



LINGUAGENS FORMAIS E AUTÓMATOS / COMPILADORES

ANÁLISE SINTÁTICA ASCENDENTE

Artur Pereira / Miguel Oliveira e Silva <{artur,mos}@ua.pt>

DETI, Universidade de Aveiro

ANÁLISE SINTÁTICA ASCENDENTE

- Considere a gramática

$$\begin{array}{lcl} D & \rightarrow & T L ; \\ T & \rightarrow & i \mid r \\ L & \rightarrow & v \mid L , v \end{array}$$

que representa uma declaração de variáveis *a la C*

- Como reconhecer a palavra “ $u = i \ v , v ;$ ” como pertencente à linguagem definida pela gramática dada?
- Se u pertence à linguagem definida pela gramática, então $D \Rightarrow^+ u$
- Tente-se chegar lá andando no sentido contrário ao de uma derivação, ie. de u para D

ANÁLISE SINTÁTICA ASCENDENTE



$$\begin{aligned} D &\rightarrow T L ; \\ T &\rightarrow i \mid r \\ L &\rightarrow v \mid L , v \end{aligned}$$



$$\begin{aligned} i \ v , v ; & \\ \Leftarrow T \ v , v ; & \quad (\text{por aplicação da produção } T \rightarrow i) \\ \Leftarrow T \ L , v ; & \quad (\text{por aplicação da produção } L \rightarrow v) \\ \Leftarrow T \ L ; & \quad (\text{por aplicação da produção } L \rightarrow L , v) \\ \Leftarrow D & \quad (\text{por aplicação da produção } D \rightarrow T L ;) \end{aligned}$$

- Colocando ao contrário

$$D \Rightarrow T L ; \Rightarrow T L , v ; \Rightarrow T v , v ; \Rightarrow i v , v ;$$

vê-se que corresponde a uma derivação à direita

ANÁLISE SINTÁTICA ASCENDENTE

- A tabela seguinte mostra como, na prática, se realiza esta (retro)derivação

pilha	entrada	ação
\$	i v , v ; \$	deslocamento
\$ i	v , v ; \$	redução por $T \rightarrow i$
\$ T	v , v ; \$	deslocamento
\$ T v	, v ; \$	redução por $L \rightarrow v$
\$ T L	, v ; \$	deslocamento
\$ T L ,	v ; \$	deslocamento
\$ T L , v	; \$	redução por $L \rightarrow L , v$
\$ T L	; \$	deslocamento
\$ T L ;	\$	redução por $D \rightarrow T L ;$
\$ D	\$	aceitação

ANÁLISE SINTÁTICA ASCENDENTE

- Veja-se a reação deste procedimento a uma entrada errada, por exemplo a palavra $i \ v \ v \ ;$.

pilha	entrada	ação
\$	$i \ v \ v \ ; \ \$$	deslocamento
\$ i	$v \ v \ ; \ \$$	redução por $T \rightarrow i$
\$ T	$v \ v \ ; \ \$$	deslocamento
\$ $T \ v$	$v \ ; \ \$$	rejeição

- Se se transformar v em L , atinge-se uma configuração que não corresponde ao prefixo de nenhuma produção existente

ANÁLISE SINTÁTICA ASCENDENTE: CONFLITOS

- O procedimento acabado de descrever pode acarretar situações ambíguas chamadas **conflitos**. Considere a gramática

$$\begin{array}{l} S \rightarrow i \ c \ S \\ \quad | \ i \ c \ S \ e \ S \\ \quad | \ a \end{array}$$

e a palavra `i c i c a e a`

pilha	entrada	ação
\$	i c i c a e a \$	deslocamento
\$ i	c i c a e a \$	deslocamento
\$ i c	i c a e a \$	deslocamento
\$ i c i	c a e a \$	deslocamento
\$ i c i c	a e a \$	deslocamento
\$ i c i c a	e a \$	redução por $S \rightarrow a$
\$ i c i c S	e a \$	conflito: redução por $S \rightarrow i \ c \ S$ ou deslocamento para tentar $S \rightarrow i \ c \ S \ e \ S \ ?$

ANÁLISE SINTÁTICA ASCENDENTE: CONFLITOS

- Também pode haver conflitos entre reduções (*reduce-reduce conflict*). Considere a gramática

$$\begin{array}{l} S \rightarrow A \\ \quad | B \\ A \rightarrow c \\ \quad | A a \\ B \rightarrow c \\ \quad | B b \end{array}$$

e a palavra c

pilha	entrada	ação
\$	c \$	deslocamento
\$ c	\$	conflito: redução usando $A \rightarrow c$ ou $B \rightarrow c$?

ANÁLISE SINTÁTICA ASCENDENTE: CONFLITOS

- Veja-se agora a situação de um falso conflito. Considere a gramática

$$\begin{aligned} S &\rightarrow a \mid (S) \mid a P \mid (S) S \\ P &\rightarrow (S) \mid (S) S \end{aligned}$$

e reconheça-se a palavra $a (a) a$

pilha	entrada	ação
\$	$a (a) a \$$	deslocamento
\$ a	$(a) a \$$	redução usando $S \rightarrow a$ ou deslocamento para tentar $S \rightarrow a P$?

- Deslocamento, porque se se optasse pela redução no topo da pilha ficaria um S e $(\notin \text{follow}(S)$

ANÁLISE SINTÁTICA ASCENDENTE: CONFLITOS

- Optando pelo deslocamento e continuando...

pilha	entrada	ação
\$	a (a) a \$	deslocamento
\$ a	(a) a \$	deslocamento, porque (\notin follow (S)
\$ a (a) a \$	deslocamento
\$ a (a) a \$	redução por $S \rightarrow a$
\$ a (S) a \$	deslocamento
\$ a (S)	a \$	deslocamento, porque a \notin follow (P)
\$ a (S) a	\$	redução por $S \rightarrow a$
\$ a (S) S	\$	redução por $P \rightarrow (S) S$
\$ a P	\$	redução por $S \rightarrow a P$
\$ S	\$	aceitação

ANÁLISE SINTÁTICA ASCENDENTE: CONFLITOS

- Pode ser possível alterar uma gramática de modo a eliminar a fonte de conflito
- Considerando que se pretendia optar pelo deslocamento, a gramática seguinte gera a mesma linguagem que a do slide 6 e está isenta de conflitos.

$$\begin{array}{l} S \rightarrow a \\ \quad | \ i \ c \ S \\ \quad | \ i \ c \ S' \ e \ S \end{array}$$

$$\begin{array}{l} S' \rightarrow a \\ \quad | \ i \ c \ S' \ e \ S' \end{array}$$

CONSTRUÇÃO DE UM RECONHECEDOR ASCENDENTE

- Q Como determinar de forma sistemática a ação a realizar (deslocamento, redução, aceitação, rejeição)?
- Q Como o fazer?

CONSTRUÇÃO DE UM RECONHECEDOR ASCENDENTE

pilha	entrada	ação
\$	i v v ; \$	deslocamento
\$ i	v v ; \$	redução por $T \rightarrow i$
\$ T	v v ; \$	deslocamento
\$ T v	v ; \$	rejeição

- A ação a realizar em cada passo do procedimento de reconhecimento — deslocamento, redução, aceitação ou rejeição — depende da configuração em cada momento
- Uma *configuração* é formada pelo conteúdo da pilha mais a parte da entrada ainda não processada
- A pilha é conhecida — na realidade, é preenchida pelo procedimento de reconhecimento
- Da entrada, em cada momento, apenas se conhece o *lookahead*

CONSTRUÇÃO DE UM RECONHECEDOR ASCENDENTE

ilha	entrada	ação
\$	i v v ; \$	deslocamento
\$ i	v v ; \$	redução por $T \rightarrow i$
\$ T	v v ; \$	deslocamento
\$ T v	v ; \$	rejeição

- Quantos símbolos da pilha usar?
- Poder-se-á usar apenas um?
- Se se quiser e puder construir um reconhecedor que apenas use o símbolo no topo, uma pilha onde se guardam os símbolos terminais e não terminais tem pouco interesse
- Mas pode-se definir um alfabeto adequado para a pilha

CONSTRUÇÃO DE UM RECONHECEDOR ASCENDENTE

- Os símbolos a colocar na pilha devem representar estados no processo de deslocamento-redução
- O alfabeto da pilha representa assim o conjunto de estados nesse processo de reconhecimento
- Cada estado representa um conjunto de itens
- Um item representa o quanto de uma produção já foi processado e o quanto ainda falta processar
- Usa-se um ponto (·) sobre a produção para o indicar
- A produção $A \rightarrow B_1 B_2 B_3$ introduz 4 itens:
$$A \rightarrow \cdot B_1 B_2 B_3$$
$$A \rightarrow B_1 \cdot B_2 B_3$$
$$A \rightarrow B_1 B_2 \cdot B_3$$
$$A \rightarrow B_1 B_2 B_3 \cdot$$
- A produção $A \rightarrow \varepsilon$ introduz um único item:
$$A \rightarrow \cdot$$

CONJUNTO DOS CONJUNTOS DE ITENS

- Considere a gramática

$$\begin{aligned} S &\rightarrow E \\ E &\rightarrow a \mid (E) \end{aligned}$$

- Reconhecer a palavra $u = u_1 u_2 \cdots u_n$, significa reduzir $u\$$ a $S\$$
- Então, o estado inicial pode ser definido por

$$Z_0 = \{ S \rightarrow \cdot E \$ \}$$

- O facto de o ponto (\cdot) se encontrar imediatamente à esquerda de um símbolo não terminal, significa que para se avançar no processo de reconhecimento é preciso obter esse símbolo
- Isso é considerado juntando ao conjunto Z_0 os itens iniciais das produções cuja cabeça é E

$$Z_0 = \{ S \rightarrow \cdot E \$ \} \cup \{ E \rightarrow \cdot a, E \rightarrow \cdot (E) \}$$

CONJUNTO DOS CONJUNTOS DE ITENS

$$Z_0 = \{ S \rightarrow \cdot E \$ \} \cup \{ E \rightarrow \cdot a, E \rightarrow \cdot (E) \}$$

- O estado Z_0 pode evoluir por ocorrência de um E , um a ou um $($, que correspondem aos símbolos que aparecem imediatamente à direita do ponto (\cdot)

$$Z_1 = \delta(Z_0, E) = \{ S \rightarrow E \cdot \$ \}$$

$$Z_2 = \delta(Z_0, a) = \{ E \rightarrow a \cdot \}$$

$$Z_3 = \delta(Z_0, () = \{ E \rightarrow (\cdot E) \}$$

- Z_3 tem de ser extendido pela função de fecho, uma vez que o ponto (\cdot) ficou imediatamente à esquerda de um símbolo não terminal (E)

$$Z_3 = \delta(Z_0, () = \{ E \rightarrow (\cdot E) \} \cup \{ E \rightarrow \cdot a, E \rightarrow \cdot (E) \}$$

CONJUNTO DOS CONJUNTOS DE ITENS

- Z_1 representa um situação de aceitação se o símbolo à entrada (*lookahead*) for igual a $\$$ e de erro caso contrário

$$Z_1 = \delta(Z_0, E) = \{ S \rightarrow E \cdot \$ \}$$

- Z_2 representa uma possível situação de redução pela regra $E \rightarrow a$

$$Z_2 = \delta(Z_0, a) = \{ E \rightarrow a \cdot \}$$

- no entanto, só faz sentido fazê-lo se o símbolo à entrada (*lookahead*) for um elemento do conjunto **follow**(E); caso contrário corresponde a uma situação de erro

CONJUNTO DOS CONJUNTOS DE ITENS

- Voltando ao Z_3

$$Z_3 = \delta(Z_0, () = \{ E \rightarrow (\cdot E) \} \cup \{ E \rightarrow \cdot a, E \rightarrow \cdot (E) \}$$

- Z_3 pode evoluir por ocorrência de um E , um a ou um $($

$$Z_4 = \delta(Z_3, E) = \{ E \rightarrow (E \cdot) \}$$

$$\delta(Z_3, a) = Z_2$$

$$\delta(Z_3, () = Z_3$$

- apenas um novo estado foi gerado (Z_4)
- Z_4 apenas evolui por ocorrência de $)$

$$Z_5 = \delta(Z_4,) = \{ E \rightarrow (E) \cdot \}$$

CONJUNTO DOS CONJUNTOS DE ITENS

- Pondo tudo junto

$$Z_0 = \{ S \rightarrow \cdot E \$ \} \cup \{ E \rightarrow \cdot a, E \rightarrow \cdot (E) \}$$

$$Z_1 = \delta(Z_0, E) = \{ S \rightarrow E \cdot \$ \}$$

$$Z_2 = \delta(Z_0, a) = \{ E \rightarrow a \cdot \}$$

$$Z_3 = \delta(Z_0, () = \{ E \rightarrow (\cdot E) \} \cup \{ E \rightarrow \cdot a, E \rightarrow \cdot (E) \}$$

$$Z_4 = \delta(Z_3, E) = \{ E \rightarrow (E \cdot) \}$$

$$Z_5 = \delta(Z_4,) = \{ E \rightarrow (E) \cdot \}$$

•

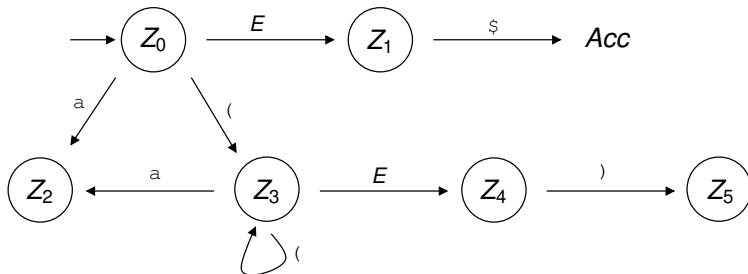


TABELA DE DECISÃO DE UM REC. ASCENDENTE

- A tabela de decisão de um reconhecedor ascendente é uma dupla matriz
 - as linhas são indexadas pelo alfabeto da pilha (conjunto de conjuntos de itens)
 - as colunas são indexadas pelos símbolos terminais (uma das matrizes) e não terminais (a outra) da gramática
- Estas matrizes representam respetivamente as funções ACTION e GOTO
 - ACTION tem como argumentos um estado (símbolo da pilha) e um símbolo terminal (incluindo o \$) e define a ação a realizar (*shift*, *reduce*, *accept* ou *error*)
 - GOTO mapeia um estado e um símbolo não terminal num estado, sendo usada após uma operação de redução

	a	()	\$	E
Z ₀	shift, Z ₂	shift, Z ₃			Z ₁
Z ₁				accept	
Z ₂			reduce, $E \rightarrow a$	reduce, $\bar{E} \rightarrow a$	
Z ₃	shift, Z ₂	shift, Z ₃			Z ₄
Z ₄			shift, Z ₅		
Z ₅			reduce, $E \rightarrow (E)$	reduce, $E \rightarrow (\bar{E})$	

ALGORITMO DE RECONHECIMENTO

- Com base numa tabela deste tipo, o procedimento de reconhecimento pode ser implementado pelo seguinte algoritmo

ALGORITMO

```
push( $Z_0$ )
forever
    if top() ==  $Z_1$  && lookahead() == $
        ACCEPT
    action = table[top(), lookahead()]
    if action is (shift,  $Z_i$ )
        adv(); push( $Z_i$ );
    else if action is (reduce  $A \rightarrow \alpha$ )
        pop  $|\alpha|$  símbolos; push(table[top(),  $A$ ]);
    else
        REJECT
```

RECONHECEDOR ASCENDENTE: EXEMPLO

- Aplicando este algoritmo à palavra $((a))$

pilha	entrada	ação
Z_0	$((a)) \$$	shift Z_3
$Z_0 Z_3$	$(a)) \$$	shift Z_3
$Z_0 Z_3 Z_3$	$a)) \$$	shift Z_2
$Z_0 Z_3 Z_3 Z_2$	$)) \$$	reduce $E \rightarrow a$
$Z_0 Z_3 Z_3$	$)) \$$	goto Z_4
$Z_0 Z_3 Z_3 Z_4$	$)) \$$	shift Z_5
$Z_0 Z_3 Z_3 Z_4 Z_5$	$) \$$	reduce $E \rightarrow (E)$
$Z_0 Z_3$	$) \$$	goto Z_4
$Z_0 Z_3 Z_4$	$) \$$	shift Z_5
$Z_0 Z_3 Z_4 Z_5$	$\$$	reduce $E \rightarrow (E)$
Z_0	$\$$	goto Z_1
$Z_0 Z_1$	$\$$	accept

TABELA DE DECISÃO: EXERCÍCIO

- Q Determine-se a tabela de decisão para um reconhecedor ascendente com *lookahead* 1 da gramática seguinte

$$\begin{aligned} S &\rightarrow a \mid (S) \mid a P \mid (S) S \\ P &\rightarrow (S) \mid (S) S \end{aligned}$$

- O primeiro passo corresponde a alterar a gramática de modo ao símbolo inicial não aparecer do lado direito

$$\begin{aligned} S_0 &\rightarrow S \\ S &\rightarrow a \mid (S) \mid a P \mid (S) S \\ P &\rightarrow (S) \mid (S) S \end{aligned}$$

TABELA DE DECISÃO: EXERCÍCIO

- O passo seguinte corresponde a calcular o conjunto de conjunto de itens

$$Z_0 = \{S_0 \rightarrow \cdot S\} \\ \cup \{S \rightarrow \cdot a, S \rightarrow \cdot (S), S \rightarrow \cdot a P, S \rightarrow \cdot (S) S\}$$

$$Z_1 = \delta(Z_0, S) = \{S_0 \rightarrow S \cdot\}$$

$$Z_2 = \delta(Z_0, a) = \{S \rightarrow a \cdot, S \rightarrow a \cdot P\} \\ \cup \{P \rightarrow \cdot (S), P \rightarrow \cdot (S) S\}$$

$$Z_3 = \delta(Z_0, () = \{S \rightarrow (\cdot S), S \rightarrow (\cdot S) S\} \\ \cup \{S \rightarrow \cdot a, S \rightarrow \cdot (S), S \rightarrow \cdot a P, S \rightarrow \cdot (S) S\}$$

$$Z_4 = \delta(Z_2, P) = \{S \rightarrow a P \cdot\}$$

$$Z_5 = \delta(Z_2, () = \{P \rightarrow (\cdot S), P \rightarrow (\cdot S) S\} \\ \cup \{S \rightarrow \cdot a, S \rightarrow \cdot (S), S \rightarrow \cdot a P, S \rightarrow \cdot (S) S\}$$

$$Z_6 = \delta(Z_3, S) = \{S \rightarrow (S \cdot), S \rightarrow (S \cdot) S\}$$

$$\delta(Z_3, a) = \delta(Z_5, a) = Z_2$$

$$\delta(Z_3, () = \delta(Z_5, () = Z_3$$

TABELA DE DECISÃO: EXERCÍCIO

- continuando

$$Z_7 = \delta(Z_5, S) = \{P \rightarrow (S \cdot), P \rightarrow (S \cdot) S\}$$

$$Z_8 = \delta(Z_6, ()) = \{S \rightarrow (S) \cdot, S \rightarrow (S) \cdot S\} \\ \cup \{S \rightarrow \cdot a, S \rightarrow \cdot (S), S \rightarrow \cdot a P, S \rightarrow \cdot (S) S\}$$

$$Z_9 = \delta(Z_7, ()) = \{P \rightarrow (S) \cdot, P \rightarrow (S) \cdot S\} \\ \cup \{S \rightarrow \cdot a, S \rightarrow \cdot (S), S \rightarrow \cdot a P, S \rightarrow \cdot (S) S\}$$

$$Z_{10} = \delta(Z_8, S) = \{S \rightarrow (S) S \cdot\}$$

$$Z_{11} = \delta(Z_9, S) = \{P \rightarrow (S) S \cdot\}$$

$$\delta(Z_8, a) = \delta(Z_9, a) = Z_2$$

$$\delta(Z_8, ()) = \delta(Z_9, ()) = Z_3$$

TABELA DE DECISÃO: EXERCÍCIO

- Q Determine-se a tabela de decisão para um reconhecedor ascendente com *lookahead* 1 da gramática seguinte

$$S \rightarrow \epsilon \mid a B \mid b A$$

$$A \rightarrow a S \mid b A A$$

$$B \rightarrow a B B \mid b S$$