

20222 CET058-CMP

Proj2d

Entrega: 31 de outubro

Sumário

- 1. Análise Sintática Ascendente**
- 2. Tabela LR**
- 3. Algoritmo Shift Reduce**
- 4. Sentenças para teste**
- 5. Relatório**

1. Análise Sintática Ascendente

- O objetivo deste projeto é implementar um simulador do algoritmo de Análise Sintática Ascendente Shift-Reduce fornecido no item 3 (Capítulo 4, Seção Seção 4.6.3 “O Algoritmo de Análise Sintática LR” do Livro texto de Aho “Compiladores: princípios, técnicas e ferramentas”) utilizando a tabela LR fornecida no item 2, com as sentenças de entrada listadas no item 4.

2. Tabela LR

State	Action						Goto		
	id	+	*	()	\$	E	T	F
0	S5			S4			1	2	3
1		S6				accept			
2		R2	S7		R2	R2			
3		R4	R4		R4	R4			
4	S5			S4			8	2	3
5		R6	R6		R6	R6			
6	S5			S4				9	3
7	S5			S4					10
8		S6			S11				
9		R1	S7		R1	R1			
10		R3	R3		R3	R3			
11		R5	R5		R5	R5			

FIGURA 4.5

A tabela de análise sintática LR para uma gramática de expressões aritméticas.

3. Algoritmo Shift-Reduce

Faça a o primeiro símbolo de $w\$$;

```
while(1) { /* repetir indefinidamente */  
    fazer que  $s$  seja o estado na parte superior da pilha;  
    if ( ACTION[ $s, a$ ] = Shift  $t$  ) {  
        Push( $a$ ); Push( $t$ ) // empilhar  $at$   
        fazer que  $a$  seja o seguinte símbolo de entrada;  
    } else if ( ACTION[ $s, a$ ] = Reduce  $A \rightarrow \beta$  ) {  
        Pop  $2|\beta|$  símbolos de la pilha;  
        o estado  $t$  agora é o que está no topo da pilha;  
        Push( $A$ )  
        Push( GOTO[ $t, A$ ] );  
        imprimir a produção  $A \rightarrow \beta$ ;  
    } else if ( ACTION[ $s, a$ ] = aceitar ) break; /* terminou */  
    else chamar rotina de recuperação de erros;  
}
```

4. Sentenças para teste

- $(id+id)*id$
- $(id*id)+id$
- $id*(id+id)$
- $((id*id)+id)$

5. Relatório

- O relatório deve conter:
 - Folha de rosto
 - Sumário
 - Link para download da implementação.
 - Comando para compilação
 - Comando para a execução.
 - Saída para testes
 - Referências.