

Análise sintática

Análise sintática *top-down*

Prof. Edson Alves

Faculdade UnB Gama

Sumário

1. Análise sintática *top-down*

Análise sintática recursiva-descendente

- ▶ A análise sintática *top-down* pode ser interpretada como uma tentativa de se encontrar uma derivação mais à esquerda para uma cadeia da entrada

Análise sintática recursiva-descendente

- ▶ A análise sintática *top-down* pode ser interpretada como uma tentativa de se encontrar uma derivação mais à esquerda para uma cadeia da entrada
- ▶ Outra forma de interpretar esta análise é como uma tentativa de construção de uma árvore gramatical a partir da raiz, criando os nós da árvore em pré-ordem

Análise sintática recursiva-descendente

- ▶ A análise sintática *top-down* pode ser interpretada como uma tentativa de se encontrar uma derivação mais à esquerda para uma cadeia da entrada
- ▶ Outra forma de interpretar esta análise é como uma tentativa de construção de uma árvore gramatical a partir da raiz, criando os nós da árvore em pré-ordem
- ▶ No caso geral, a análise sintática recursiva-descendente pode envolver algum nível de retrocesso

Análise sintática recursiva-descendente

- ▶ A análise sintática *top-down* pode ser interpretada como uma tentativa de se encontrar uma derivação mais à esquerda para uma cadeia da entrada
- ▶ Outra forma de interpretar esta análise é como uma tentativa de construção de uma árvore gramatical a partir da raiz, criando os nós da árvore em pré-ordem
- ▶ No caso geral, a análise sintática recursiva-descendente pode envolver algum nível de retrocesso
- ▶ Um caso especial da análise sintática recursiva-descente é o dos analisadores preditivos, que não demandam retrocesso

Retrocesso

- ▶ Certas gramáticas impõem ao analisador recursivo-descendente a necessidade de um retrocesso na análise

Retrocesso

- ▶ Certas gramáticas impõem ao analisador recursivo-descendente a necessidade de um retrocesso na análise
- ▶ Por exemplo, considere a entrada $w = cad$ e a gramática

$$\begin{aligned} S &\rightarrow cAd \\ A &\rightarrow ab \mid a \end{aligned}$$

Retrocesso

- ▶ Certas gramáticas impõem ao analisador recursivo-descendente a necessidade de um retrocesso na análise
- ▶ Por exemplo, considere a entrada $w = cad$ e a gramática

$$\begin{aligned} S &\rightarrow cAd \\ A &\rightarrow ab \mid a \end{aligned}$$

- ▶ A análise inicia construindo a raiz da árvore sintática, rotulada como S

Retrocesso

- ▶ Certas gramáticas impõem ao analisador recursivo-descendente a necessidade de um retrocesso na análise
- ▶ Por exemplo, considere a entrada $w = cad$ e a gramática

$$\begin{aligned} S &\rightarrow cAd \\ A &\rightarrow ab \mid a \end{aligned}$$

- ▶ A análise inicia construindo a raiz da árvore sintática, rotulada como S
- ▶ Ao ler o primeiro símbolo da entrada, o caractere c , o analisador expande a produção de S para gerar os filhos e obter

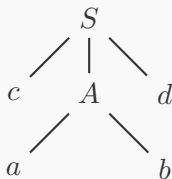


Retrocesso

- ▶ O filho mais à esquerda reconhece o caractere c , de modo que o apontado avança para o segundo símbolo da entrada, o caractere a

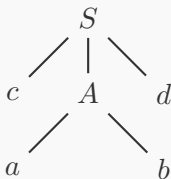
Retrocesso

- ▶ O filho mais à esquerda reconhece o caractere c , de modo que o apontado avança para o segundo símbolo da entrada, o caractere a
- ▶ A análise usa a primeira alternativa das produções- A para obter



Retrocesso

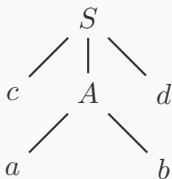
- ▶ O filho mais à esquerda reconhece o caractere c , de modo que o apontado avança para o segundo símbolo da entrada, o caractere a
- ▶ A análise usa a primeira alternativa das produções- A para obter



- ▶ A folha à esquerda reconhece o caractere a , mas a folha à direita, rotulada b , não reconhece o caractere d

Retrocesso

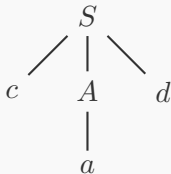
- ▶ O filho mais à esquerda reconhece o caractere c , de modo que o apontado avança para o segundo símbolo da entrada, o caractere a
- ▶ A análise usa a primeira alternativa das produções- A para obter



- ▶ A folha à esquerda reconhece o caractere a , mas a folha à direita, rotulada b , não reconhece o caractere d
- ▶ Aqui acontece o retrocesso: os filhos são descartados e, ao retornar ao nó A , é usada a segunda alternativa das produções- A

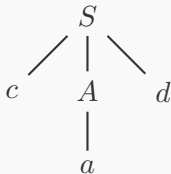
Retrocesso

- ▶ Esta nova substituição resulta em



Retrocesso

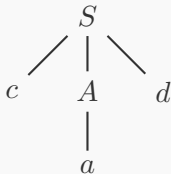
- ▶ Esta nova substituição resulta em



- ▶ Nesta nova árvore o caractere a é reconhecido, e a folha mais à esquerda reconhece o caractere d , de modo que esta árvore produz a cadeia w , finalizando a análise sintática

Retrocesso

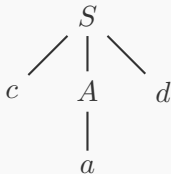
- ▶ Esta nova substituição resulta em



- ▶ Nesta nova árvore o caractere a é reconhecido, e a folha mais à esquerda reconhece o caractere d , de modo que esta árvore produz a cadeia w , finalizando a análise sintática
- ▶ Gramáticas recursivas à esquerda podem levar a laços infinitos, mesmo com o retrocesso

Retrocesso

- ▶ Esta nova substituição resulta em



- ▶ Nesta nova árvore o caractere a é reconhecido, e a folha mais à esquerda reconhece o caractere d , de modo que esta árvore produz a cadeia w , finalizando a análise sintática
- ▶ Gramáticas recursivas à esquerda podem levar a laços infinitos, mesmo com o retrocesso
- ▶ Isto pode acontecer com múltiplas expansões de um não-terminal que não consomem a entrada

Analísadores sintáticos preditivos

- Uma escrita cuidadosa da gramática, por meio de eliminação de recursão à esquerda e do uso de fatorações à esquerda, pode dispensar complementamente o retrocesso

Analísadores sintáticos preditivos

- ▶ Uma escrita cuidadosa da gramática, por meio de eliminação de recursão à esquerda e do uso de fatorações à esquerda, pode dispensar complementamente o retrocesso
- ▶ Neste caso, a análise sintática recursiva-descendente se torna uma análise gramatical preditiva

Analísadores sintáticos preditivos

- ▶ Uma escrita cuidadosa da gramática, por meio de eliminação de recursão à esquerda e do uso de fatorações à esquerda, pode dispensar complementamente o retrocesso
- ▶ Neste caso, a análise sintática recursiva-descendente se torna uma análise gramatical preditiva
- ▶ Na análise gramatical preditiva a alternativa a se escolhida deve ser detectável examinando-se apenas o primeiro símbolo da cadeia que a mesma deriva

Analísadores sintáticos preditivos

- ▶ Uma escrita cuidadosa da gramática, por meio de eliminação de recursão à esquerda e do uso de fatorações à esquerda, pode dispensar complementamente o retrocesso
- ▶ Neste caso, a análise sintática recursiva-descendente se torna uma análise gramatical preditiva
- ▶ Na análise gramatical preditiva a alternativa a se escolhida deve ser detectável examinando-se apenas o primeiro símbolo da cadeia que a mesma deriva
- ▶ Por exemplo, as produções abaixo permitem uma análise preditiva:

```
cmd  →  if expr then cmd else cmd  
      |  while expr do cmd  
      |  begin lista_de_commands end
```

Diagramas de transição para analisadores sintáticos preditivos

- ▶ Assim como foi feito na análise léxica, é possível criar diagramas de transição para analisadores sintáticos

Diagramas de transição para analisadores sintáticos preditivos

- ▶ Assim como foi feito na análise léxica, é possível criar diagramas de transição para analisadores sintáticos
- ▶ Cada não-terminal da gramática deve ter um diagrama próprio

Diagramas de transição para analisadores sintáticos preditivos

- ▶ Assim como foi feito na análise léxica, é possível criar diagramas de transição para analisadores sintáticos
- ▶ Cada não-terminal da gramática deve ter um diagrama próprio
- ▶ Os rótulos das arestas são tokens e não-terminais

Diagramas de transição para analisadores sintáticos preditivos

- ▶ Assim como foi feito na análise léxica, é possível criar diagramas de transição para analisadores sintáticos
- ▶ Cada não-terminal da gramática deve ter um diagrama próprio
- ▶ Os rótulos das arestas são tokens e não-terminais
- ▶ Uma transição rotulada por um token deve ser seguida se o token do rótulo for o próximo token da entrada

Diagramas de transição para analisadores sintáticos preditivos

- ▶ Assim como foi feito na análise léxica, é possível criar diagramas de transição para analisadores sintáticos
- ▶ Cada não-terminal da gramática deve ter um diagrama próprio
- ▶ Os rótulos das arestas são tokens e não-terminais
- ▶ Uma transição rotulada por um token deve ser seguida se o token do rótulo for o próximo token da entrada
- ▶ Uma transição rotulada por um não-terminal A implica em uma passagem pelo diagrama de A

Geração de diagrams de transição para analisadores sintáticos preditivos

Input: uma gramática G

Output: os diagrams de transição para todos os não-terminais de G

- 1: Elimine qualquer recursão à esquerda de G , se necessário
- 2: Fatore G à esquerda, se necessário
- 3: **for** cada não-terminal A de G **do**
- 4: crie um estado inicial e um estado final (de retorno)
- 5: **for** cada produção $A \rightarrow X_1X_2 \dots X_n$ **do**
- 6: crie um percurso que parte do estado inicial até o estado final com arestas rotuladas por X_1, X_2, \dots, X_n

Comportamento do analisador sintático preditivo

- ▶ O analisador parte do estado inicial do símbolo de partida

Comportamento do analisador sintático preditivo

- ▶ O analisador parte do estado inicial do símbolo de partida
- ▶ Estando o analisador no estado s , e se este possui uma aresta rotulada com o token a apontando para o estado t , e o próximo token da entrada é a , o analisador reconhece a e se move para o estado t

Comportamento do analisador sintático preditivo

- ▶ O analisador parte do estado inicial do símbolo de partida
- ▶ Estando o analisador no estado s , e se este possui uma aresta rotulada com o token a apontando para o estado t , e o próximo token da entrada é a , o analisador reconhece a e se move para o estado t
- ▶ Se a aresta é rotulada pelo não-terminal A , o analisador segue para o estado inicial de A , sem reconhecer o próximo token da entrada

Comportamento do analisador sintático preditivo

- ▶ O analisador parte do estado inicial do símbolo de partida
- ▶ Estando o analisador no estado s , e se este possui uma aresta rotulada com o token a apontando para o estado t , e o próximo token da entrada é a , o analisador reconhece a e se move para o estado t
- ▶ Se a aresta é rotulada pelo não-terminal A , o analisador segue para o estado inicial de A , sem reconhecer o próximo token da entrada
- ▶ Se, em algum momento, o analisador atinge o estado final de A , ele deve seguir para t , tendo “lido A ” (isto é, reconhecendo quaisquer tokens que surgiram no caminho do estado inicial ao final de A)

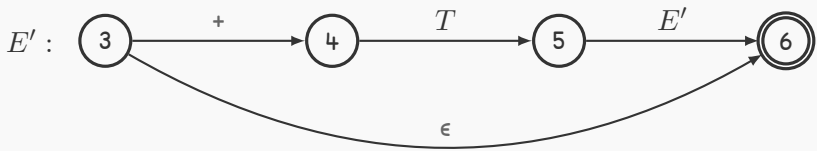
Comportamento do analisador sintático preditivo

- ▶ O analisador parte do estado inicial do símbolo de partida
- ▶ Estando o analisador no estado s , e se este possui uma aresta rotulada com o token a apontando para o estado t , e o próximo token da entrada é a , o analisador reconhece a e se move para o estado t
- ▶ Se a aresta é rotulada pelo não-terminal A , o analisador segue para o estado inicial de A , sem reconhecer o próximo token da entrada
- ▶ Se, em algum momento, o analisador atinge o estado final de A , ele deve seguir para t , tendo “lido A ” (isto é, reconhecendo quaisquer tokens que surgiram no caminho do estado inicial ao final de A)
- ▶ Se a aresta é rotulada por ϵ , o analisador segue para t sem reconhecer a entrada

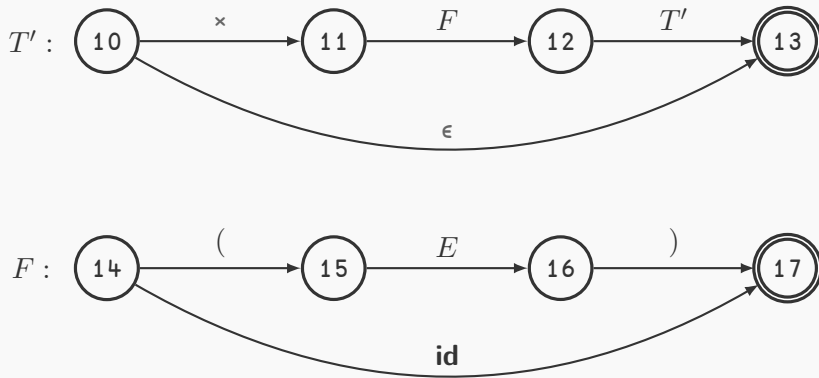
Exemplo de gramática G sem recursão à esquerda e fatorada à esquerda

$$\begin{aligned} E &\rightarrow TE' \\ E' &\rightarrow +TE' \mid \epsilon \\ T &\rightarrow FT' \\ T' &\rightarrow \times FT' \mid \epsilon \\ F &\rightarrow (E) \mid \mathbf{id} \end{aligned}$$

Diagramas de transição para a análise sintática preditiva da gramática G



Diagramas de transição para a análise sintática preditiva da gramática G



Limitações da análise gramatical preditiva

- ▶ A análise gramatical preditiva só pode ser realizada se os diagramas dos não-terminais forem determinísticos, isto é, caso não exista mais de uma transição de um mesmo estado para outros para a mesma entrada

Limitações da análise gramatical preditiva

- ▶ A análise gramatical preditiva só pode ser realizada se os diagramas dos não-terminais forem determinísticos, isto é, caso não exista mais de uma transição de um mesmo estado para outros para a mesma entrada
- ▶ Caso exista ambiguidades, estas devem ser resolvidas de forma *ad hoc*

Limitações da análise gramatical preditiva

- ▶ A análise gramatical preditiva só pode ser realizada se os diagramas dos não-terminais forem determinísticos, isto é, caso não exista mais de uma transição de um mesmo estado para outros para a mesma entrada
- ▶ Caso exista ambiguidades, estas devem ser resolvidas de forma *ad hoc*
- ▶ Se não for possível eliminar as ambiguidades, não será possível realizar uma análise sintática preditiva

Limitações da análise gramatical preditiva

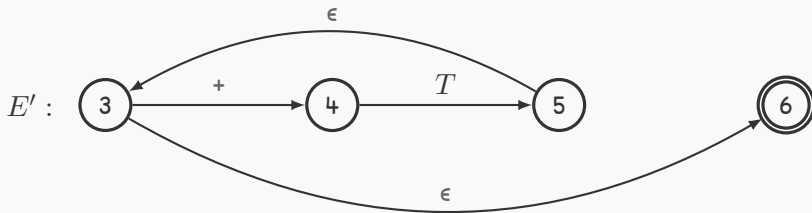
- ▶ A análise gramatical preditiva só pode ser realizada se os diagramas dos não-terminais forem determinísticos, isto é, caso não exista mais de uma transição de um mesmo estado para outros para a mesma entrada
- ▶ Caso exista ambiguidades, estas devem ser resolvidas de forma *ad hoc*
- ▶ Se não for possível eliminar as ambiguidades, não será possível realizar uma análise sintática preditiva
- ▶ Neste caso, será preciso conduzir uma análise sintática recursiva-descendente com retrocesso, avaliando cada um dos caminhos possíveis

Simplificação dos diagramas de transição

- ▶ Os diagramas de transição podem ser simplificados por meio da aplicação da substituição de uns nos outros e da observação das produções

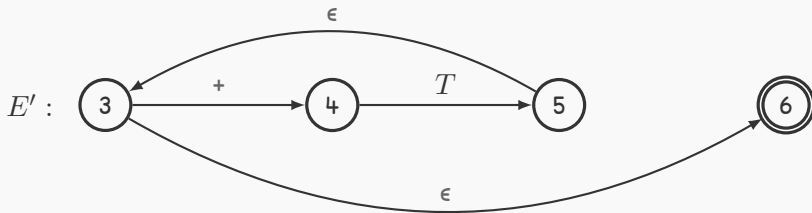
Simplificação dos diagramas de transição

- ▶ Os diagramas de transição podem ser simplificados por meio da aplicação da substituição de uns nos outros e da observação das produções
- ▶ Por exemplo, o diagrama de E' pode ser modificado por meio de uma aresta retornando diretamente para E' , sem recursão:



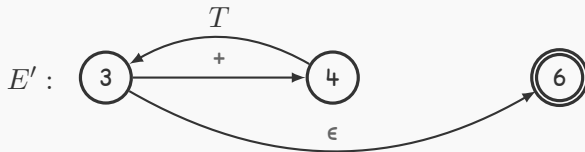
Simplificação dos diagramas de transição

- ▶ Os diagramas de transição podem ser simplificados por meio da aplicação da substituição de uns nos outros e da observação das produções
- ▶ Por exemplo, o diagrama de E' pode ser modificado por meio de uma aresta retornando diretamente para E' , sem recursão:



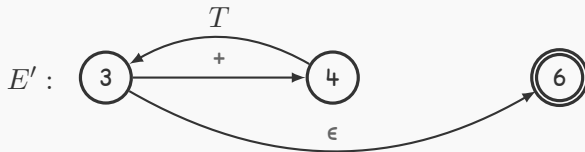
Simplificação dos diagramas de transição

- A transição- ϵ de 5 para 3 pode ser eliminada, de modo que o estado 5 pode ser removido:

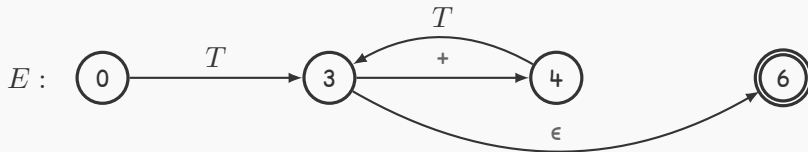


Simplificação dos diagramas de transição

- ▶ A transição- ϵ de 5 para 3 pode ser eliminada, de modo que o estado 5 pode ser removido:

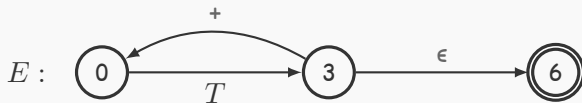


- ▶ Estas modificações podem ser inseridas no diagrama de E , resultando em



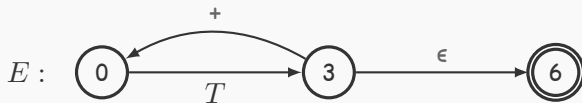
Simplificação dos diagramas de transição

- Por fim, o estado 4 também pode ser eliminado:



Simplificação dos diagramas de transição

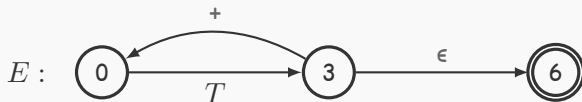
- ▶ Por fim, o estado 4 também pode ser eliminado:



- ▶ Aplicando técnicas semelhantes os diagramas apresentados anteriormente podem ser simplificados, reduzindo substancialmente o número de estados e consequentemente a memória usada pelo analisador sintático

Simplificação dos diagramas de transição

- ▶ Por fim, o estado 4 também pode ser eliminado:



- ▶ Aplicando técnicas semelhantes os diagramas apresentados anteriormente podem ser simplificados, reduzindo substancialmente o número de estados e consequentemente a memória usada pelo analisador sintático
- ▶ A simplificação também pode eliminar recursões, reduzindo o tempo de execução do analisador sintático

Diagramas simplificados para expressões aritméticas

