Tradução Dirigida Pela Sintaxe

Julho 2006

Sugestão de leitura: Livro do Aho, Sethi, Ullman (dragão)- Seções 5.1-5.5

Tradução dirigida pela sintaxe

- É uma técnica que permite realizar tradução (geração de código) concomitantemente com a análise sintática
- Ações semânticas são associadas às regras de produção da gramática de modo que, quando uma dada produção é processada, essas ações sejam executadas

Tradução dirigida pela Sintaxe

- A execução das ações semânticas pode:
  - gerar código intermediário
  - gerar ou interpretar código
  - armazenar informações na tabela de símbolos
  - checar a semântica dos comandos
  - Emitir mensagens de erro, etc.

Tradução dirigida pela sintaxe

- Para tornar as ações semânticas mais efetivas, pode-se associar variáveis aos símbolos (terminais e não terminais) da gramática
- Assim, os símbolos gramaticais passam a conter atributos (ou parâmetros) capazes de armazenar valores durante o processo de reconhecimento

Tradução dirigida pela sintaxe

- Um atributo é qualquer propriedade de uma construção de linguagem de programação.
- O processo de computar um atributo e associar o seu valor computado com a construção da linguagem em questão recebe o nome de **amarração do atributo**.

Tradução dirigida pela sintaxe

- Na tradução dirigida pela sintaxe, os atributos são diretamente associados aos símbolos gramaticais da linguagem
- Se X for um símbolo gramatical (terminal ou não terminal), e *a* for um atributo associado a X, escrevemos X.*a* para o valor de a associado a X.

Tradução dirigida pela sintaxe

Exemplo:

D → **var id** : *type* { addTAbSimb(id.lexeme, type.tipo}

type → **id** { type.tipo = buscaTS(id.lexeme);
if (type.tipo não for identificador de tipo)
erro("Tipo indefinido na linha "+linha);
}

- Toda vez que uma regra de produção é usada no processo de reconhecimento de uma sentença, os símbolos gramaticais são alocados junto com seus atributos

Tradução dirigida pela sintaxe

- Pensando na árvore de derivação da sentença sob análise, é como se a cada nó da árvore (símbolo gramatical) correspondesse a uma instanciação de um símbolo e seus atributos
- É como se o nó contivesse campos para armazenar valores correspondentes ao símbolo

Tradução dirigida pela sintaxe

Lexeme é um atributo do átomo

D → **var id** : *type* { addTAbSimb(id.lexeme, type.tipo }

type → **id** { type.tipo = buscaTS(id.lexeme);

if (type.tipo não for identificador de tipo)

erro("Tipo indefinido na linha "+linha);

}

O símbolo gramatical type tem um atributo tipo que indica qual o tipo encontrado

Tradução dirigida pela sintaxe

1) D → var id : type { addTAbSimb(id.lexeme, type.tipo}
2) type -> id { type.tipo = buscaTS(id.lexeme);
if (type.tipo não for identificador de tipo)
erro(“Tipo indefinido na linha ”+linha)
}

- Durante o processo de reconhecimento, quando a produção 2 é reduzida, o atributo tipo associado ao não terminal *type* recebe o valor do tipo reconhecido

Tradução dirigida pela sintaxe

1) D → var id : type { addTAbSimb(id.lexeme, type.tipo}
2) type -> id { type.tipo = buscaTS(id.lexeme);
if (type.tipo não for identificador de tipo)
erro(“Tipo indefinido na linha ”+linha)
}

- A redução da produção 1 adiciona na tabela de símbolos o nome da variável e seu respectivo tipo

Tradução dirigida pela sintaxe

- Sabemos que em nosso modelo sendo implementado, cada símbolo não terminal foi convertido em uma função
- Na maioria das linguagens não é possível associar atributos à nome de funções
- No entanto, esses atributos podem ser representados por parâmetros passados à função, ou mesmo um valor de retorno

Tradução dirigida pela sintaxe

- Por exemplo, na gramática:

```
D → var id : type   { addTAbSimb( id.lexeme, type.tipo}
type -> id          { type.tipo = buscaTS(id.lexeme);
                      if ( type.tipo não for identificador de tipo)
                          erro(“Tipo indefinido na linha ”+linha)
                    }
```

- Temos dois símbolos não terminais
- Logo, necessitamos de duas funções para implementar um analisador sintático descendente recursivo

Tradução dirigida pela sitaxe

```
void D (void)
{
    match(VAR);
    IDVar = atomoCorrente;
    match(ID);
    match(COLON);
    type( &tipo);
    addTabSimb(IDVar.lexeme, tipo);
}
```

Tradução dirigida pela sintaxe

```
void type( ttype *t)
{
    atomoIDTipo = atomoCorrente;
    match(ID);
    ptrID = buscaTS(atomoIDTipo.lexeme);

    if ( !prtID)
        erro("Identificador de tipo" +atomoIDTipo.lexeme + "indefinido");

    if (ptrID.categoria  != IDTIPO)
        erro("Identificador " +atomoIDTipo.lexeme + " não é identificador
                                                    de tipo");

    t = ptrID.tipo;
}
```

Exercício

- Quais seriam as ações semânticas para o comando:

declaracao_de_variaveis ::=

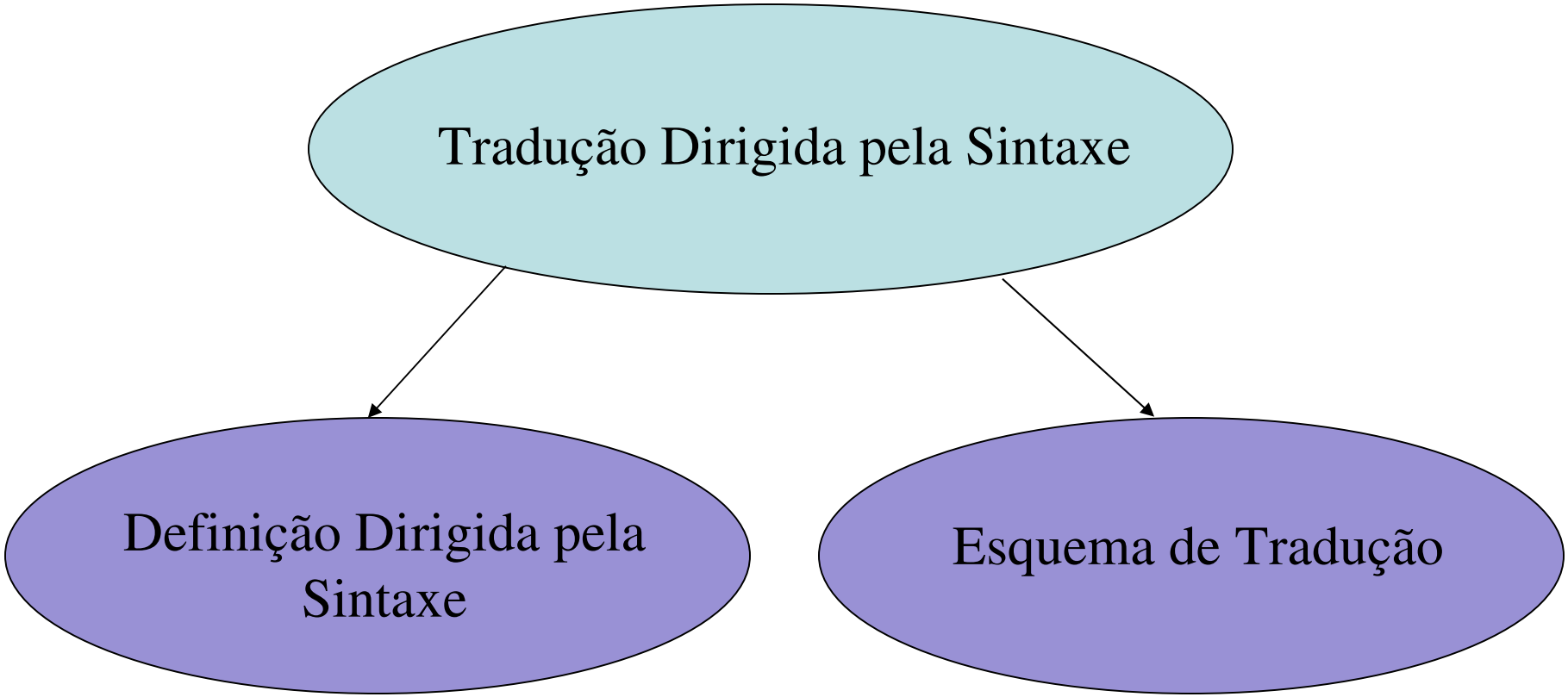
declare *lista_de_ID*$_1$ **:** *tipo* { **;** *lista_de_ID*$_i$ **:** *tipo* }**;**

lista_de_ID ::= **identificador** { **,** **identificador** }

Tradução Dirigida pela Sintaxe

- Trata da compilação de linguagens guiadas por gramáticas livres de contexto.
- Associa informações atrelando atributos aos símbolos gramaticais que representam as construções.
- Os valores dos atributos são computados pelas ações semânticas.
- Existem duas notações:
  - Definição dirigida pela sintaxe
  - Esquema de tradução

Tradução Dirigida pela Sintaxe



Tradução de Linguagens

- O fluxo de *tokens* é sintaticamente analisado
- É construída a árvore gramatical
- A árvore é percorrida avaliando-se as **regras semânticas**
- Regras semânticas podem:
  - Gerar código
  - Salvar informações na **TABELA DE SÍMBOLOS**
  - Emitir mensagens de erro
  - Realizar outras atividades
- Pode ser de 1 ou mais passagens
- Pode ser implementada em uma passagem, executando as ações semânticas juntamente com a análise sintática.

Definição dirigida pela sintaxe

- Especificações de alto nível
- Cada símbolo gramatical possui um conjunto de atributos associados divididos em:
  - **Atributos sintetizados**: valor calculado em função dos filhos
  - **Atributos herdados**: valor calculado em função dos pais e irmãos

Definição dirigida pela sintaxe

- Regras semânticas estabelecem dependências entre os atributos representados em um grafo
- O grafo determina a ordem de avaliação das regras
- A avaliação das regras semânticas gera os valores dos atributos
- A árvore gramatical que mostra os valores dos atributos é chamada de **árvore gramatical anotada**

Definição dirigida pela sintaxe

- Cada produção $A \rightarrow \alpha$ está associado a um conjunto de regras semânticas na forma:

$$B := f(c_1, c_2, \ldots, c_k)$$

Definição dirigida pela sintaxe

- Onde ***f*** é uma função que vigora em *uma das seguintes situações*:
  - *B* é um **atributo sintetizado** de A e c_1, c_2, ..., c_k são atributos pertencentes aos símbolos gramaticais da produção
  - *B* é um **atributo herdado**, pertencente a um dos símbolos gramaticais do lado direito da produção e c_1, c_2, ..., c_k são atributos pertencentes aos símbolos gramaticais da produção
- Em ambos os casos, dizemos que o atributo B depende dos atributos c_1, c_2, ..., c_k

Definição dirigida pela sintaxe

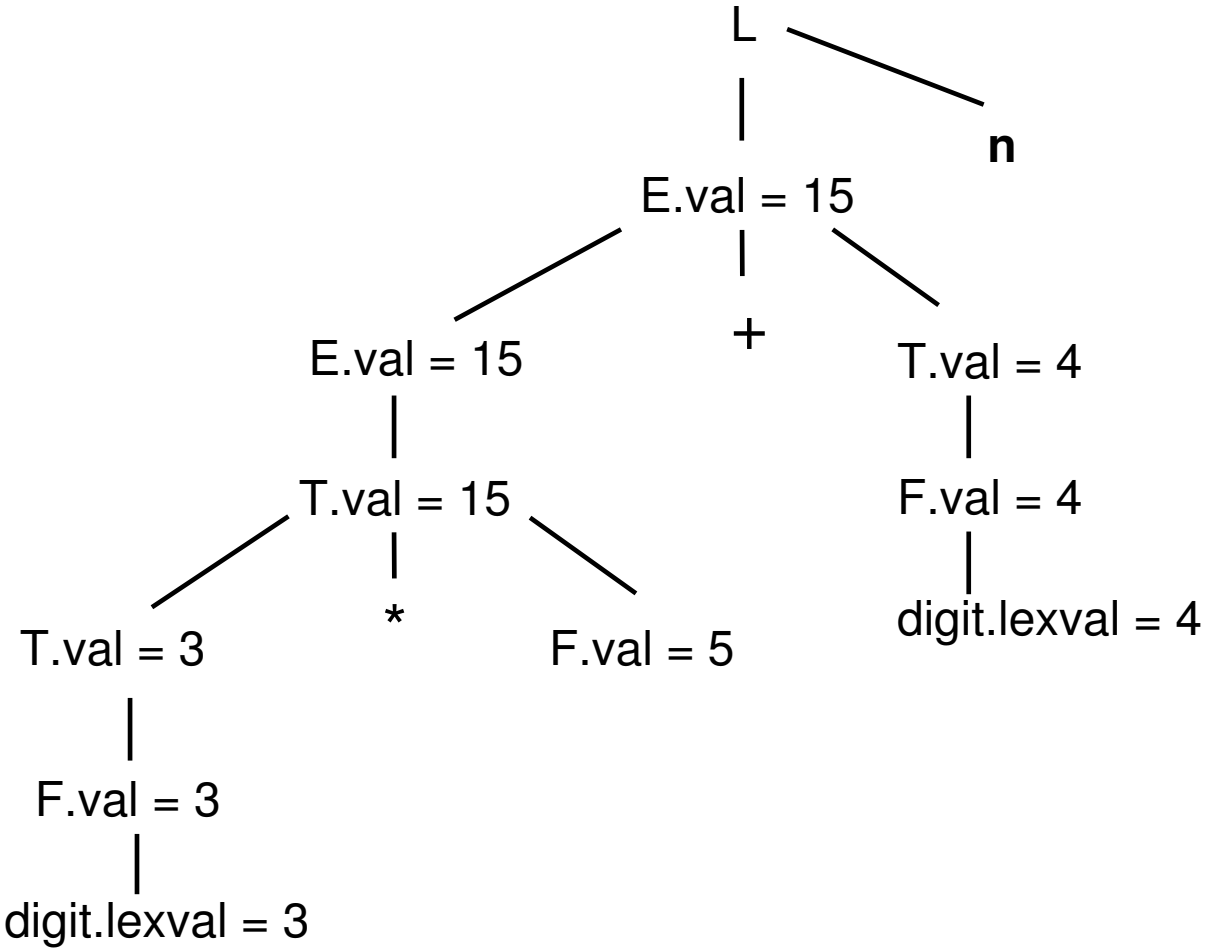
- Vamos analisar uma definição definida pela sintaxe para uma calculadora.
- Esta definição associa um atributo sintetizado inteiro com os não terminais E, T e F
- Na gramática, o terminal n representa um caractere de nova linha

Definição dirigida pela sintaxe

Produção	**Regra Semântica**
$L \rightarrow E$ **n**	Imprimir(E.val)
$E \rightarrow E_1$**+**T	E.val:= E_1.val + T.val
$E \rightarrow T$	E.val:= T.val
$T \rightarrow T_1$ * F	T.val:= T_1.val * F.val
$T \rightarrow F$	T.val:= F.val
$F \rightarrow$ **(**E**)**	F.val:= E.val
$F \rightarrow$ **dígito**	F.val:= **dígito**.lexval

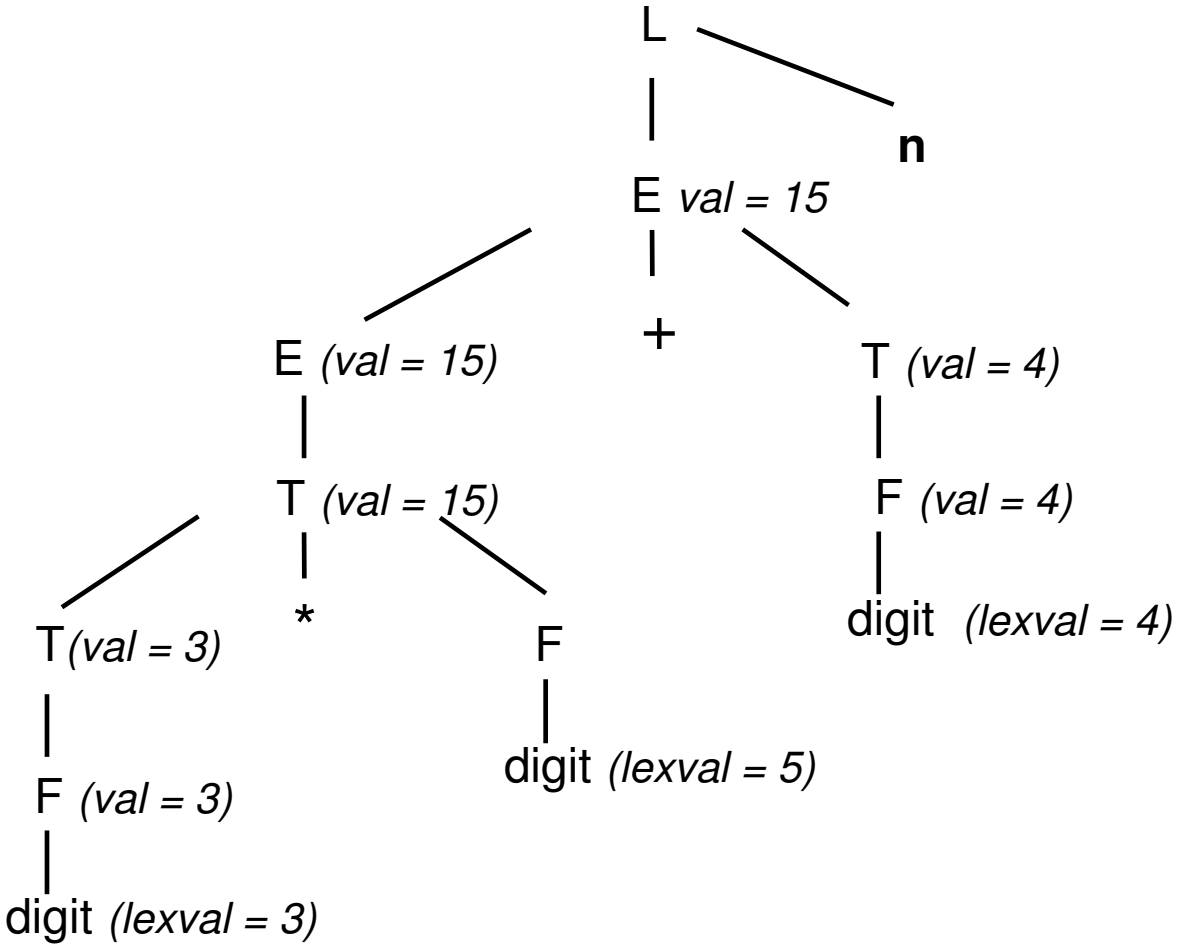
Definição dirigida pela sintaxe

- Árvore gramatical anotada



Definição dirigida pela sintaxe

- Árvore gramatical anotada (forma alternativa)



Definição dirigida pela sintaxe

- Considere a gramática simples a seguir para números sem sinal

número → *número dígito* |

dígito

dígito → **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9**

Exercício 1: Mostre a árvore de derivação para reconhecer o número 345

Definição dirigida pela sintaxe

- Considere a gramática simples a seguir para números sem sinal (cont.)

número → *número dígito* |
dígito

dígito → **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9**

Exercício 2: Escreva ações semânticas para produzir e escrever como saída um número inteiro sem sinal

Definição dirigida pela sintaxe

Produção	Regra Semântica
número$_1$ → *número*$_2$ *dígito*	*número*$_1$*.val* = *número*$_2$*.val* * digito.val
número → *dígito*	*número.val* = *digito.val*
dígito → **0**	digito.val = 0
dígito → **1**	digito.val = 1
dígito → **2**	digito.val = 1
dígito → **3**	digito.val = 3
dígito → **4**	digito.val = 4
dígito → **5**	digito.val = 5
dígito → **6**	digito.val = 6
dígito → **7**	digito.val = 7
dígito → **8**	digito.val = 8
dígito → **9**	digito.val = 9

Definição dirigida pela sintaxe

- Exercício: Construa a árvore gramatical anotada para o número 345

Atributos Sintetizados

- Na **tradução dirigida pela sintaxe** assume-se que os terminais tenham somente atributos sintetizados na medida em que as definições não providencie quaisquer regras semânticas
- Os valores para os atributos dos terminais costumam ser fornecidos pelo léxico:

F $\rightarrow$ **dígito** F.val= **dígito**.lexval

Atributos Sintetizados

- **Atributos sintetizados** são bastante usados na prática.
- Uma definição dirigida pela sintaxe somente com atributos sintetizados é chamada de **definição S-atribuída.**
- Os atributos costumam ser avaliados de baixo para cima, das folhas para a raiz.

Atributos Herdados

- Atributos herdados são convenientes para expressar construções de LPs em relação ao contexto em que aparecem
- Bastante útil na verificação de tipos
- Controlar se um identificador aparece do lado esquerdo (endereço) ou direito (valor) de uma atribuição

Atributos Herdados

Produção	Regra Semântica
$D \rightarrow T\ L$	L.in:= T.tipo
$T \rightarrow int$	T.tipo:= inteiro
$T \rightarrow float$	T.tipo:= real
$L \rightarrow L1, id$	L1.in:= L.in incluir_tipo(id.entrada, L.in)
$L \rightarrow id$	incluir_tipo(id.entrada, L.in)

L.in é um atributo herdado

Atributos Herdados

- Monte a árvore gramatical anotada para a cadeia

int a, B, X

Grafo de Dependências

- Se um atributo b em um nó de uma árvore de derivação depende de um atributo c, então a regra semântica de b deve ser avaliada após a regra semântica que define c
- As interdependências entre os atributos herdados e atributos sintetizados em uma árvore gramatical é denotada por um grafo dirigido chamado *grafo de dependências*

Grafo de Dependências

- Um Grafo de Dependências é construído para uma árvore gramatical:
  - Ele mostra as interdependências entre os atributos dos nós da árvore gramatical
  - Objetivo: Definir uma ordem de avaliação das regras semânticas em uma tradução dirigida pela sintaxe

Grafo de Dependências

- Cada regra semântica deve ser da forma b := f(c_1, c_2, ...,c_k)
- O grafo possui um nó para cada atributo. Existe uma aresta de c_i para b se o atributo b depende de c_i

```
para cada nó n na árvore gramatical faça
        para cada atributo a do símbolo gramatical em n faça
                gere um nó para a no grafo de dependências
        fim para
fim para
para cada nó n na árvore gramatical faça
        para cada regra semântica b:= f(c1, c2, ...,ck) faça
                para i= 1 até k faça
                        gere uma aresta a partir do nó ci até o nó b
                fim para
        fim para
fim para
```

Ordem de Avaliação

- Uma classificação topológica é uma ordenação dos nós de um grafo que segue a regra:
  - Se $m_i \rightarrow m_j$ é uma aresta do grafo de dependências, então m_i aparece antes de m_j na classificação topológica

Ordem de Avaliação

- Uma classificação topológica indica uma ordem válida na qual as regras semânticas devem ser avaliadas
- Os atributos $c_1, ..., c_k$ dos quais uma regra semântica dependa b:= f($c_1, ..., c_k$) devem estar disponíveis antes de *f* ser avaliada

Tradução de Linguagens

- Crie uma ordenação topológica baseada em regras para a cadeia

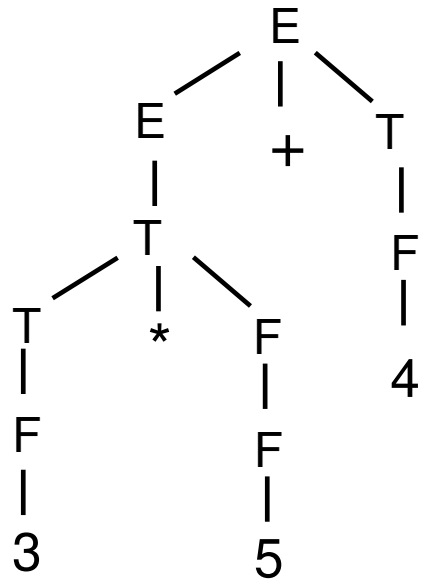
float x, y

Árvores Sintáticas

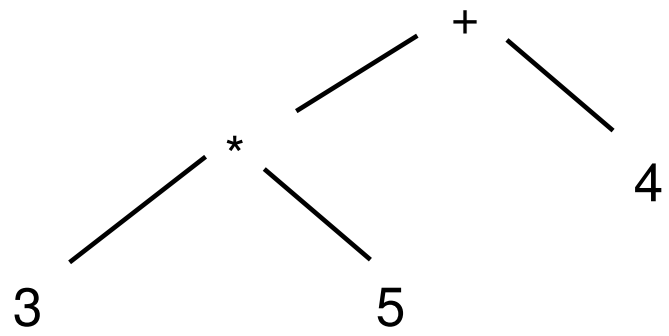
- A árvore sintática é uma representação intermediária que pode ser construída por definições dirigidas pela sintaxe
- Separa a tradução da análise sintática
- Trata-se de uma forma **condensada** de árvore gramatical

Árvore Sintática

- Uma árvore sintática (abstrata) é uma forma condensada da árvore gramatical na qual:
  - Operadores e palavras chaves não aparecem como folhas, mas como nós interiores da árvore



Árvore Gramatical



Árvore Sintática

Árvores Sintáticas

- Na árvore sintática operadores e palavras-chave não aparecem nas folhas, mas sim em nós interiores
- Traduções dirigida pela sintaxe podem ser usadas tanto em árvores gramaticais quanto em árvores sintáticas

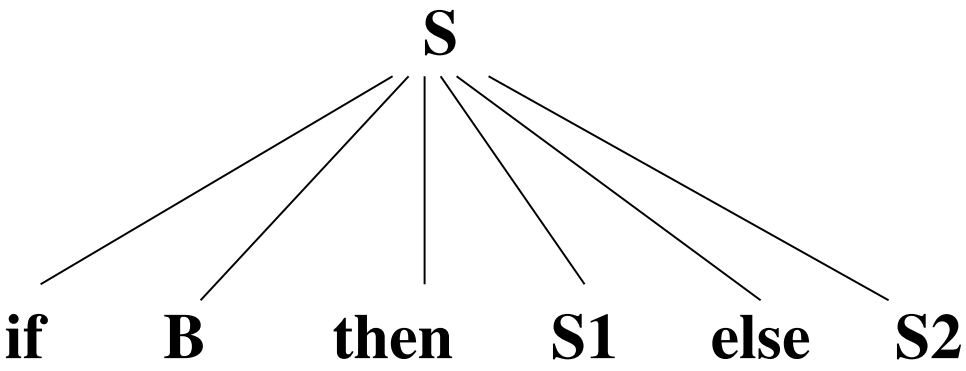
Árvores Sintáticas

Construa uma árvore gramatical e uma árvore sintática para a produção

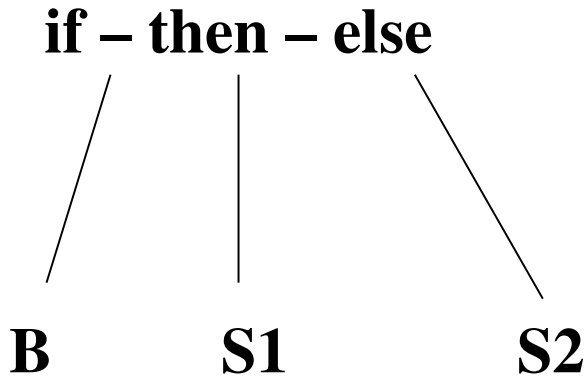
S → if B then S1 else S2

Árvores Sintáticas

Árvore Gramatical



Árvore Sintática do Comando condicional if



Árvores Sintáticas para Expressões

- A árvore sintática para expressões é correspondente à construção da expressão na forma posfixa
- É possível escrever uma definição dirigida pela sintaxe para construir uma árvore sintática de uma expressão

Árvores Sintáticas para Expressões

- Cada nó da árvore sintática pode ser implementado como um registro com vários campos.
- Em um nó temos:
  - Um campo que identifica o operador
  - E campos que são ponteiros para os operandos
- O operador é comumente chamado de label do nó.

Árvores Sintáticas para Expressões

- Para criar os nós de uma árvore sintática usaremos as funções:
  - cria_no(op, direita, esquerda)
    - Cria um nó com o operador igual a op e dois campos com ponteiros para a subárvores esquerda e direita
  - cria_folha(id, entrada_TS)
    - Cria um nó com identificador com label id e um campo contendo a entrada da Tabela de Símbolos
  - cria_folha(num, val)
    - Cria um nó com número com label id e um campo contendo o valor.

Construindo Árvores Sintáticas

Produção	Regra Semântica
$E \rightarrow E_1 + T$	E.ptr:= cria_no('+', E_1.ptr, T.ptr)
$E \rightarrow E_1 - T$	E.ptr:= cria_no('-', E_1.ptr, T.ptr)
$E \rightarrow T$	E.ptr:= T.ptr
$T \rightarrow (E)$	T.ptr:= E.ptr
$T \rightarrow id$	T.ptr:= cria_folha(id, id.entrada)
$T \rightarrow num$	T.ptr:= cria_folha(num, num.val)

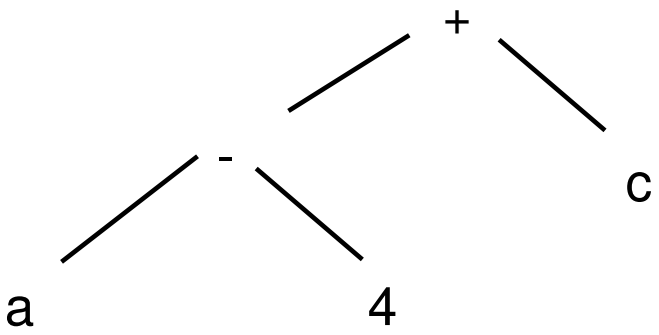
Figura 1: Definições dirigidas pela sintaxe para construção de uma árvore sintática de uma expressão

Construindo Árvores Sintáticas

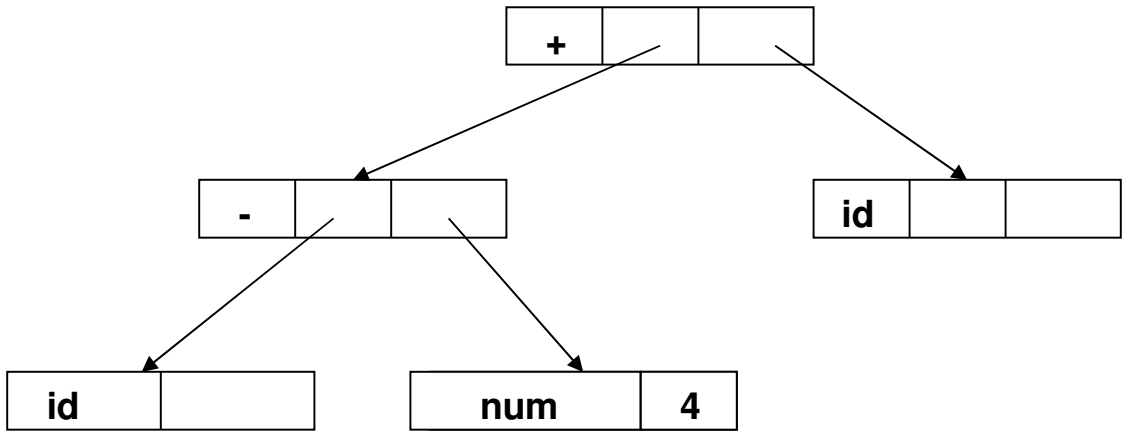
- Gere a árvore gramatical anotada para a expressão a – 4 + c

Construindo Árvores Sintáticas

- Representação da árvore sintática em uma estrutura de dados



Árvore Sintática



Estrutura de dados

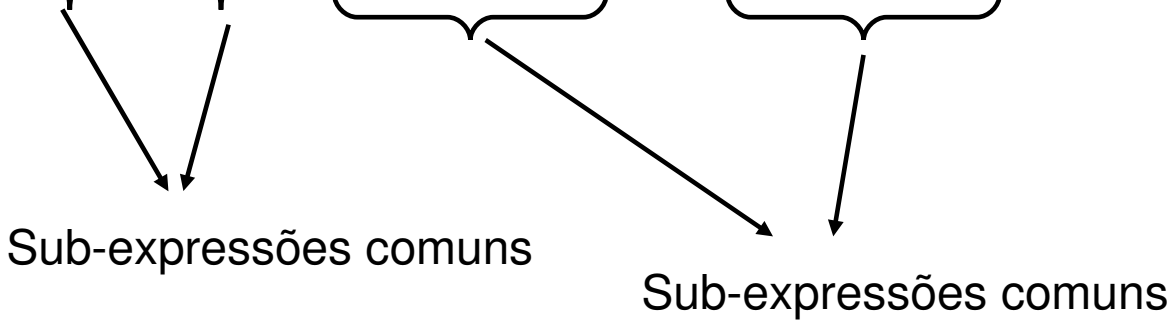
Grafo Dirigido Acíclico (DAG)

- DAG = Directed Acyclic Graph
- É um grafo usado para identificar as subexpressões comuns em uma expressão
- Tal como uma árvore sintática, um DAG possui um nó para cada subexpressão da expressão
  - A diferença é que no DAG um nó pode ter mais de um pai.

Grafo Dirigido Acíclico (DAG)

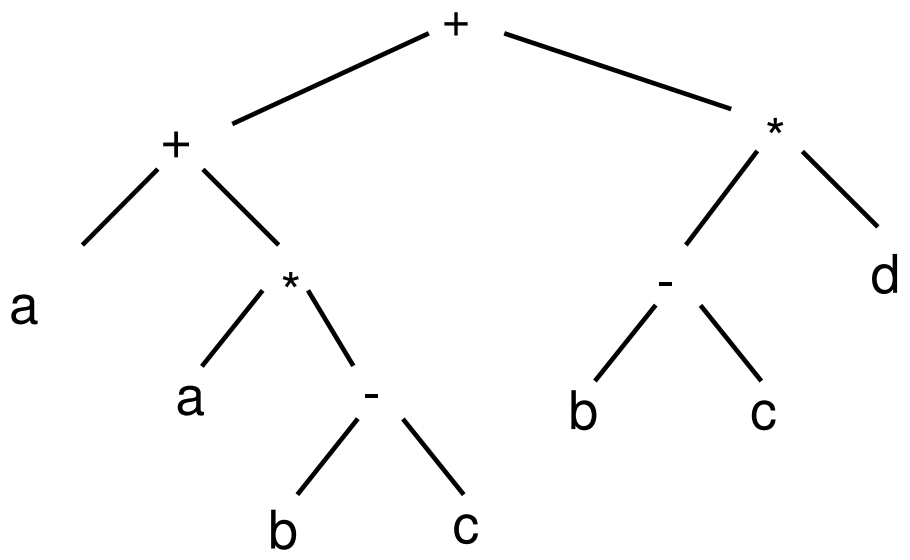
- Numa árvore sintática, sub-expressões comuns são representadas como sub-árvores duplicadas
- Seja a expressão:

a + a * (b – c) + (b – c) * d



Grafo Dirigido Acíclico (DAG)

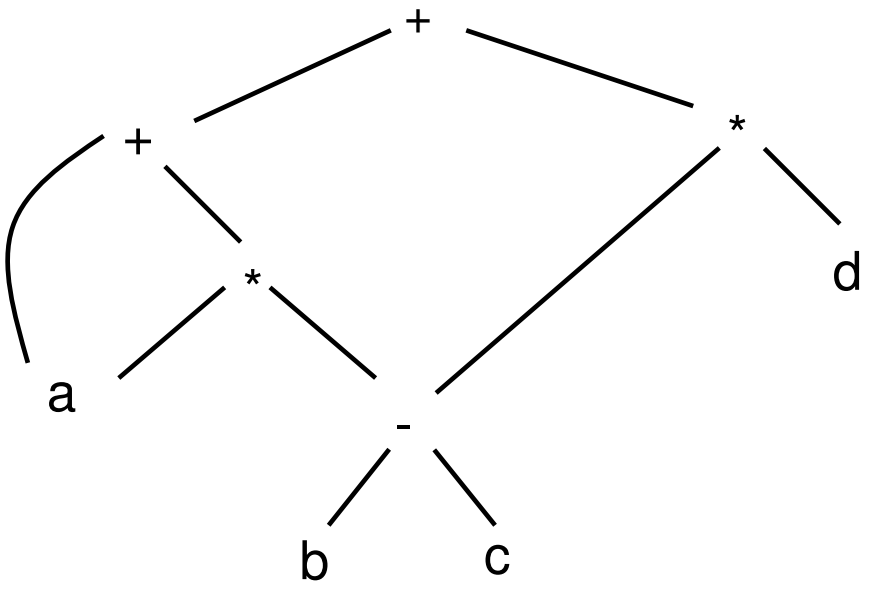
- Árvore Sintática para a expressão
  a + a * (b – c) + (b – c) * d



Árvore Sintática

Grafo Dirigido Acíclico (DAG)

- DAG da a expressão
  a + a * (b – c) + (b – c) * d



Grafo Dirigido Acíclico

Grafo Dirigido Acíclico (DAG)

- As mesmas definições dirigidas pela sintaxe da Figura 1 podem ser usadas para construir o DAG
- Basta mudarmos as funções
  - cria_no(op, direita, esquerda)
  - cria_folha(id, entrada_TS)
  - cria_folha(num, val)
- As funções deverão checar se o nó sendo criado já existe.

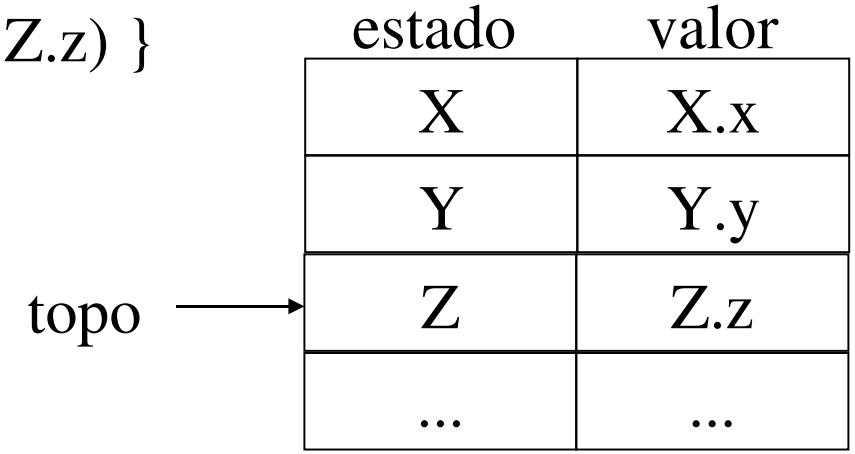
Avaliação S-atribuída

- Definições S-atribuídas são aquelas que só possuem atributos sintetizados
- O valor dos atributos pode ser controlado pela pilha sintática em um parser *bottom-up*
- A cada redução, o valor de um atributo sintetizado é calculado a partir dos símbolos empilhados que aparecem do lado direito da produção

Avaliação S-atribuída

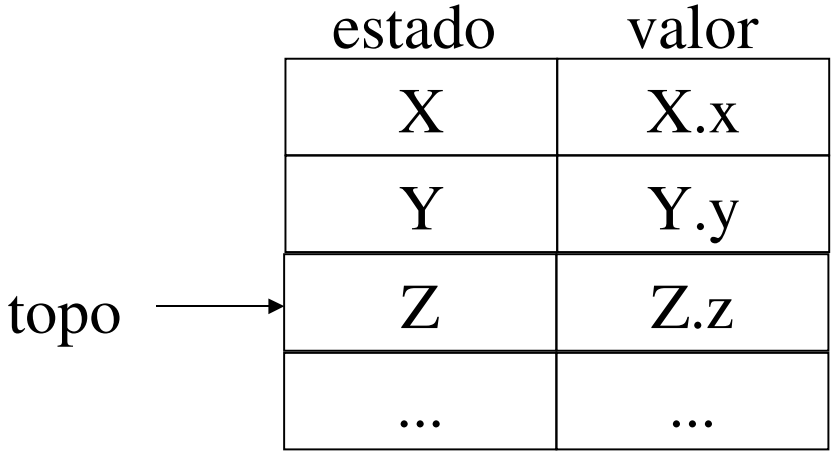
- O estado é um ponteiro ou índice de uma tabela sintática (usa-se no entanto o símbolo gramatical para facilitar)
- O valor é o valor do atributo associado ao nó gramatical correspondente a um estado

A-> XYZ { A.a := f(X.x, Y.y, Z.z) }



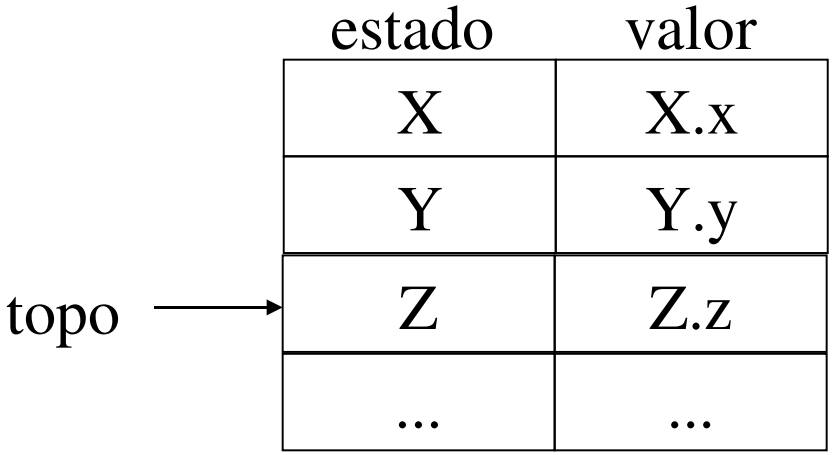
Avaliação S-atribuída

- Os atributos sintetizados são avaliados ***exatamente antes de cada redução***
- Seja A $\rightarrow$ XYZ, o configuração da pilha será:



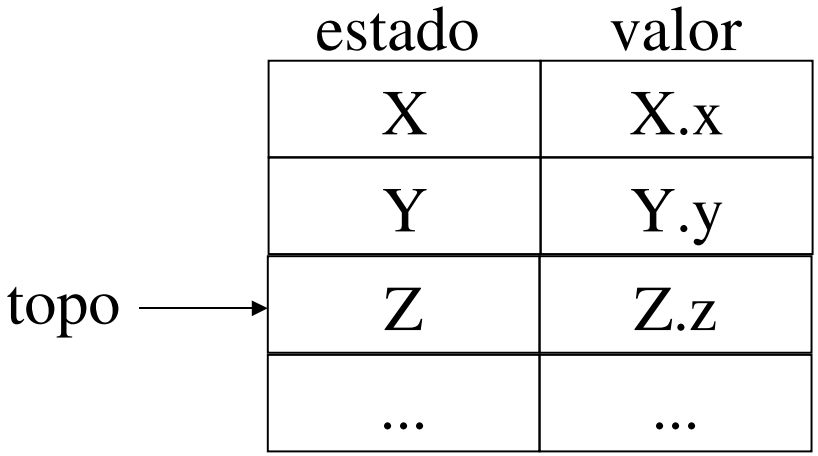
Avaliação S-atribuída

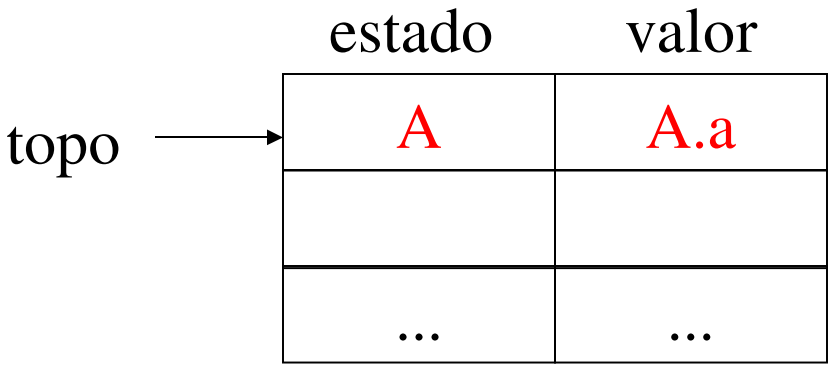
- O atributo Z.z está em *val[topo]*
- O atributo Y.y está em *val[topo-1]*
- O atributo X.x está em *val[topo-2]*
- Após a redução topo é decrementado de 2



Avaliação S-atribuída

- O estado que A será colocado onde estava X
- O valor do atributo sintetizado A.a será colocado em *val[topo-1]*





Avaliação S-atribuída

Produção	Regra Semântica
$L \rightarrow E$	Imprimir(val[topo])
$E \rightarrow E_1$**+**T	val[ntopo]:= val[topo-2] + val[topo]
$E \rightarrow T$	val[ntopo]:= val[topo]
$T \rightarrow T_1$ * F	val[ntopo]:= val[topo-2] x val[topo]
$T \rightarrow F$	val[ntopo]:= val[topo]
$F \rightarrow$ **(**E**)**	val[ntopo]:= val[topo-1]
$F \rightarrow$ **dígito**	val[ntopo]:= **dígito**.lexval

Avaliação S-atribuída

- Quando uma produção é reduzida com *r* símbolos gramaticais do lado direito, tem-se *ntopo= topo-r+1*
- Depois da execução de cada conjunto de regras semânticas, *topo= ntopo*
- Exemplo: avaliação da expressão 3*5+4

Definição L-atribuída

- Definições em que os atributos são sempre avaliados em uma ***ordem de pesquisa em profundidade***
- Uma definição dirigida pela sintaxe é L-atribuída se cada atributo herdado de X_j, onde $1 \leq j \leq n$ que esteja do lado direito de uma produção $A \rightarrow X_1 X_2 \ldots X_n$ depender somente:
  - Dos atributos dos símbolos $X_1 X_2 \ldots X_{j-1}$ à esquerda de X_j
  - Dos atributos herdados de A

Definição L-atribuída

procedimento visitar_df(n: nó)

para cada filho *m* de *n* da esquerda para direita, **faça**

avaliar os atributos herdados de *m*

visitar_df(m)

fim para

avaliar os atributos sintetizados de *n*

fim

Esquemas de Tradução

- Também são ações semânticas, tal como definições dirigidas pela sintaxe.
- Indica a ordem em que as regras semânticas devem ser avaliadas
- Exibem detalhes de implementação

Esquema de Tradução

- O esquema de tradução é uma gramática livre de contexto na qual as ações semânticas são inseridas **no lado direito** das produções

$E \rightarrow TR$

$T \rightarrow$ **num** {imprimir(**num**.val)}

$R \rightarrow$ **op_aditivo** T {imprimir(**op_aditivo**.lexema)} R

$R \rightarrow \varepsilon$

Esquemas de Tradução

- Com esquemas de tradução podemos determinar a ordem na qual ações e avaliações de atributos acontecem.

Tradução Top-Down

- Definições L-Atribuídas serão implementadas durante um analisador preditivo.
- Usaremos esquemas de tradução.

Eliminação de recursão à esquerda de um esquema de tradução

$E \rightarrow E_1 + T$	$\{ E.val = E_1.val + T.val \}$
$E \rightarrow E_1 - T$	$\{ E.val = E_1.val - T.val \}$
$E \rightarrow T$	$\{ E.val = T.val \}$
$T \rightarrow (E)$	$\{ T.val = E.val \}$
$T \rightarrow$ **num**	$\{ T.val = num.val \}$

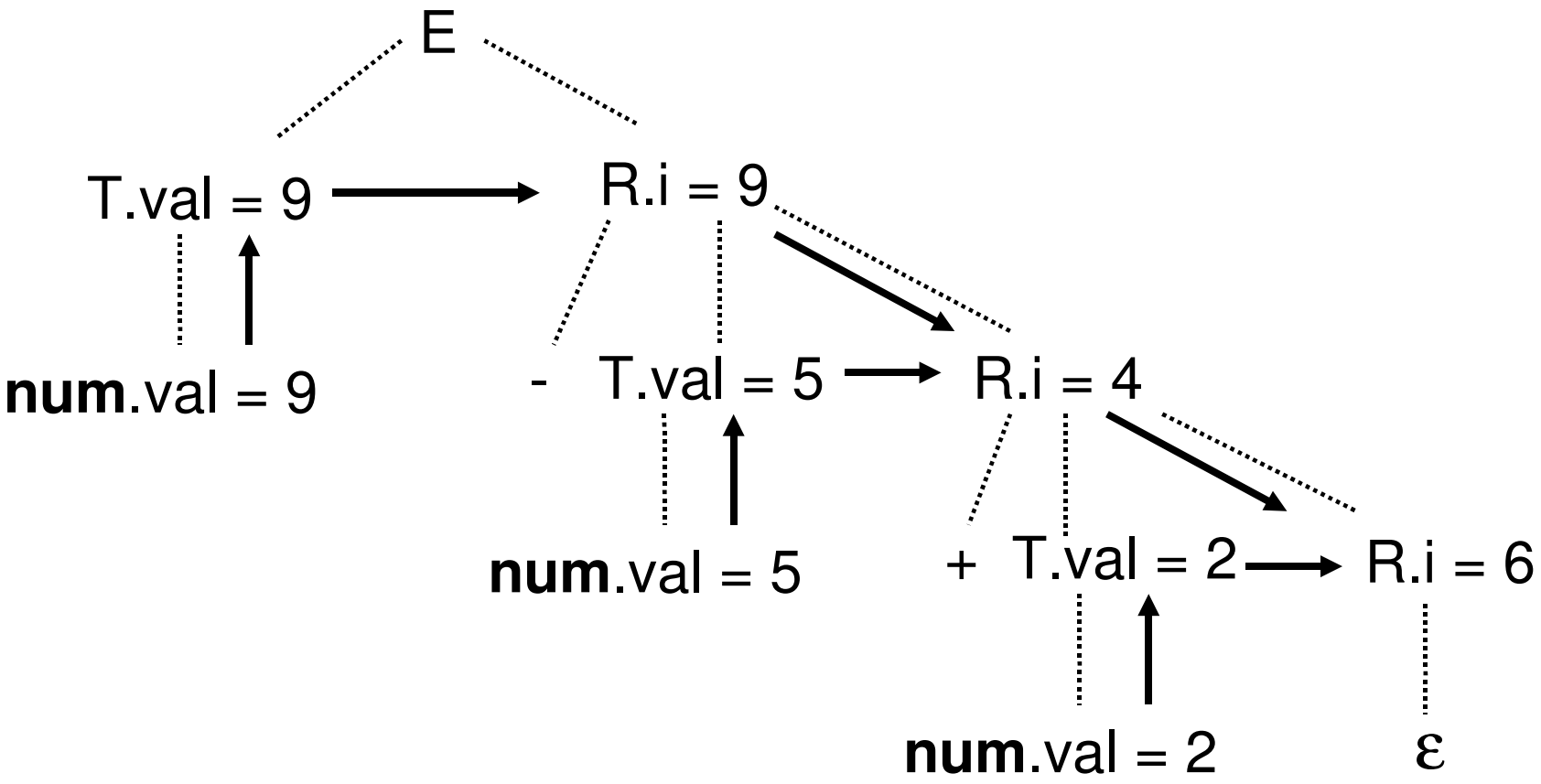
- Esquema de tradução com uma gramática recursiva à esquerda.

Eliminação de recursão à esquerda de um esquema de tradução

E -> T { *R.i = T.val* }
R { *E.val = R.s* }
R -> +
T { $R_1.i = R.i + T.val$ }
R_1 { $R.s = R_1.s$ }
R -> -
T { $R_1.i = R.i - T.val$ }
R_1 { $R.s = R_1.s$ }
R -> ε { *R.s = R.i* }
T -> (E) { *T.val := E.val* }
T -> **num** { *T.val = **num**.val* }

Esquema de tradução transformado com uma gramática recursiva à direita.

Eliminação de recursão à esquerda de um esquema de tradução



- Avaliação da expressão 9 – 5 + 2

Esquema de Tradução

- Quando usamos atributos herdados e sintetizados, usamos as heurísticas a seguir para criar esquemas de tradução
  - Um **atributo herdado** para um símbolo do lado direito precisa ser calculado em uma ação ***anterior***
  - Uma ação não pode se referir a uma atributo sintetizado de um símbolo à direita da ação
  - Um **atributo sintetizado** para um não terminal à esquerda deve ser calculado somente após o cálculo de todos os atributos que o mesmo referencie