

20222 CET058-CMP

Proj3c

Entrega: 21 de novembro

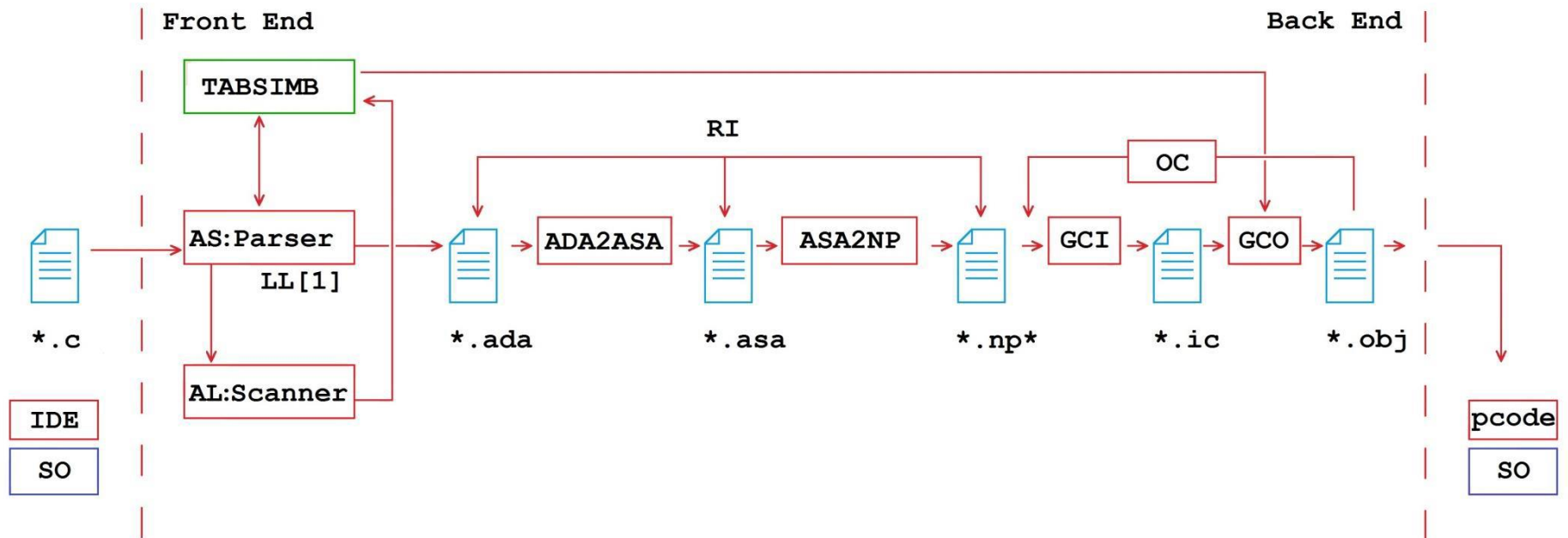
Sumário

- 1. Objetivos**
- 3. Estrutura geral de um compilador**
- 3. Dicionário**
- 4. E/S, CI, CO**
- 5. GLC LL[1] para o projeto**
- 6. Exemplo1: `m(){h=(1+1);r(0);}`**
- 7. Tabela de Análise**
- 8. Árvore de Análise**
- 9. Árvore de Sintaxe Abstrata**
- 10. Representação vetorial**
- 11. Expressão polonesa**
- 12. Pseudo-código**
- 13. Outros programas**
- 14. Relatório**

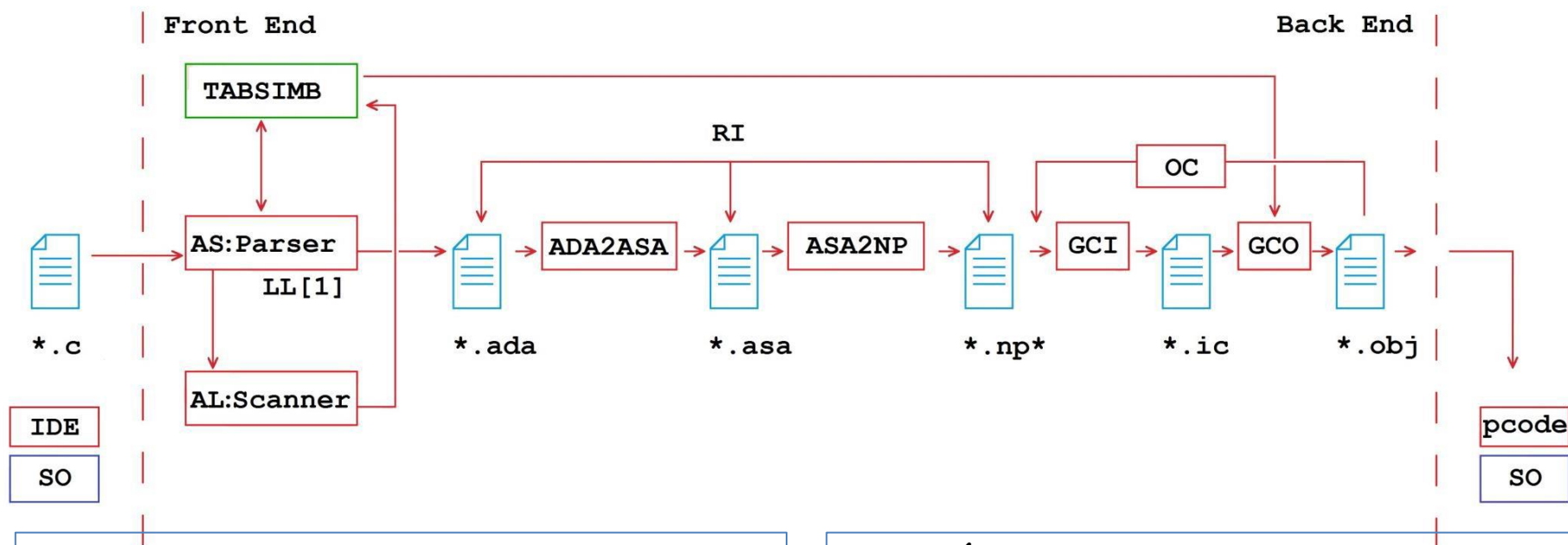
1. Objetivo

Gerar, a partir expressão polonesa e a expressão polonesa reversa (Proj3b) que corresponde à árvore de sintaxe abstrata o pseudocódigo correspondente.

2. Estrutura geral de um compilador



3. Dicionário



4. E/S, CI, PCO

$E \rightarrow E + T \mid T$
 $E \rightarrow T * F \mid F$
 $F \rightarrow a \mid b \mid c \mid (E)$

Front End

Back End

TABSIMB

AS:Parser

LL[1]

AL:Scanner

*.c

IDE

SO

$a+(b*c)$

*.ada

ADA2ASA

RI

*.asa

ASA2NP

.np

GCI

*.ic

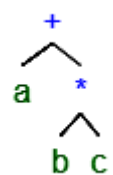
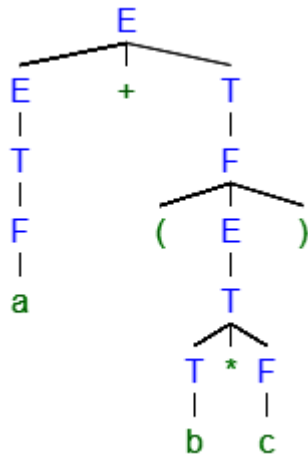
OC

GCO

*.obj

pcode

SO



$abc*+$

LOD a
LOD b
LOD c
MUL
ADD

5. GLC LL[1] para o projeto

- $p_1: S \rightarrow M \mid G M \mid N G M$
- $p_7: M \rightarrow m() \{ A; r(E); \}$
- $p_8: E \rightarrow 0 \mid 1 \mid \dots \mid 9 \mid x \mid y \mid (EXE)$
- $p_{22}: X \rightarrow + \mid - \mid * \mid /$
- $p_{26}: A \rightarrow C \mid ; A$
- $p_{28}: C \rightarrow h=g() \mid i=n() \mid j=E \mid k=E \mid z=E \mid$
 $w(E) \{ A; \} \mid f(E) \{ A; \} \mid o(E; E; E) \{ A; \}$

N G M

- $p_4: \mathbf{N} \rightarrow \mathbf{n}() \{ \mathbf{A}; \mathbf{r}(\mathbf{E}); \}$
- $p_5: \mathbf{G} \rightarrow \mathbf{g}() \{ \mathbf{A}; \mathbf{r}(\mathbf{E}); \}$
- $p_6: \mathbf{M} \rightarrow \mathbf{m}() \{ \mathbf{A}; \mathbf{r}(\mathbf{E}); \}$

6. Exemplo1: `m(){h=(1+1);r(0);}`

```
main () {  
    h = (1+1) ;  
    return (0) ;  
}
```

7. Tabela de Análise

Passo	Estado	Cadeia	Pilha	Transição	Produção
0	0	.m(){h=x;r(0);}	-	t0	-
1	1	.m(){h=x;r(0);}	M	t1	M->m(){C;r(E);}
2	1	.m(){h=x;r(0);}	m(){C;r(E);}	t38	-
3	1	m.(){h=x;r(0);}	() {C;r(E);}	t22	-
4	1	m(.){h=x;r(0);}) {C;r(E);}	t23	-
5	1	m().{h=x;r(0);}	{C;r(E);}	t46	-
6	1	m(){.h=x;r(0);}	C;r(E);}	t13	C->h=E
7	1	m(){.h=x;r(0);}	h=E;r(E);}	t34	-
8	1	m(){h.=x;r(0);}	=E;r(E);}	t31	-
9	1	m(){h=.x;r(0);}	E;r(E);}	t6	E->x
10	1	m(){h=.x;r(0);}	x;r(E);}	t43	-
11	1	m(){h=x.;r(0);}	;r(E);}	t30	-
12	1	m(){h=x;.r(0);}	r(E);}	t41	-
13	1	m(){h=x;r.(0);}	(E);}	t22	-
14	1	m(){h=x;r(.0);}	E);}	t4	E->0
15	1	m(){h=x;r(.0);}	0);}	t28	-
16	1	m(){h=x;r(0.);});}	t23	-
17	1	m(){h=x;r(0).;}	;}	t30	-
18	1	m(){h=x;r(0);.}	}	t47	-
19	1	m(){h=x;r(0);}.	-	-	-

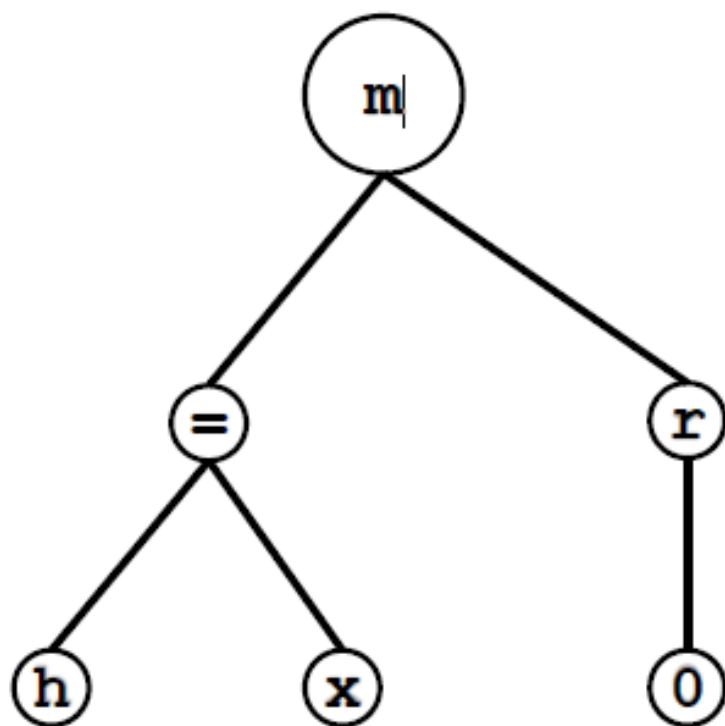
8. Árvore de Análise

Passo	Estado	Cadeia	Transicao	Producao
0	0	.m() {h=x;r(0);}	t0	-
1	1	.m() {h=x;r(0);}	t1	M->m() {C;r(E);}
2	1	.m() {h=x;r(0);}	t38	-
3	1	m.() {h=x;r(0);}	t22	-
4	1	m.() {h=x;r(0);}	t23	-
5	1	m.() {h=x;r(0);}	t46	-
6	1	m.() {h=x;r(0);}	t15	>h=E
7	1	m() {h=x;r(0);}	t34	-
8	1	m() {h=x;r(0);}	t31	-
9	1	m() {h=x;r(0);}	t6	E->x
10	1	m() {h=x;r(0);}	t43	-
11	1	m() {h=x;r(0);}	t30	-
12	1	m() {h=x;r(0);}	t41	-
13	1	m() {h=x;r(0);}	t22	-
14	1	m() {h=x;r(0);}	t4	E->0
15	1	m() {h=x;r(0);}	t28	-
16	1	m() {h=x;r(0);}	t23	-
17	1	m() {h=x;r(0);}	t30	-
18	1	m() {h=x;r(0);}	t47	-
19	1	m() {h=x;r(0);}	-	-

```

graph TD
    M((M)) --> m((m))
    M --> LP1(( ))
    M --> RP1(( ))
    M --> LFP1(( ))
    M --> C((C))
    M --> S1((;))
    M --> r((r))
    M --> LP2(( ))
    M --> E((E))
    M --> RP2(( ))
    M --> S2((;))
    M --> RP3(( ))
    C --> h((h))
    C --> EQ((=))
    C --> E2((E))
    E2 --> 0((0))
    E2 --> x((x))
  
```

9. Árvore de Sintaxe Abstrata



10. Representação vetorial

i	ADP[i]
0	S
1	m
2	(
3)
4	{
5	C
6	;
7	r
8	(
9	E
10)
11	;
12	}

i	ASA[i]
0	m
1	=
2	r
3)
4	h
5	x
6	0

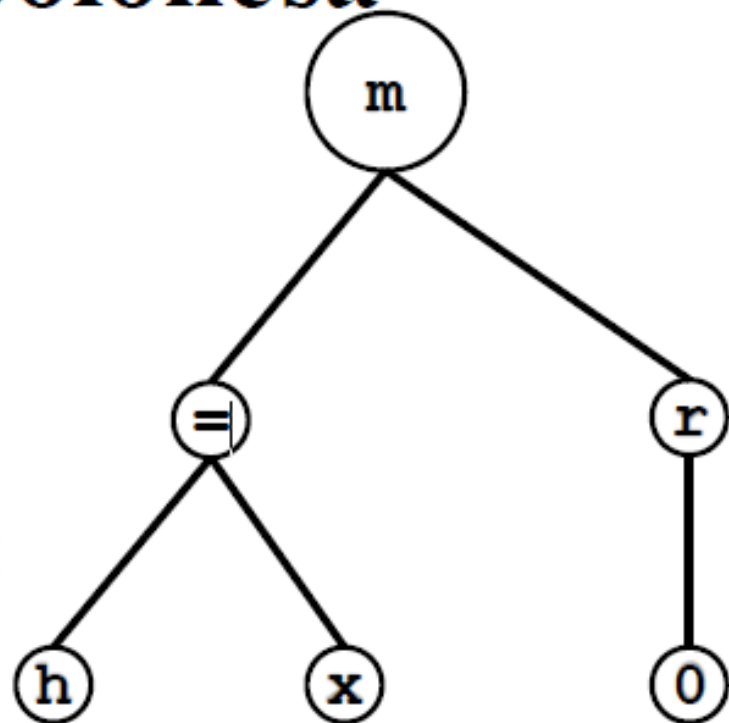
11. Expressão polonesa

- Expressão polonesa:

$$m = hxr0$$

- Expressão polonesa reversa:

$$hx = 0rm$$



12. Pseudo-código

- Expressão polonesa: mhxr0

- M

INT 0 6 // reserva memoria p/h x 0

LOD 0 4 // carrega o valor de x

STO 0 3 // h recebe o valor de x

LIT 0 0 // carrega 0

STO 0 5 // guarda 0 na pos 5

- EM

13. Outros programas

1.m(){h=g();w(x){k=(1+(1*0))};r(y);}

2.m(){w(x){k=(1+(1*0));};r(n());}

3.m(){w(x){f(1){i=n();x=(0+1)};r(y);}

4.m(){w(x){f(1){w((x+y)){k=g();}};};r(y);}

5.n(){w(1){x=0};r(1);}m(){h=n();i=g();r(0);}

14. 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.