

Tradução dirigida por sintaxe

- Tradução/geração código
em conjunto com A.S.
- regras de produção
- ações semânticas
- atributos/variáveis
 - Terminais
 - não Terminais

$$\begin{aligned} D &::= \text{var} : T && \{ \text{addTS}(\text{var.name}, T.\text{tipo}) \} \\ T &::= \text{real} && \{ T.\text{tipo} = 'r' \} \\ T &::= \text{int} && \{ T.\text{tipo} = 'i' \} \end{aligned}$$

Esquemas de Tradução

- conjunto de ações semânticas
- Atributos
 - herdados
 - sintetizados
- passos ES:
 - análise produções
 - árvore de derivação
 - grafo dependência

Exemplo

mat infixa \rightarrow posfixa

$E ::= TR$

$R ::= Op \ T \ \{pT(Op, Sb)\} \ R \ | \ \epsilon$

$T ::= num \ \{pT(num, lexical)\}$

7 - 5 + 3

Esquemas de Tradução

- conjunto de ações semânticas
- Atributos
 - herdados
 - sintetizados
- passos ET:
 - análise produções
 - árvore de derivação
 - grafo de dependência

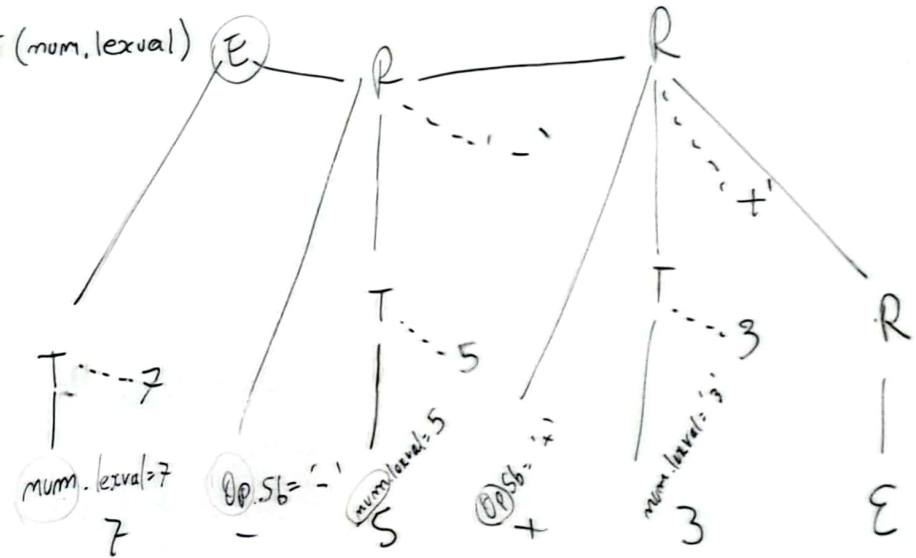
Exemplo

mat infixa \rightarrow pós fixa

$E ::= TR$

$R ::= Op\ T\ \{prT(Op.Sb)\}\ R\ |\ \epsilon$

$T ::= num\ \{prT(num.lexval)\}$



Esquemas de Tradução

- conjunto de ações semânticas

- Atributos

 - herdados

 - sintetizados

- passos ET:

 - análise produções

 - árvore de derivação

 - grafo de dependência

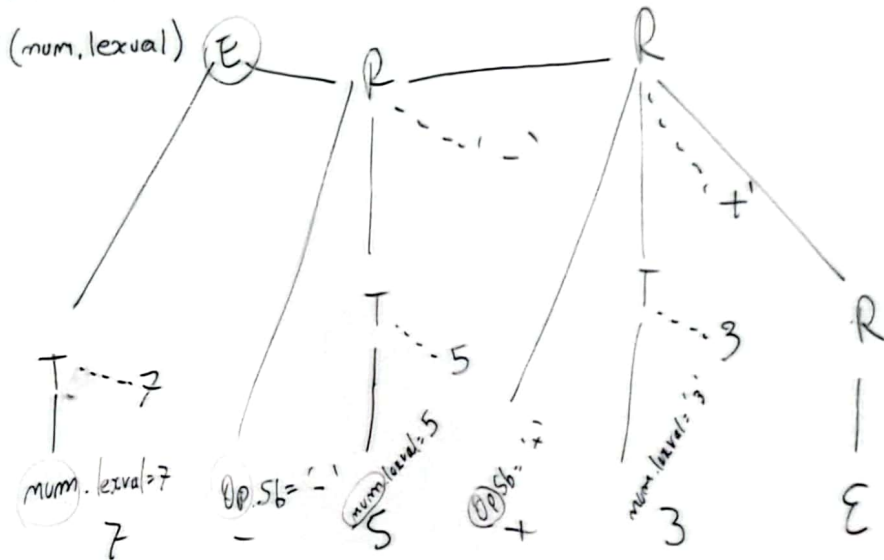
Exemplo

mat infixa \rightarrow pós fixa

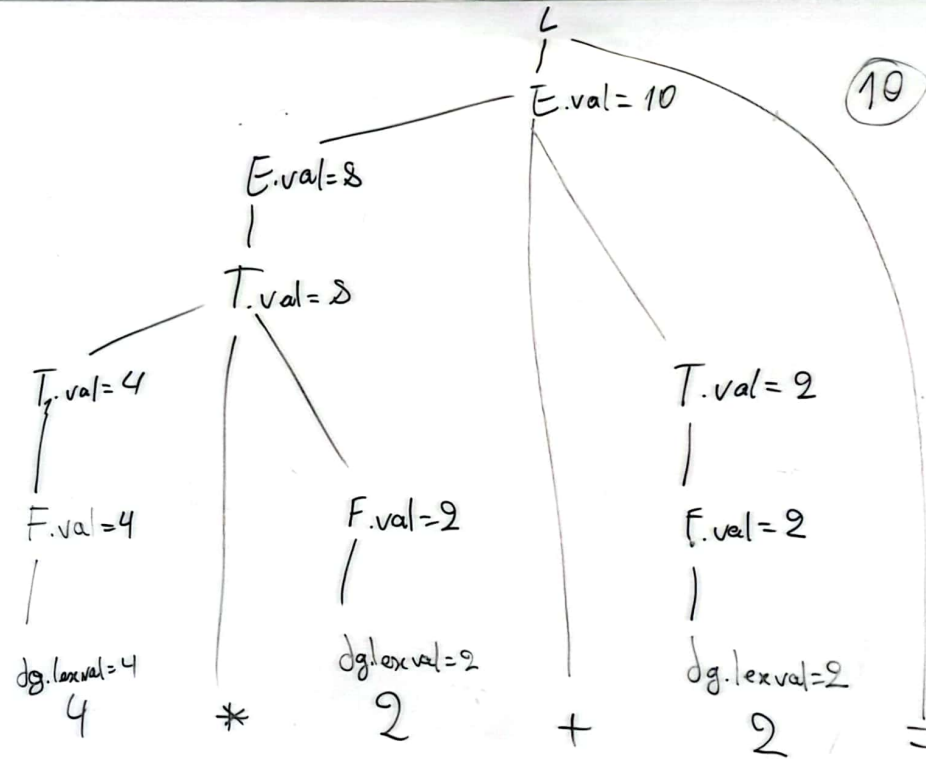
$E ::= TR$

$R ::= Op\ T\ \{pT(Op.Sb)\}\ R\ |\ \epsilon$

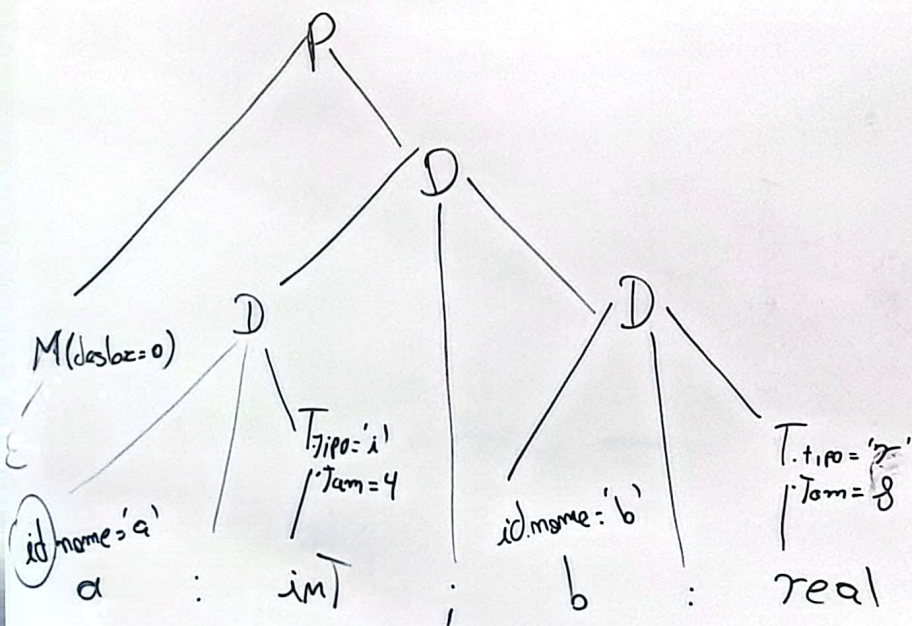
$T ::= num\ \{pT(num.lexval)\}$



$L ::= E = \{ \text{prf}(E.\text{val}) \}$
 $E ::= E_1 + T \{ E.\text{val} = E_1.\text{val} + T.\text{val} \}$
 $E ::= T \{ E.\text{val} = T.\text{val} \}$
 $T ::= T_1 * F \{ T.\text{val} = T_1.\text{val} * F.\text{val} \}$
 $T ::= F \{ T.\text{val} = F.\text{val} \}$
 $F ::= (E) \{ F.\text{val} = E.\text{val} \}$
 $F ::= dg \{ F.\text{val} = dg.\text{lexval} \}$



Ações semânticas para construção de TS



$P ::= M D$

$M ::= E$

$\{desloc = 0\}$

$D ::= D ; D$

$D ::= id : T$

$\{addSb(id.nome, T.tipo, desloc, dc); desloc = desloc + T.tam\}$

$T ::= int$

$\{T.tipo = i ; T.tam = 4\}$

$T ::= real$

$\{T.tipo = r ; T.tam = 8\}$

$T ::= array[num] of T_1$

$\{T.tipo = matriz(num.val, T_1.tipo); T.tam = num.val * T_1.tam\}$

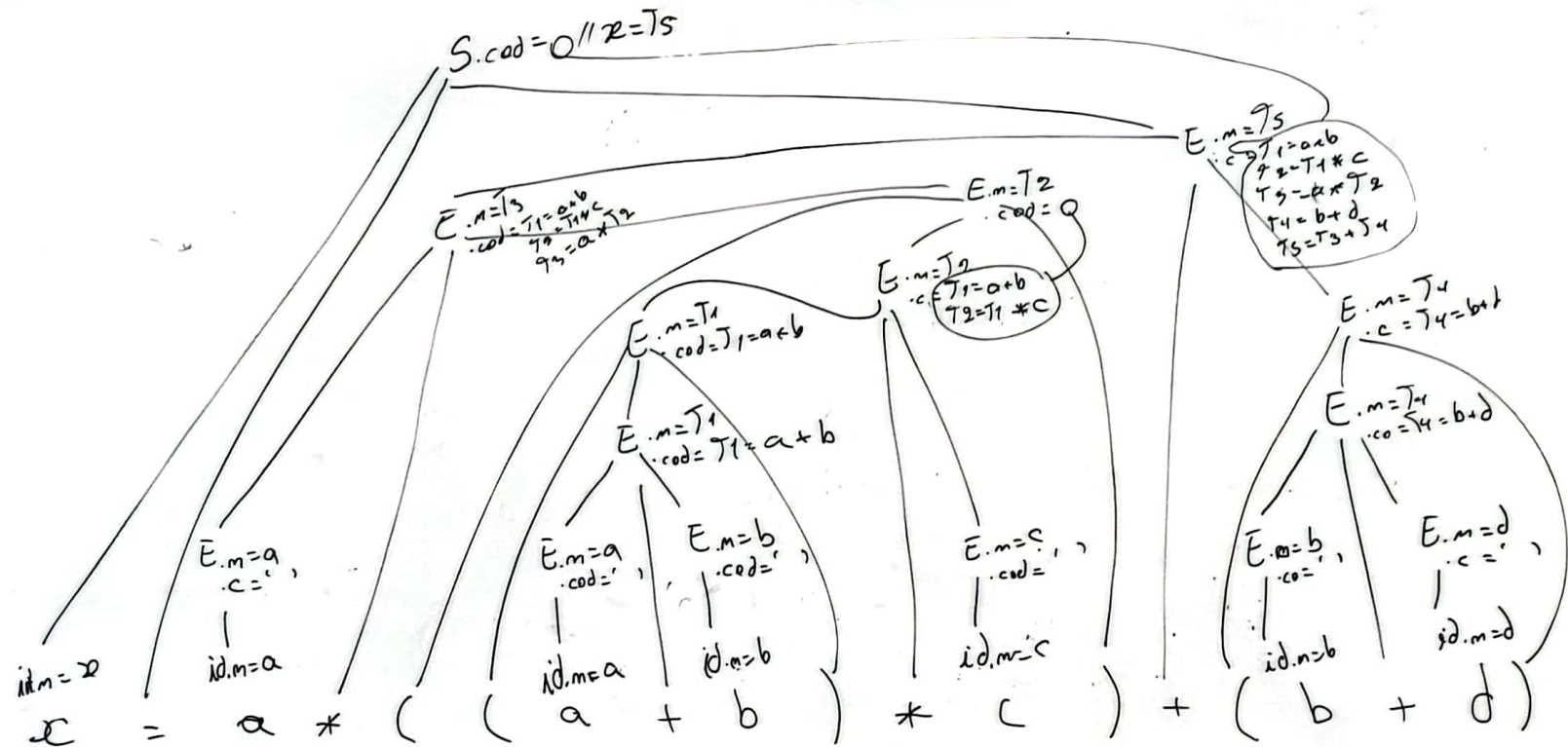
$T ::= ^T_1$

$\{T.tipo = ponteiro(T_1.tipo); T.tam = 4\}$

desloc = 0 4 12

T			I
a	i	0	
b	r	4	

a : array [mem] of int ; b : ^ real



Código de 3 endereços

$A ::= B \text{ op } C$

$A ::= \text{op } B$

$A ::= B$

goto L

if A op1 B goto L

$S ::= id = E \quad \{ S.cod = E.cod // \text{geracod}(id.mnome := E.mnome) \}$

$E ::= E_1 + E_2 \quad \{ E.mnome = \text{geraTemp}; E.cod = E_1.cod // E_2.cod // \\ \text{geracod}(E.mnome = 'E_1.mnome' + 'E_2.mnome') \}$

$E ::= E_1 * E_2 \quad \{ E.mnome = \text{geraTemp}; E.cod = E_1.cod // E_2.cod // \\ \text{geracod}(E.mnome = 'E_1.mnome' * 'E_2.mnome') \}$

$E ::= (E_1) \quad \{ E.mnome = E_1.mnome; E.cod = E_1.cod \}$

$E ::= id \quad \{ E.mnome = id.mnome; E.cod = ' ' \}$

$x = a * ((a + b) * c) + (b + d)$

Otimização código por GAD

- Passos construção GAD

1 - Se y não está no GAD

criar folha y

Se existir operando x

criar folha x

2 - Se existe 'op' com filhos y e x

criar nó de op⁽²⁾ com arestas

para y e x .

$$y = ((a+b) * (a-b)) + ((a+b) * (a-c))$$

$$T_1 = a + b$$

$$T_2 = a - b$$

$$T_3 = T_1 * T_2$$

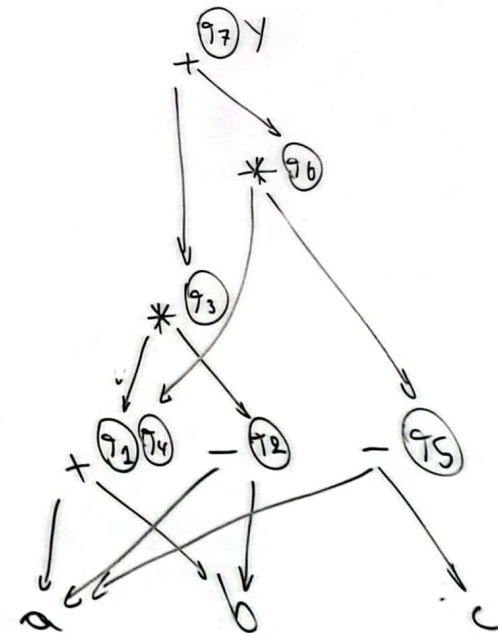
$$T_4 = a + b$$

$$T_5 = a - c$$

$$T_6 = T_4 * T_5$$

$$T_7 = T_3 + T_6$$

$$y = T_7$$



$$T_1 = a + b$$

$$T_2 = a - b$$

$$T_3 = T_1 * T_2$$

$$T_4 = a - c$$

$$T_5 = b - c$$

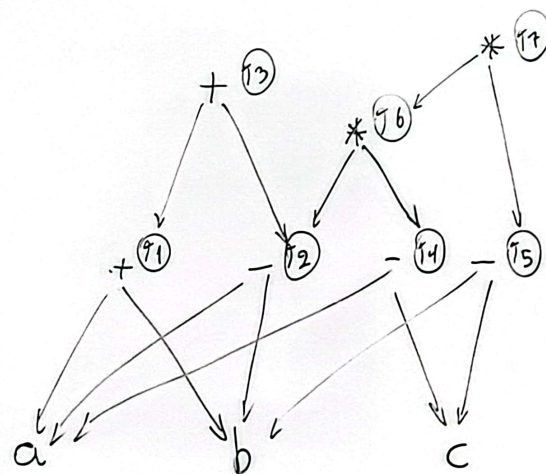
$$T_6 = T_2 * T_4$$

$$T_7 = T_6 * T_5$$

```

LOAD a
ADD b
STR T1
LOAD a
SUB b
STR T2
LOAD T1
MULT T2
STR T3
LOAD a
SUB c
STR T4
LOAD b
SUB c
STR T5
LOAD T6
MULT T4
MULT T5
STR T7

```



$$T_1 = a + b$$

$$T_2 = a - b$$

$$T_3 = T_1 * T_2$$

$$T_4 = a - c$$

$$T_5 = b - c$$

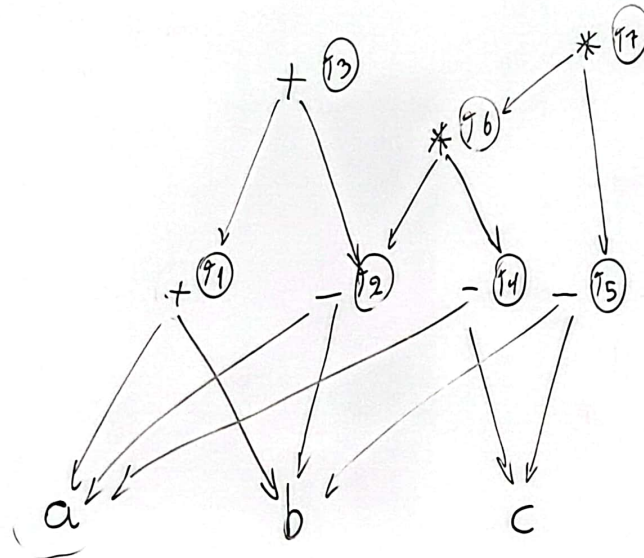
$$T_6 = T_2 * T_4$$

$$T_7 = T_6 * T_5$$

```

LOAD a
ADD b
STR T1
LOAD a
SUB b
STR T2
LOAD T1
MULT T2
STR T3
LOAD a
SUB c
STR T4
LOAD b
SUB c
STR T5
LOAD T6
MULT T4
MULT T5
STR T7

```



$$L = \{T_3, T_1, T_7, T_6, T_2, T_4, T_5\}$$

Optimização

1- Faça $L = \emptyset$

2- Escolha m do GAD que \notin em L
 de arestas que incidam em m
 se originam em nodes que $\in L$,
 então add m em L
 ir p 3

3- Se m é o último node add em L e
 aresta esq de m incide em node n que $\in L$ e
 predecessores de m estão em L então
 add m em L
 Senão $\rightarrow 2$

$$T_5 = b - c$$

$$T_4 = a - c$$

$$T_2 = a - b$$

$$T_6 = T_2 * T_4$$

$$T_7 = T_6 * T_5$$

$$T_1 = a + b$$

$$T_3 = T_1 * T_2$$

```

LOAD b
SUB c
STR T5
LOAD a
SUB c
STR T4
LOAD a
SUB b
MUL T4
MUL T5
STR T7
LOAD a
ADD b
MUL T2
STR T3
    
```