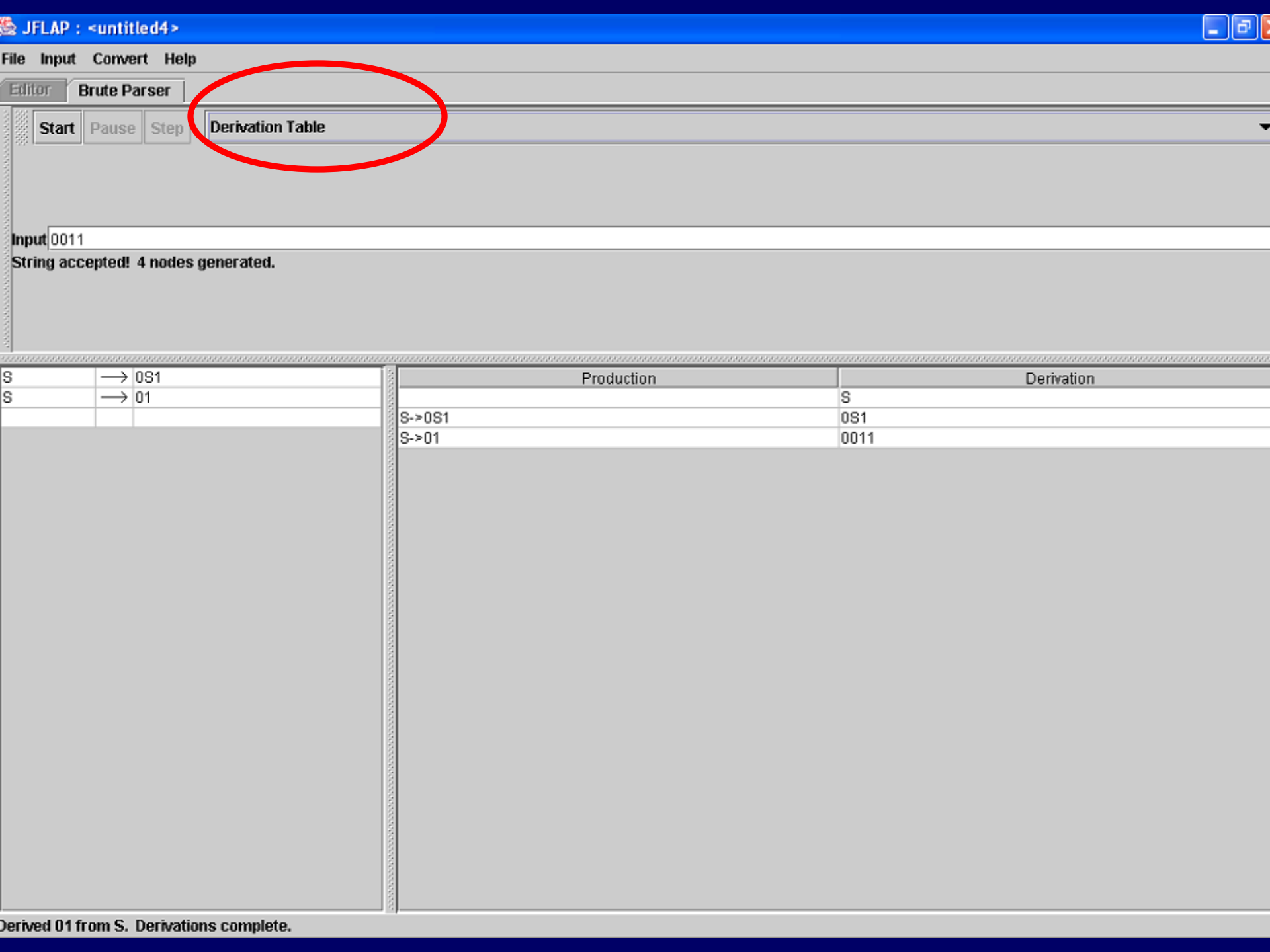
Input: 0011

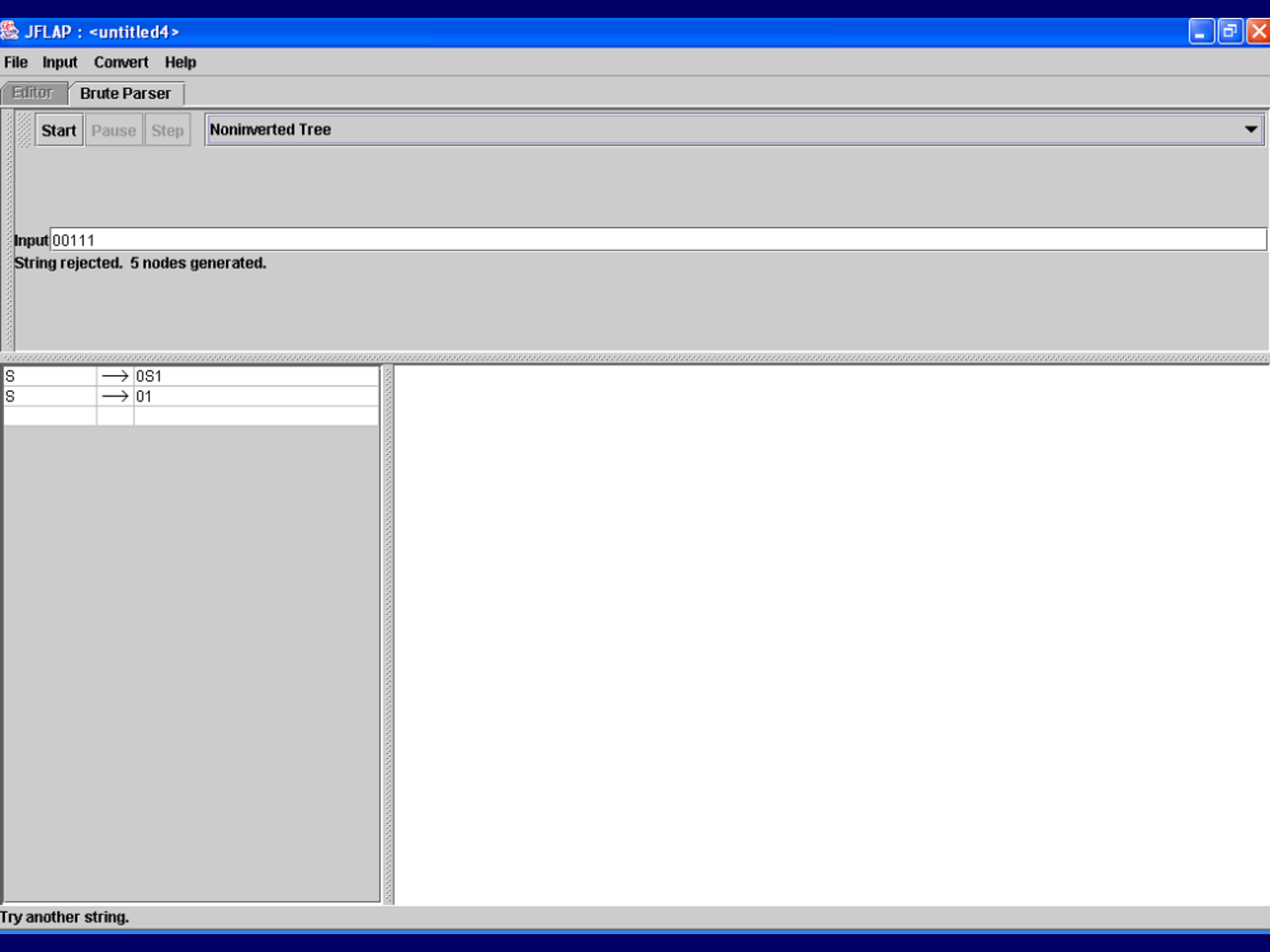
String accepted! 4 nodes generated.

S	→	0S1
S	→	01



Derived 01 from S. Derivations complete.


JFLAP : <untitled4>
File Input Convert Help
Editor Brute Parser
Start Pause Step
Derivation Table
Input 0011
String accepted! 4 nodes generated.
S ⟶ 0S1
S ⟶ 01
Production
Derivation
S
S->0S1
0S1
S->01
0011
Derived 01 from S. Derivations complete.


JFLAP : <untitled4>
File Input Convert Help
Editor Brute Parser
Start Pause Step
Noninverted Tree
Input 00111
String rejected. 5 nodes generated.
S ⟶ 0S1
S ⟶ 01
Try another string.

G2

- A menor cadeia gerada é abc: $S =>^2 aBC =>^4 abC =>^6 abc$
- Usamos 1 n-1 vezes: $S =>^* a^{n-1}S(BC)^{n-1}$
- Usamos a 2 uma vez: $S =>^* a^n(BC)^n$
- A 3 permite trocar B com C para que B´s precedam os C´s
  - Para n = 2 aaBCBC => aaBBCC (usamos a regras 3 1 vez)
  - Para n = 3 aaaBCBCBC => aaaBBCCBC => aaaBBCBCC => aaaBBBCCC (usamos a regra 3 3 vezes)
  - Para n = 4 aaaaBCBCBCBC => aaaaBCBCBBCC => aaaaBCBBCBCC => aaaaBCBBBCCC => aaaaBBCBBCCC => aaaaBBBCBCCC => aaaaBBBBCCCC (usamos a regra 3 6 vezes);
  - Para n = 5 usamos a 3 10 vezes.
- Assim $S =>^* a^nB^nC^n$
- Usamos a 4 uma vez: $S =>^* a^nbB^{n-1}C^n$
- Aplicamos a 5 n-1 vezes: $S =>^* a^nb^nC^n$
- Aplicamos a 6 uma vez: $S =>^* a^nb^ncC^{n-1}$
- Aplicamos a 7 n-1 vezes: $S =>^* a^nb^nc^n$

$$L(G2) = \{a^nb^nc^n \mid n >= 1\}$$

Fecho e a definição de Linguagem

A definição formal de linguagem é baseada em um subconjunto do fecho de um vocabulário terminal, como já vimos:

$L(G) = \{x \mid x \in Vt^* \text{ e } S \Rightarrow_G^* x\}$ ou {x | x é sentença de G}

Então, a maior linguagem sobre um vocabulário Vt é Vt^*.

A menor linguagem sobre um vocabulário Vt é ∅ (conjunto vazio) ou seja, a linguagem vazia, composta por zero cadeias/sentenças. Observe que é diferente do conjunto que tem a cadeia vazia { λ }

O conjunto de todos os subconjuntos possíveis de Vt^* é 2^{Vt^*} e representa o conjunto de todas as linguagens que podem ser definidas a partir de Vt^*.

∅ (conjunto vazio) e também $Vt^* \in 2^{Vt^*}$

Exercício

Seja Vt = {a,b,c} e a propriedade "todas as cadeias são iniciadas pelo símbolo a"

1. Qual a menor linguagem?
2. Qual a maior linguagem?

3. A linguagem {a, ab, ac, abc, acb} satisfaz a propriedade acima?
4. A linguagem {a} {a}* {b}* {c}* satisfaz a propriedade acima?