Universidade Estadual de Santa Cruz – UESC

Relatórios de Implementações da Análise Sintática Ascendente

Docente César Alberto Bravo Pariente

Discente Matheus Miranda Brandão

Matrícula 201820065

Disciplina Compiladores.

Curso Ciência da Computação

Semestre 2022.2

Ilhéus – BA
2022

Índice

Análise Sintática Ascendente

O projeto consiste na implementação em C de um algoritmo que busca simular um Analisador Sintático Ascendente Shift-Reduce que reconhece a linguagem gerada por uma gramática livre de contexto. A execução do código recebe como entrada a palavra e retorna o passo a passo da execução consistindo em stack, input, action e ao fim imprime se houve um erro sintático ou foi aceita.

Produções:

p_1: E -> E + T

p_2: E -> T

p_3: T -> T * F

p_4: T -> F

p_5: F -> (E)

p_6: F -> id

Algoritmo a ser implementado:

```
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 siguiente símbolo de entrada;
        } else if ( ACTION[s, a] = Reduce A → β ) {
                Pop 2|β| símbolos de la pilha;
                o estado t ahora é o que está no topo da pilha;
                Push(A)
                Push( GOTO[t, A] );
                imprimir a produção A → β;
        } else if ( ACTION[s, a] = aceitar ) break; /* terminou */
                else chamar rotina de recuperação de erros;
}
```

Tabela LR:

	Action						Goto		
State	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			

Compilando e Executando

Para a execução não é necessário o uso de nenhuma dependência, basta compilar normalmente.

$ gcc proj2_d.c -o proj2_d

Ao executar é necessário digitar o nome do arquivo destino contendo as palavras, caso contrário resultará em erro.

Exemplo:

$ ./proj2_d examples/inputs.txt

Neste projeto pode-se adicionar num .txt todas as palavras separadas por uma quebra de linha. Em caso de erro, pularemos para a próxima palavra.

Exemplo:

```
(id+id)*id
(id*id)+id
(id*(id+id))
((id*id)+id)
```

Testes

Caso a entrada dada seja incorreta o simulador irá imprimir seus itens até o momento, então ao detectar o erro, informará ao usuário e então pulará para a próxima palavra. Como outputs temos o stack, input (palavra) e a ação atual.

Os testes utilizados encontram-se no item 4 do Proj2d.

(id+id)*id

Stack	Input	Action
0	(id+id)*id$	S4
0(4	id+id)*id$	S5
0(4id5	+id)*id$	R6
0(4F3	+id)*id$	R4
0(4T2	+id)*id$	R2
0(4E8	+id)*id$	S6
0(4E8+6	id)*id$	S5
0(4E8+6id5	)*id$	R6
0(4E8+6F3	)*id$	R4
0(4E8+6T9	)*id$	R1
0(4E8	)*id$	S11
0(4E8)11	*id$	R5
0F3	*id$	R4
0T2	*id$	S7
0T2*7	id$	S5
0T2*7id5	$	R6
0T2*7F10	$	R3
0T2	$	R2
0E1	$	Accept

(id*id)+id

Stack	Input	Action
0	(id*id)+id$	S4
0(4	id*id)+id$	S5
0(4id5	*id)+id$	R6
0(4F3	*id)+id$	R4
0(4T2	*id)+id$	S7
0(4T2*7	id)+id$	S5
0(4T2*7id5	)+id$	R6
0(4T2*7F10	)+id$	R3
0(4T2	)+id$	R2
0(4E8	)+id$	S11
0(4E8)11	+id$	R5
0F3	+id$	R4
0T2	+id$	R2
0E1	+id$	S6
0E1+6	id$	S5
0E1+6id5	$	R6
0E1+6F3	$	R4
0E1+6T9	$	R1
0E1	$	Accept

(id*(id+id))

Stack	|Input	|Action
0	|(id*(id+id))$	|S4
0(4	|id*(id+id))$	|S5
0(4id5	|*(id+id))$	|R6
0(4F3	|*(id+id))$	|R4
0(4T2	|*(id+id))$	|S7
0(4T2*7	|(id+id))$	|S4
0(4T2*7(4	|id+id))$	|S5
0(4T2*7(4id5	|+id))$	|R6
0(4T2*7(4F3	|+id))$	|R4
0(4T2*7(4T2	|+id))$	|R2
0(4T2*7(4E8	|+id))$	|S6
0(4T2*7(4E8+6	|id))$	|S5
0(4T2*7(4E8+6id5	|))$	|R6
0(4T2*7(4E8+6F3	|))$	|R4
0(4T2*7(4E8+6T9	|))$	|R1
0(4T2*7(4E8	|))$	|S11
0(4T2*7(4E8)11	|)$	|R5
0(4T2*7F10	|)$	|R3
0(4T2	|)$	|R2
0(4E8	|)$	|S11
0(4E8)11	|$	|R5
0F3	|$	|R4
0T2	|$	|R2
0E1	|$	|Accept

((id*id)+id)

Stack	Input	Action
0	((id*id)+id)$	S4
0(4	(id*id)+id)$	S4
0(4(4	id*id)+id)$	S5
0(4(4id5	*id)+id)$	R6
0(4(4F3	*id)+id)$	R4
0(4(4T2	*id)+id)$	S7
0(4(4T2*7	id)+id)$	S5
0(4(4T2*7id5	)+id)$	R6
0(4(4T2*7F10	)+id)$	R3
0(4(4T2	)+id)$	R2
0(4(4E8	)+id)$	S11
0(4(4E8)11	+id)$	R5
0(4F3	+id)$	R4
0(4T2	+id)$	R2
0(4E8	+id)$	S6
0(4E8+6	id)$	S5
0(4E8+6id5	)$	R6
0(4E8+6F3	)$	R4
0(4E8+6T9	)$	R1
0(4E8	)$	S11
0(4E8)11	$	R5
0F3	$	R4
0T2	$	R2
0E1	$	Accept

Link para download

Código fonte e exemplos encontram-se para download no seguinte link:

https://github.com/MatBrands/Compiladores/tree/master/Proj2/Proj2d