

20222 CET058-CMP

Proj3d

Entrega: 01 de dezembro

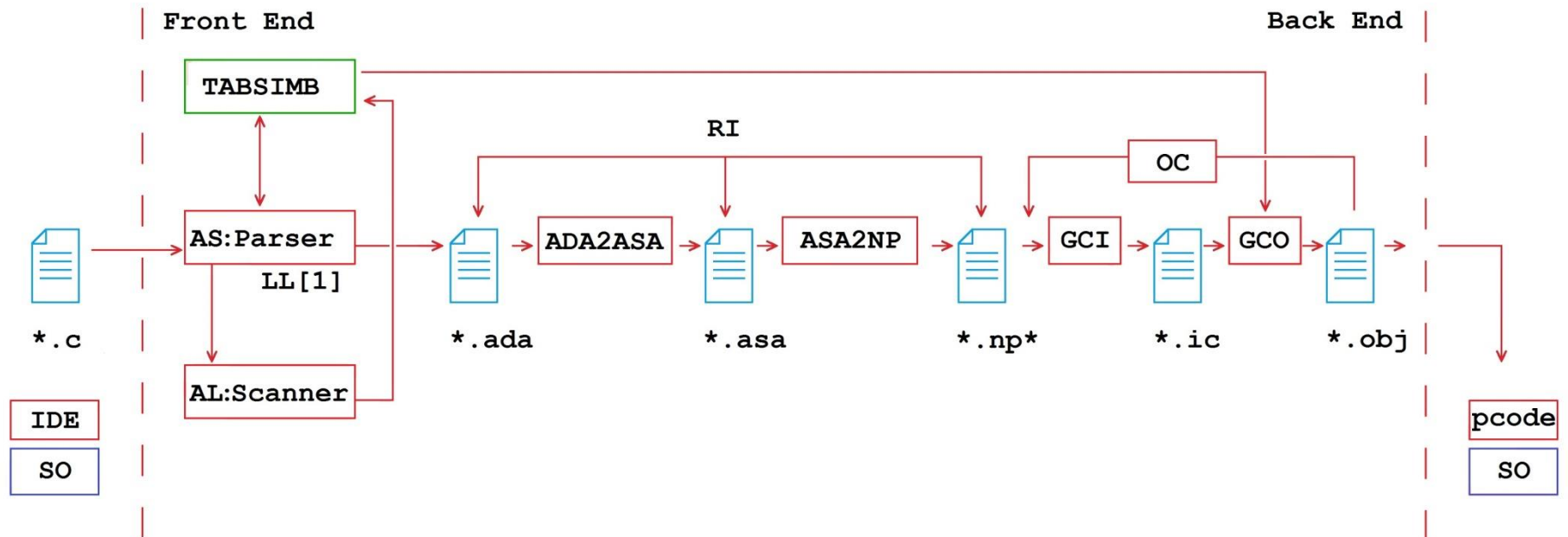
Sumário

- 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. Outros programas**
- 13. Outra gramática**
- 14. Gerador de Analisadores Sintáticos**
- 15. Relatório**

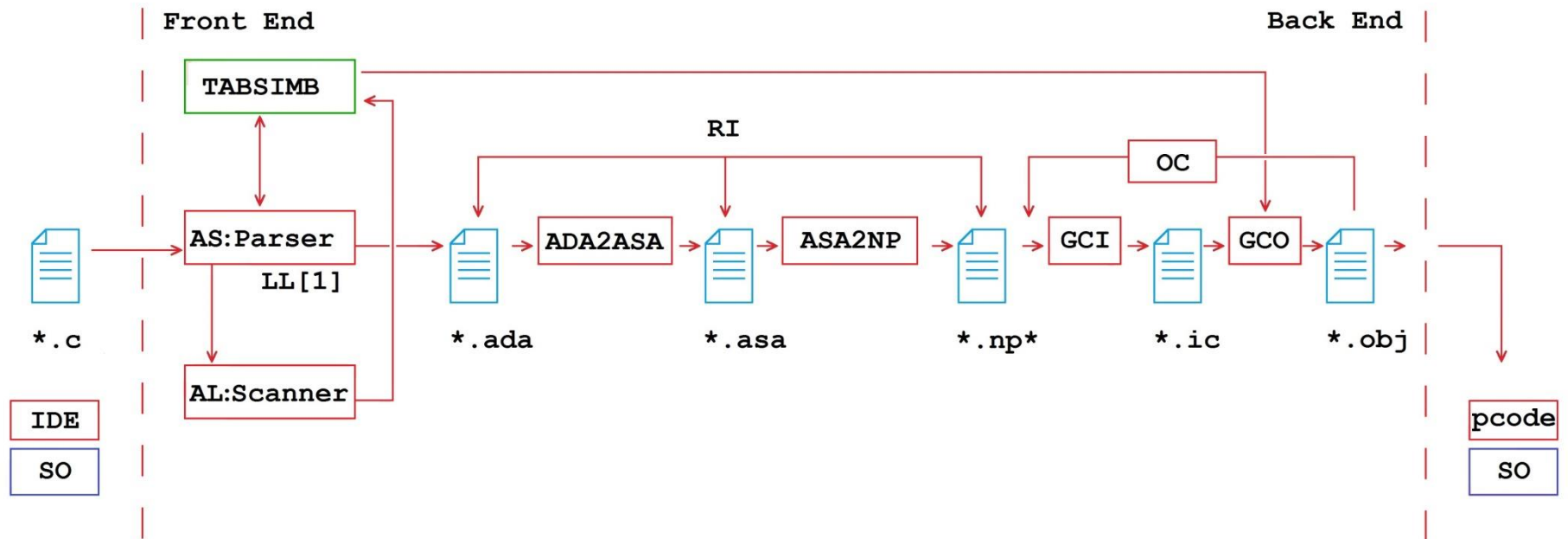
1. Objetivos

- Gerar, a partir do pseudo-código (Projeto 3c) o código p-code, correspondente, à expressão polonesa e a expressão polonesa reversa que corresponde a essa árvore de sintaxe abstrata.

2. Estrutura geral de um compilador



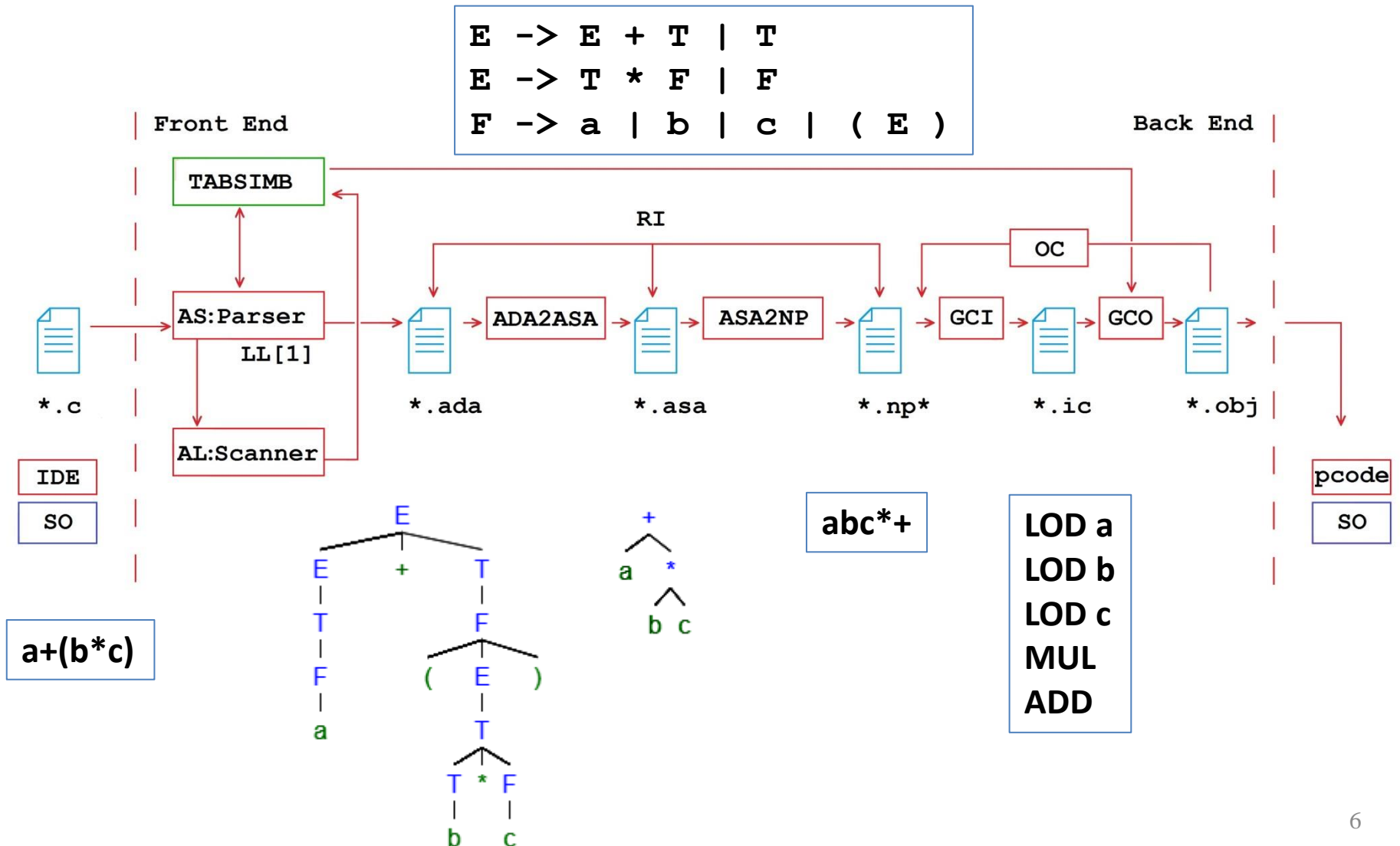
3. Dicionário



IDE: Ambiente Desenvolvimento Integrado
SO: Sistema operacional
TABSIMB: Tabela de Símbolos
AS: Analisador Sintático
AL: Analisador Léxico

ADA: Árvore de Análise
ASA: Árvore de Sintaxe Abstrata
GCI: Gerador de Código Intermediário
GCO: Gerador de Código Objeto
Pcode: máquina virtual objeto

4. E/S, CI, CO



Sequencia de comandos

$\{ C; C; C; C.; r(E); \}$

- $S \rightarrow \{ A; r(E); \}$
- $A \rightarrow CB$
- $B \rightarrow . \mid ;CB$

$\{ C; C; C; C; C; C; C \}$

- $D \rightarrow \{ CB$
- $B \rightarrow \} \mid ;CB$

5. GLC LL[1] para o projeto

- $p_1: S \rightarrow M \mid G M \mid N G M$
- $p_6: M \rightarrow m() \{ A; r(E); \}$
- $p_7: A \rightarrow CB$
- $p_8: B \rightarrow . \mid ;CB$
- $p_{10}: E \rightarrow 0 \mid 1 \dots \mid 9 \mid x \mid y \mid (EXE)$
- $p_{24}: X \rightarrow + \mid - \mid * \mid /$
- $p_{28}: C \rightarrow h=g() \mid i=n() \mid j=E \mid k=E \mid$
 $z=E \mid (EXE) \mid w(E) \{CD \mid f(E) \{ CD \mid o(E;$
 $E; E) \{CD$
- $p_{37}: D \rightarrow \} \mid ;CD$

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) ;  
}
```

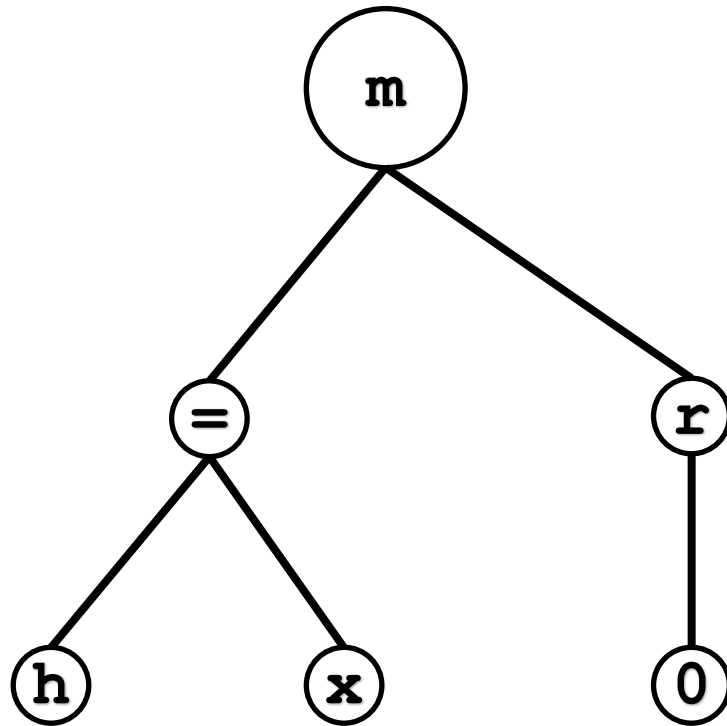
7. Tabela de Análise

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

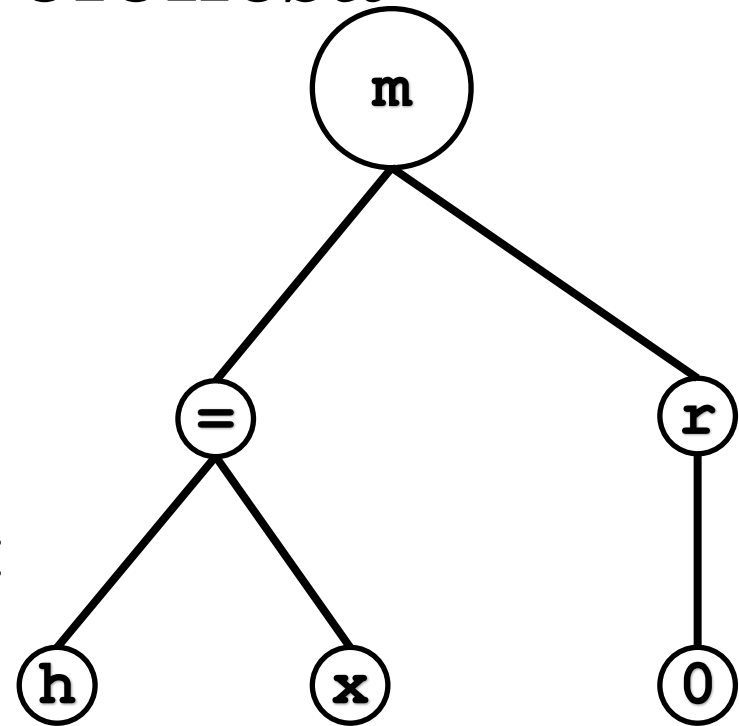
11. Expressão polonesa

- 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 Semântica de **while**, **if**, **for**

- **while**: enquanto a condição for verdadeira, executar o corpo do **while**.
- **if**: executar o corpo do **if**, só quando a condição for verdadeira.
- **for**: executar atribuição E1; se E2 avalia para verdade executar o corpo do **for**; executar atribuição E3; voltar a avaliar E;

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.