Linguagens Formais e Autômatos

Prof. Sergio D Zorzo
Departamento de Computação - UFSCar

1º semestre / 2017

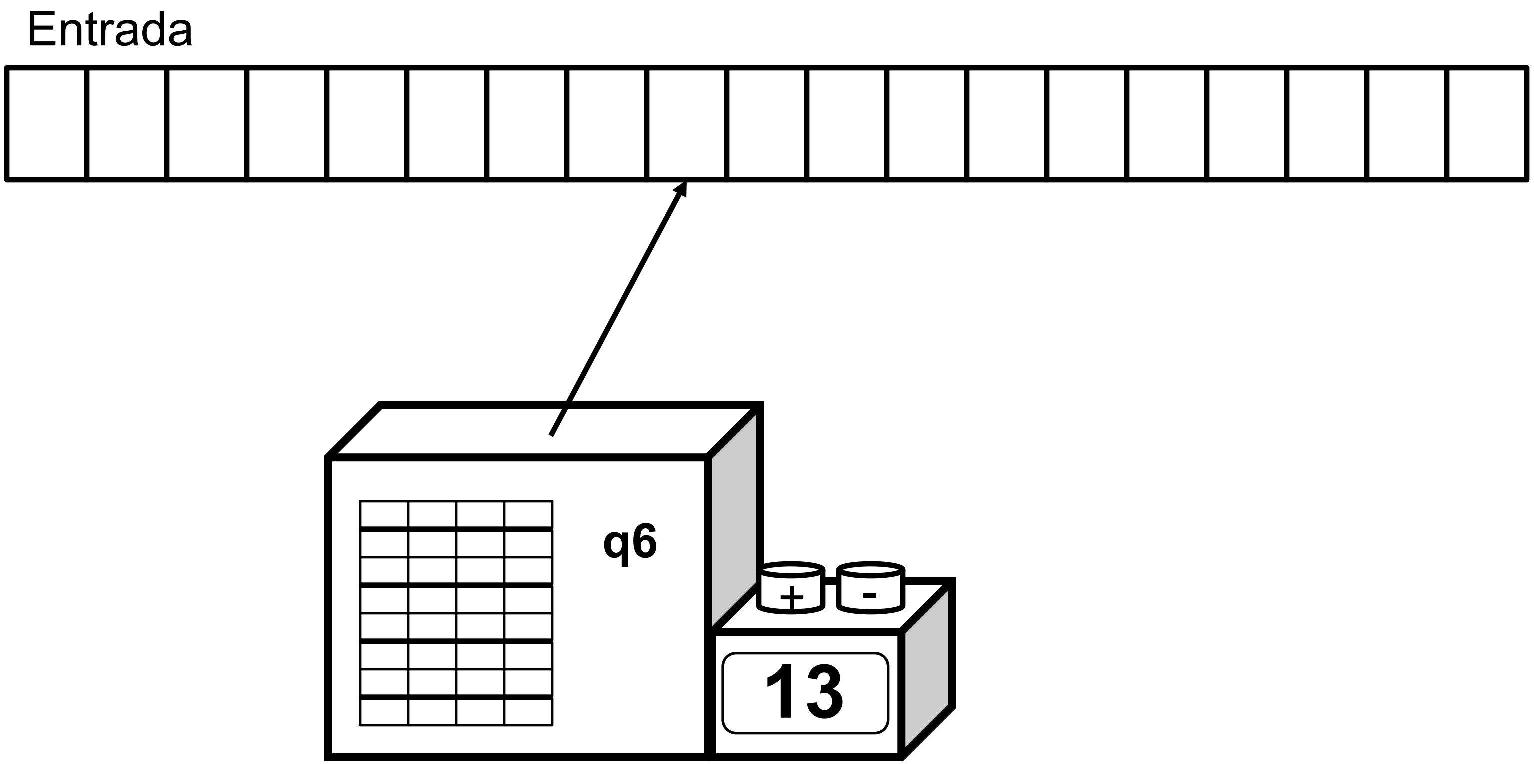
Aula 09

Linguagens livres de contexto

Linguagens livres de contexto

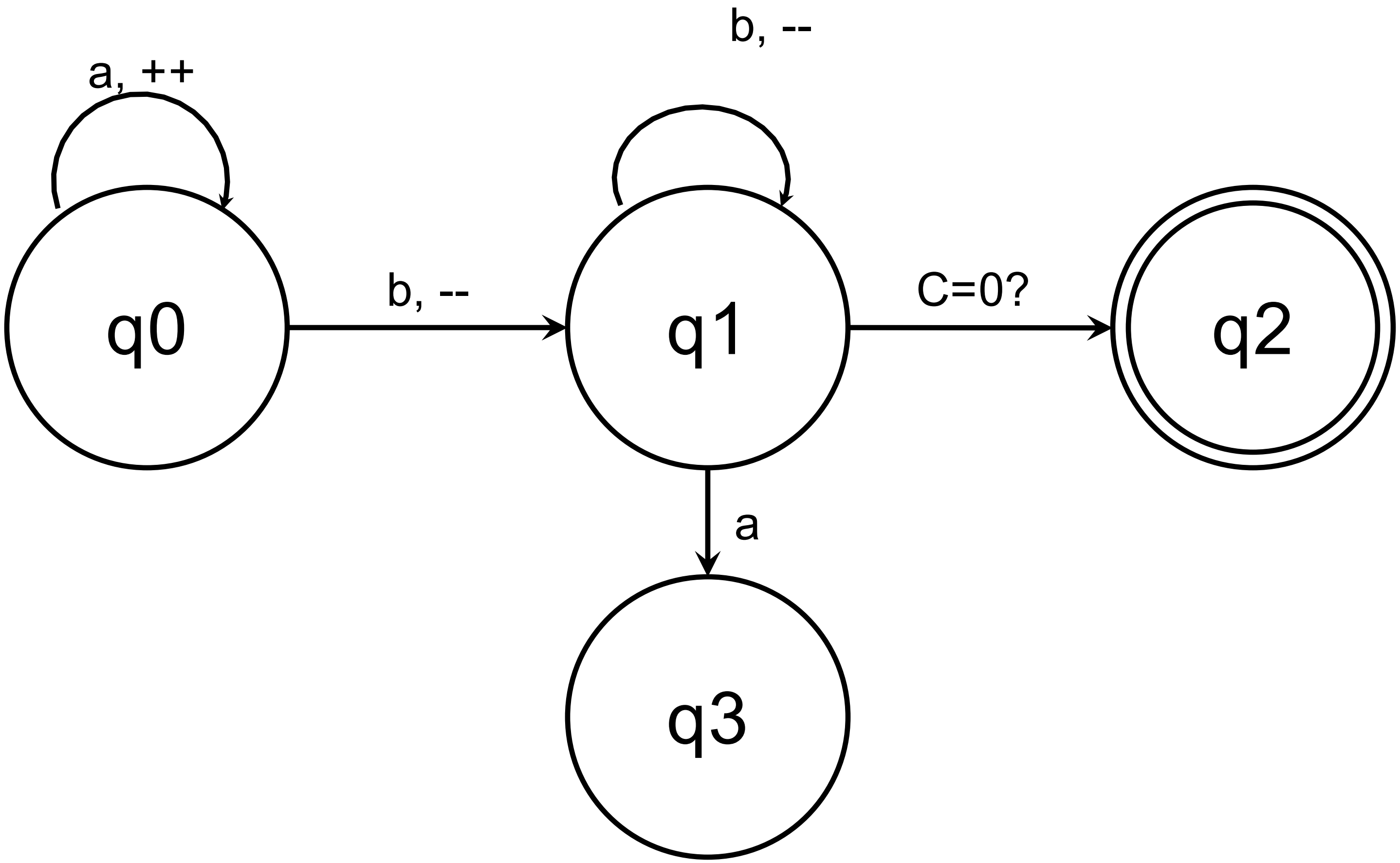
- Linguagens regulares permitem descrever muitas coisas práticas
  - Ex: busca textual, mecanismos simples de comunicação, máquinas e protocolos simples
- Mas são limitadas
  - "Não conseguem contar" → autômatos finitos
- Mas e se adicionarmos um contador aos autômatos finitos?

Linguagens livres de contexto



Linguagens livres de contexto

- Ex: $a^n b^n$



Linguagens livres de contexto

- Uma classe maior de linguagens
  - Linguagens livres de contexto

- Inicialmente, foram uma maneira de entender a linguagem humana
  - Gramáticas livres de contexto
    - Formalização das regras gramaticais da linguagem humana

- Característica principal: recursão
  - Ex: linguagens naturais: frases nominais dentro de frases verbais, e vice-versa

Linguagens livres de contexto

- A partir do estudo da linguagem humana, chegou-se a um modelo matemático formal sobre uma classe de linguagens
- O conceito é o mesmo das linguagens regulares
  - Linguagens = problemas
  - Um problema é:
    - Dada uma cadeia, determinar se pertence ou não à linguagem
    - A diferença é que aqui, o conceito cadeia é diferente ...
    - ... e as regras de definição da linguagem são mais poderosas do que simples transições em um autômato finito

Linguagens livres de contexto

- Podemos comparar linguagens regulares e linguagens livres de contexto nos seguintes aspectos:
  - Linguagens regulares:
    - Base: símbolos de um alfabeto
    - Regras ("gramática"): expressões regulares
    - Característica:
      - estados finitos / não conseguem contar / lema do bombeamento para linguagens regulares
  - Linguagens livres de contexto:
    - Base: símbolos terminais
      - Pode-se pensar que um símbolo terminal é um símbolo de um alfabeto (Mas cada símbolo terminal pode ser uma cadeia sobre uma linguagem regular - uma palavra)
    - Regras ("gramática"): gramáticas livres de contexto
    - Característica:
      - recursão simples / consegue contar (de forma limitada) / lema do bombeamento para linguagens livres de contexto

Gramáticas livres de contexto

Gramáticas livres de contexto

- As linguagens livres de contexto são descritas por gramáticas livres de contexto
- As regras de produção tem leis de formação específica

$$A \rightarrow 0A1$$
$$A \rightarrow B$$
$$B \rightarrow \#$$

Regras de substituição
ou
produções

Gramáticas livres de contexto

$A \rightarrow 0A1$

$A \rightarrow B$

$B \rightarrow \#$

Lado esquerdo ou **cabeça**: sempre um único símbolo. Esses símbolos são chamados de **variáveis** ou **não-terminais**

Lado direito ou **corpo**: uma cadeia de símbolos. Podem ter variáveis e outros símbolos, chamados de **terminais**

Uma das variáveis é designada como a variável ou símbolo inicial. É a variável que aparece do lado esquerdo da primeira regra. (Neste exemplo, **A** é o símbolo inicial)

Gramáticas livres de contexto

$A \rightarrow 0A1$
$A \rightarrow B$
$B \rightarrow \#$

$\Rightarrow$

$A \rightarrow 0A1 \mid B$
$B \rightarrow \#$

Sempre que houver mais de uma produção para uma mesma variável, podemos agrupá-las com o símbolo "|".

Gramáticas livres de contexto

- Variáveis normalmente são representadas por letras maiúsculas (A,B,C, ...)
- Terminais normalmente são representados por letras minúsculas, números, símbolos, etc... (a,b,c,0,1,2,#,...)
  - Assim como nos alfabetos que estudamos até então
- Mas na prática usamos nomes mais significativos

```
OraçãoSubstantiva  → SubstantivoComplexo
                   |  SubstantivoComplexo FrasePreposicional ;
FraseVerbal  → VerboComplexo
             |   VerboComplexo  FrasePreposicional  ;
FrasePreposicional  → Preposição  SubstantivoComplexo  ;
SubstantivoComplexo  → Artigo  Substantivo  | OraçãoSubstantiva  ;
VerboComplexo  → Verbo  | Verbo  OraçãoSubstantiva  ;
Artigo  → o  |  a  |  um  |  uma  |  ε ;
Substantivo  → armário  |  funcionário  |  gerente  |  cadeira  |  braços  |  aço  ;
Verbo  → gosta  |  brinca  |  olha  ;
Preposição  → para  |  com  |  de  |  sem  ;
```

Gramáticas livres de contexto

- Terminais podem ser cadeias (sequências) simples:
  - Ex: “menino”, “if”, “while”, “int”, “char”, etc...
- Podem também ser cadeias descritas com linguagens regulares
  - Ex: identificadores válidos de uma linguagem de programação
    - `[a-zA-Z][a-zA-Z0-9]*`
      - Ex: `posX1, valorAluguel, rgCliente`
    - `[0-9]+` : números inteiros

  - Em compiladores este tipo de terminal é muito usado
    - Mas para o estudo das características das linguagens livres de contexto (esta disciplina), tanto faz, portanto usaremos com mais frequência símbolos simples, como a,b,0,1

Gramáticas livres de contexto

A GLC G é definida pela quadrupla (V,T,P,S)

- V = conjunto de variáveis
- T = conjunto de terminais
- P = conjunto de produções
- S = símbolo inicial

Ex: $G_{palíndromos} = (\{P\},\{0,1\},A,P)$

- A = {
  - $P \rightarrow \varepsilon$
  - $P \rightarrow 0$
  - $P \rightarrow 1$
  - $P \rightarrow 0P0$
  - $P \rightarrow 1P1$
- }

Gramáticas livres de contexto

- Como uma gramática descreve uma linguagem?
- Duas formas:
  - Inferência recursiva
  - Derivação
- Ex: Gramática para expressões aritméticas
  - V = {E,I}
  - T = {+,*,(,),a,b,0,1}
  - P = conjunto de regras ao lado
  - S = E

$$
\begin{aligned}
E &\rightarrow I \\
E &\rightarrow E + E \\
E &\rightarrow E * E \\
E &\rightarrow (E) \\
I &\rightarrow a \\
I &\rightarrow b \\
I &\rightarrow Ia \\
I &\rightarrow Ib \\
I &\rightarrow I0 \\
I &\rightarrow I1
\end{aligned}
$$

Gramáticas livres de contexto

- Inferência recursiva
  - Dada uma cadeia (conjunto de símbolos terminais)
  - Do corpo para a cabeça
- Ex: a*(a+b00)
  - a*(a+b00) ⇐ a*(a+I00) ⇐ a*(a+I0) ⇐ a*(a+I) ⇐ a*(a+E) ⇐ a*(I+E) ⇐ a*(E+E) ⇐ a*(E) ⇐ a*E ⇐ I*E ⇐ E*E ⇐ E

Gramáticas livres de contexto

- Derivação
  - Dada uma cadeia (conjunto de símbolos terminais)
  - Da cabeça para o corpo
- Ex: a*(a+b00)
  - $E \Rightarrow E*E \Rightarrow I*E \Rightarrow a*E \Rightarrow a*(E) \Rightarrow a*(E+E) \Rightarrow a*(I+E) \Rightarrow a*(a+E) \Rightarrow a*(a+I) \Rightarrow a*(a+I0) \Rightarrow a*(a+I00) \Rightarrow a*(a+b00)$
- Símbolo de derivação: $\Rightarrow$
- Derivação em múltiplas etapas: $\Rightarrow^*$ (obs: asterisco acima da seta)
  - $E \Rightarrow^* a*(E)$
  - $a*(E+E) \Rightarrow^* a*(a+I00)$
  - $E \Rightarrow^* a*(a+b00)$

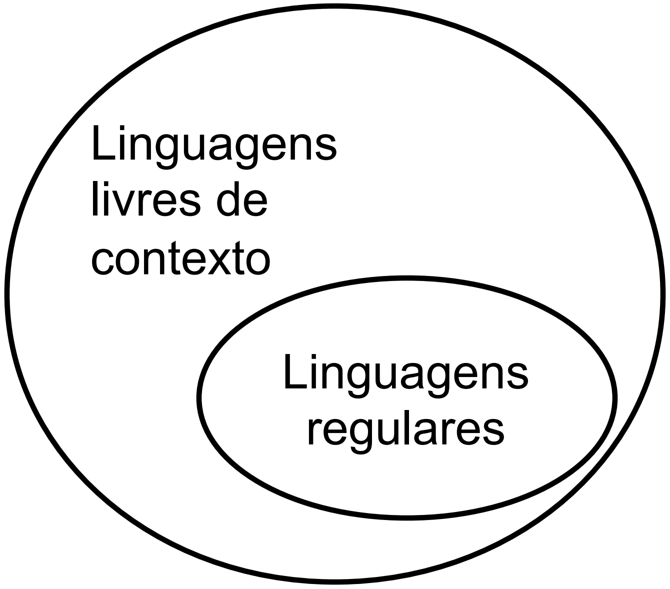
Gramáticas livres de contexto

- Derivações mais à esquerda
  - Sempre substituir a variável mais à esquerda
  - Notação: $\Rightarrow_{lm}$, $\Rightarrow^*_{lm}$
- Derivações mais à direita
  - Sempre substituir a variável mais à direita
  - Notação: $\Rightarrow_{rm}$, $\Rightarrow^*_{rm}$

Gramáticas livres de contexto

- A linguagem de uma gramática
  - $L(G) = \{w \text{ em } T^* \mid S \Rightarrow^* w\}$
- Se uma linguagem L é L(G) de alguma gramática G livre de contexto
  - L é uma linguagem livre de contexto
    - (CFL – *Context-Free Language*)
- Ex: o conjunto de palíndromos pode ser descrito por uma gramática livre de contexto
  - Portanto o conjunto de palíndromos é uma linguagem livre de contexto

Gramáticas livres de contexto



A classe de linguagens livres de contexto engloba a classe de linguagens regulares

Ou seja: toda linguagem regular é livre de contexto

Formas sentenciais

- Derivações a partir do símbolo inicial
  - Formas sentenciais à esquerda
    - Obtidas somente com derivações mais à esquerda
  - Formas sentenciais à direita
    - Obtidas somente com derivações mais à direita
- Ex: $E \Rightarrow_{lm} E*E \Rightarrow_{lm} I*E \Rightarrow_{lm} a*E \Rightarrow_{lm} a*(E)$
- Ex: $E \Rightarrow_{rm} E*E \Rightarrow_{rm} E*(E) \Rightarrow_{rm} a*(E+E) \Rightarrow_{rm} a*(E+I)$

Exercícios

- Dada a gramática descrita pelas produções a seguir:
  - $S \rightarrow A1B$
  - $A \rightarrow 0A \mid \varepsilon$
  - $B \rightarrow 0B \mid 1B \mid \varepsilon$
- Forneça derivações mais à esquerda e mais à direita das seguintes cadeias:
  - a) 00101
  - b) 1001
  - c) 00011

Exercícios

- Dada a gramática descrita pelas produções ao lado:
- Forneça derivações mais à esquerda e mais à direita das seguintes cadeias:
  - a) a+b*(0+1)
  - b) a+a+b+1
  - c) a+b*a

$$
\begin{aligned}
E &\rightarrow I \\
E &\rightarrow E + E \\
E &\rightarrow E * E \\
E &\rightarrow (E) \\
I &\rightarrow a \\
I &\rightarrow b \\
I &\rightarrow 0 \\
I &\rightarrow 1
\end{aligned}
$$

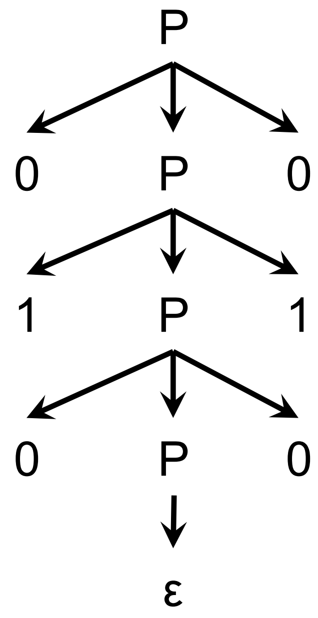
Árvores de análise sintática

Árvores de análise sintática

- Representação visual para derivações
  - Em formato de árvore
- Mostra claramente como os símbolos de uma cadeia de terminais estão agrupados em subcadeias
- Permitem analisar alguns aspectos da linguagem e ver o processo de derivação / inferência recursiva

Árvores de análise sintática

- Ex: palíndromos, cadeia 010010
- Derivações: $P \Rightarrow 0P0 \Rightarrow 01P10 \Rightarrow 010P010 \Rightarrow 010\varepsilon010 = 010010$



Árvores de análise sintática

- Seja uma gramática G = (V,T,P,S)
- A árvore é construída da seguinte forma:
  - Cada nó interior é rotulado por uma variável em V
  - Cada folha é rotulada por uma variável, um terminal, ou ε. No entanto, se a folha for rotulada por ε, ela deve ser o único filho de seu pai
  - Se um nó interior é rotulado por A e seus filhos são rotulados por

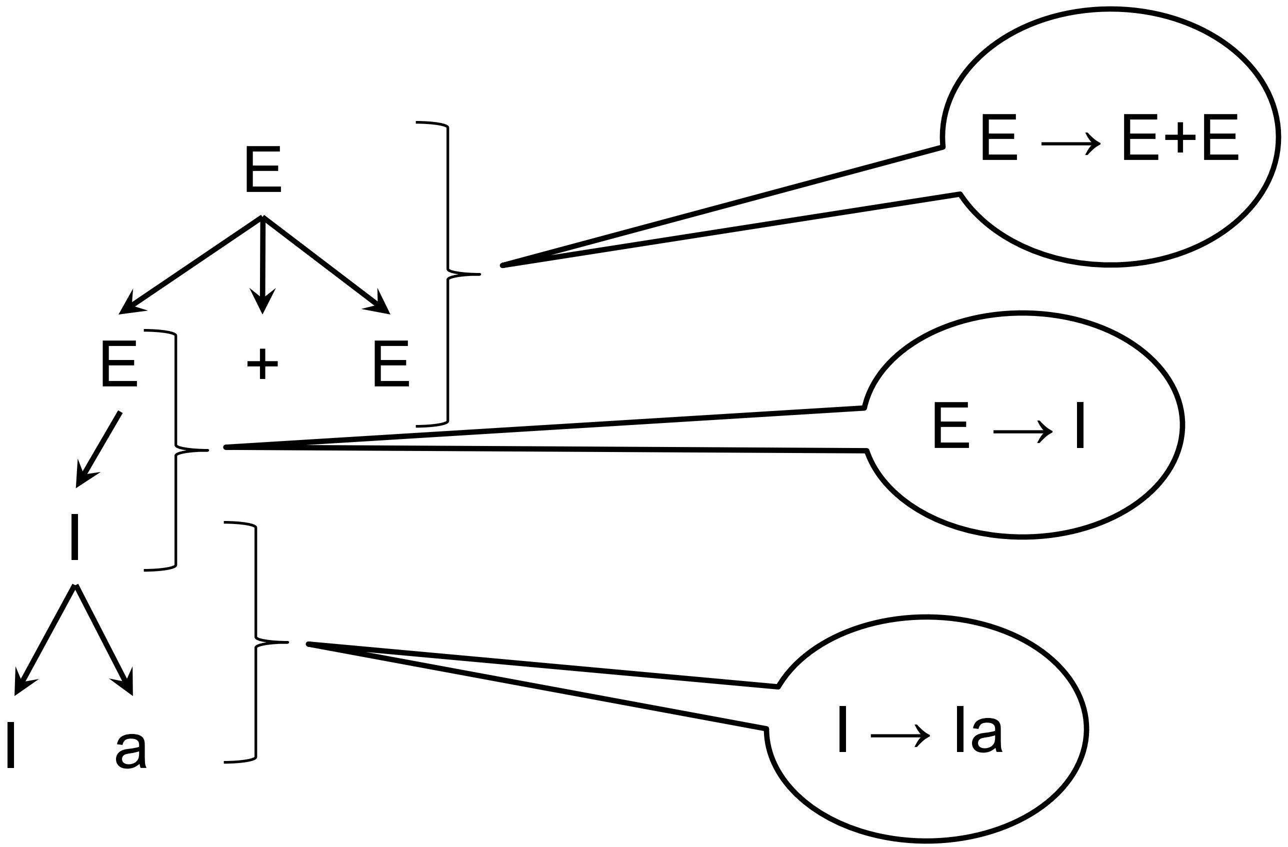
    $$X_1, X_2, \ldots, X_k$$

    Respectivamente, a partir da esquerda, então

    $A \rightarrow X_1X_2\ldots X_k$ é uma produção em P

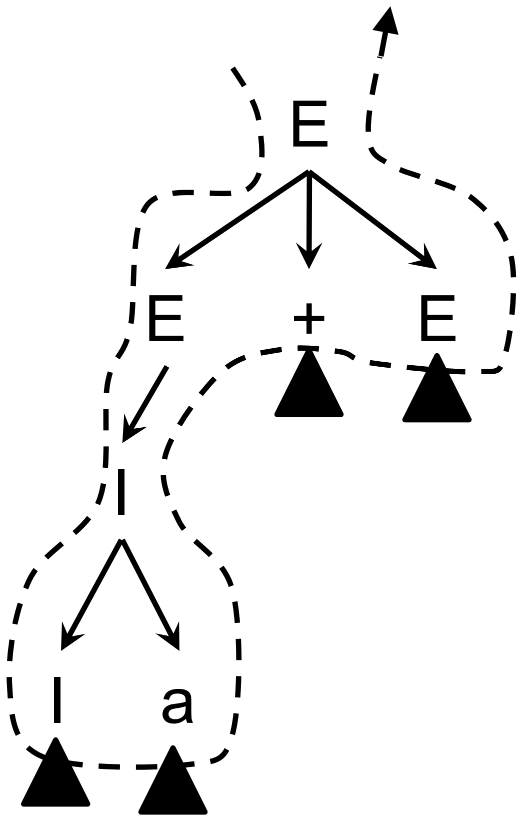
Árvores de análise sintática

- Ex:



O resultado de uma árvore de análise sintática

- Análise das folhas de uma árvore de análise sintática
  - Concatenação dos símbolos a partir da esquerda
    - ***Resultado*** da árvore
    - É sempre uma cadeia derivada da variável raiz



Ex: Ia+E é o resultado da árvore à direita
É também uma derivação a partir da raiz (E)
$E \Rightarrow E+E \Rightarrow I+E \Rightarrow Ia+E$

Árvores de análise sintática

- Árvores especiais
  - O resultado é uma cadeia composta exclusivamente por terminais
    - Isto é, as folhas são rotuladas por um terminal ou ε
  - A raiz é rotulada pelo símbolo inicial
- São árvores cujo resultado é uma cadeia na linguagem da gramática subjacente
  - Este tipo de árvore é útil para analisar alguns aspectos da linguagem, como ambiguidade (veremos mais adiante)

Exercícios

- Dada a gramática descrita pelas produções a seguir:
  - $S \rightarrow A1B$
  - $A \rightarrow 0A \mid \varepsilon$
  - $B \rightarrow 0B \mid 1B \mid \varepsilon$
- Forneça árvores de análise sintática para as seguintes cadeias:
  - a) 00101
  - b) 1001
  - c) 00011

Exercícios

- Dada a gramática descrita pelas produções ao lado:
- Forneça árvores de análise sintática para as seguintes cadeias:
  - a) a+b*(0+1)
  - b) a+a+b+1
  - c) a+b*a

$$
\begin{aligned}
E &\rightarrow I \\
E &\rightarrow E + E \\
E &\rightarrow E * E \\
E &\rightarrow (E) \\
I &\rightarrow a \\
I &\rightarrow b \\
I &\rightarrow 0 \\
I &\rightarrow 1
\end{aligned}
$$

Aplicações das gramáticas livres de contexto

- Foram concebidas originalmente para descrever linguagens naturais
  - Essa promessa não se concretizou
- Mas existem hoje aplicações
  - Descrevem linguagens de programação
    - Existe um modo mecânico de transformar a descrição da linguagem na forma de uma gramática livre de contexto em um analisador sintático, o componente do compilador que descobre a estrutura do código-fonte
    - É uma das primeiras formas de uso das idéias teóricas da ciência da computação

Aplicações das gramáticas livres de contexto

- Aplicações
  - Desenvolvimento baseado em XML (eXtensible Markup Language)
    - Formato projetado para facilitar comércio eletrônico e troca de informações em geral
    - Documento estruturado, com tags do tipo abre-fecha

```
<cliente>
  <nome>João</nome>
  <telefone>1241-1231</telefone>
</cliente>
```

    - A estrutura do documento é definida com um DTD ou XMLSchema
      - Essencialmente, é uma gramática livre de contexto

Aplicações das gramáticas livres de contexto

- Aplicações
  - Linguagens específicas de domínio
    - Linguagens menores, aplicadas a um domínio específico
    - Mais simples do que as linguagens de programação, e portanto, mais "gerenciáveis"
    - Também usadas para transferência de informações, como o XML
      - Porém, sem a "poluição" das tags
      - Mais intuitivas
    - Podem ser usadas com geradores de código

Fim