

Compiladores

Análise sintática (4)
Análise Bottom-Up
Autômatos Empilhar/Reduzir

Plano da aula

- Análise bottom-up
 - AKA:
 - Analisador ascendente
 - Analisador empilhar/reduzir
 - Exemplos
 - Handle
 - Operações básicas
 - Questões a serem tratadas
- Gramática LR
 - Tabelas LR.

Top-Down x Bottom Up

Gramática: $S \rightarrow A B$ String: ccbca
 $A \rightarrow c \mid \epsilon$
 $B \rightarrow cbB \mid ca$

Top-Down		Bottom-Up	
$S \Rightarrow AB$	$S \rightarrow AB$	$ccbca \Leftarrow Acbca$	$A \rightarrow c$
$\Rightarrow cB$	$A \rightarrow c$	$\Leftarrow AcbB$	$B \rightarrow ca$
$\Rightarrow ccbB$	$B \rightarrow cbB$	$\Leftarrow AB$	$B \rightarrow cbB$
$\Rightarrow ccbca$	$B \rightarrow ca$	$\Leftarrow S$	$S \rightarrow AB$

Redução – Exemplo 1

$S \rightarrow aABe$ abbcde
 $A \rightarrow Abc \mid b$
 $B \rightarrow d$

Redução = substituição do lado direito de uma produção pelo não terminal correspondente (lado esquerdo)

Redução – Exemplo 1

$S \rightarrow aABe$ abbcde
 $A \rightarrow Abc \mid b$ aAbcde
 $B \rightarrow d$

Redução – Exemplo 1

$S \rightarrow aABe$ abbcde
 $A \rightarrow Abc \mid b$ aAbcde
 $B \rightarrow d$

handle = seqüência de símbolos do lado direito da produção, tais que suas reduções levam, no final, ao símbolo inicial da gramática

Redução – Exemplo 1

$S \rightarrow aABe$ $abbcde$
 $A \rightarrow Abc \mid b$ $aAbcde$
 $B \rightarrow d$ $aAde$

Redução – Exemplo 1

$S \rightarrow aABe$ $abbcde$
 $A \rightarrow Abc \mid b$ $aAbcde$
 $B \rightarrow d$ $aAde$

Redução – Exemplo 1

$S \rightarrow aABe$ $abbcde$
 $A \rightarrow Abc \mid b$ $aAbcde$
 $B \rightarrow d$ $aAde$
 $aABe$

Redução – Exemplo 1

$S \rightarrow aABe$ $abbcde$
 $A \rightarrow Abc \mid b$ $aAbcde$
 $B \rightarrow d$ $aAde$
 $aABe$
 S

Redução – Exemplo 2

$E \rightarrow E + E$
 $E \rightarrow E * E$ $id + id * id$
 $E \rightarrow (E) \mid id$

Redução – Exemplo 2

$E \rightarrow E + E$
 $E \rightarrow E * E$ $id + id * id$
 $E \rightarrow (E) \mid id$ $E + id * id$

Redução – Exemplo 2

$E \rightarrow E + E$
 $E \rightarrow E * E$
 $E \rightarrow (E) \mid id$

$id + id * id$
 $E + id * id$
 $E + E * id$

13

Redução – Exemplo 2

$E \rightarrow E + E$
 $E \rightarrow E * E$
 $E \rightarrow (E) \mid id$

$id + id * id$
 $E + id * id$
 $E + E * id$
 $E * id$

14

Redução – Exemplo 2

$E \rightarrow E + E$
 $E \rightarrow E * E$
 $E \rightarrow (E) \mid id$

$id + id * id$
 $E + id * id$
 $E + E * id$
 $E * id$
 $E * E$

15

Redução – Exemplo 2

$E \rightarrow E + E$
 $E \rightarrow E * E$
 $E \rightarrow (E) \mid id$

$id + id * id$
 $E + id * id$
 $E + E * id$
 $E * id$
 $E * E$
 E

16

Análise bottom-up Handle

- Considerando uma leitura da esquerda para a direita, a análise bottom-up procura a encontrar o inverso da derivação mais a direita.
 - Deve-se identificar a forma sentencial corrente a ser "reduzida" para sua forma sentencial anterior.
 - Noção de Handle

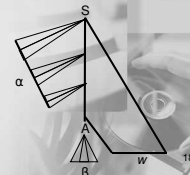
Definição

- Um handle de uma forma sentencial mais a direita γ é uma produção $A \rightarrow \beta$ e uma posição de γ , onde o string β pode ser encontrado e substituído por A para produzir a forma sentencial de γ .
 - Ou seja: se $S \rightarrow^* \alpha A w \rightarrow^* \alpha \beta \delta$, então $A \rightarrow \beta$ na posição seguinte a α

17

Análise bottom-up Handle

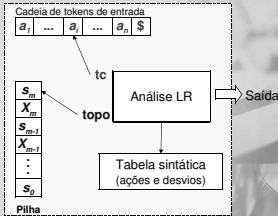
- Handles:
 - É uma subcadeia que reconhece o lado direito de uma produção e cuja redução ao não terminal do lado esquerdo da produção representa um passo ao longo do percurso de uma derivação mais a direita.
 - Exemplo:
 - $\gamma = \alpha \beta w$
 - β é a string parte do handle.
- $S \rightarrow \alpha \beta w$
 $\rightarrow \alpha A w$



18

Analizador bottom-up Estrutura Básica

- A análise Bottom-Up vai necessitar:
 - Uma pilha para guardar os símbolos
 - Um *buffer* de entrada para a sentença w a ser reconhecida.



Ações em Parsing Empilha-Reduz

- Operações lícitas:
 - empilha (shift)**: coloca no topo da pilha o símbolo que está sendo lido e avança o cabeçote de leitura na string
 - reduz (reduce)**: substitui o handle no topo da pilha pelo não terminal correspondente
 - aceita**: reconhece que a sentença foi gerada pela gramática
 - erro**: ocorrendo erro de sintaxe, chama uma subrotina de atendimento a erros

Ações em Parsing Empilha-Reduz Conflitos

Reduce-Reduce

Shift-Reduce

Problemas na análise

Motivados por:

- A gramática é ambígua e a entrada aponta uma situação
- A entrada não é ambígua, mas não é possível determinar a seqüência da análise pela entrada dada.

21

Ações em Parsing Empilha-Reduz Conflitos

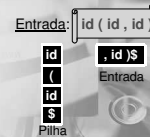
Reduce-Reduce

- Ocorre quando um handle foi obtido, mas o conteúdo da pilha e o próximo símbolo de entrada não permitem identificar a produção a ser utilizada na redução.

Exemplo: sintaxe similar para chamada de procedimentos e acesso arrays

Gramática:

proccall \rightarrow id (parlist)
 parlist \rightarrow parlist , par | par
 par \rightarrow id
 arrayexp \rightarrow id (explist)
 explist \rightarrow explist , exp | exp
 exp \rightarrow id



22

Ações em Parsing Empilha-Reduz Conflitos

Reduce-Reduce

- Ocorre quando um handle foi obtido, mas o conteúdo da pilha e o próximo símbolo de entrada não permitem identificar a produção a ser utilizada na redução.

Exemplo: sintaxe similar para chamada de procedimentos e acesso arrays

Gramática:

proccall \rightarrow id (parlist)
 parlist \rightarrow parlist , proccall | proccall
 proccall \rightarrow id
 arrayexp \rightarrow id (explist)
 explist \rightarrow explist , exp | exp
 exp \rightarrow id



23

Ações em Parsing Empilha-Reduz Conflitos

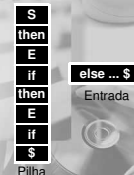
Shift-Reduce

- Não é possível determinar se deve ser executada um shift ou uma redução.

Exemplo: *if-then-else*

Gramática:

$S \rightarrow$ if E then S
 | if E then S else S



24

Ações em Parsing Empilha-Reduz Conflitos

Shift-Reduce

• Não é possível determinar se deve ser executada um shift ou uma redução.

• Exemplo: *if-then-else*

Gramática:

$S \rightarrow \text{if } E \text{ then } S$
 $\quad | \text{if } E \text{ then } S \text{ else } S$

• Resolução do conflito

– Empilhar, associando o else ao if sem else mais interno.

• Note que não é uma solução genérica.



25

Ações em Parsing Empilha-Reduz Exemplo de operação

Gramática:

$E \rightarrow E + E$
 $\quad | E * E$
 $\quad | (E)$
 $\quad | \text{id}$

Entrada:

id + id * id

Pilha	Entrada	Ação
\$	id + id * id	shift
\$id	+ id * id	Reduce $E \rightarrow \text{id}$
\$E	+ id * id	shift
\$E+	id * id	shift
\$E+id	* id	Reduce $E \rightarrow \text{id}$
\$E+E	* id	Reduce $E \rightarrow E + E$
\$E	* id	shift
\$E*	id	shift
\$E*id	\$	Reduce $E \rightarrow \text{id}$
\$E*E	\$	Reduce $E \rightarrow E * E$
\$E	\$	ACEITA

Ações em Parsing Empilha-Reduz Algoritmo

• Reduz

– Se:

- $\gamma\alpha$ está na pilha
- $A \rightarrow \alpha$
- existe um $\beta \in T^*$ tal que $S \Rightarrow^* \gamma A \beta \Rightarrow \gamma \alpha \beta$

– Então:

- Pode-se remover o "handle" α ;
- **Reduz-se** $\gamma \beta$ para γA na pilha

– $\gamma\alpha$ é um prefixo viável

- Ele pode derivar numa seqüência de terminais

• Shift (empilha)

– Empilhar o terminal, avançar entrada

• Erro

• Aceitar

27

Bibliografia

• Livro do dragão, Seção 4.5

– Atenção, na edição em português há um erro. Na página 87, primeira coluna, no parágrafo que começa com "Formalmente, um *handle* de uma forma..." aparece:

$$S \xRightarrow{\text{mad}} \alpha A w \xRightarrow{\text{mad}} \alpha b w$$

mas o correto é:

$$S \xRightarrow{\text{mad}} \alpha A w \xrightarrow{\text{mad}} \alpha \beta w$$

• Price & Toscani, Seção 3.3 (até página 58)

28