

Outline

Avaliação dirigida pela sintaxe

Gramáticas atributivas

Grafos de dependências

Abstract syntax tree

Avaliação dirigida pela sintaxe

O resultado do analisador sintáctico é:

- **TRUE** - se o programa estiver correcto sintacticamente
- **FALSE** - se o programa não estiver correcto sintacticamente

Avaliação dirigida pela sintaxe

Tarefas

As regras sintácticas têm associadas acções, que são guiadas pela estrutura sintáctica (*syntax-directed translation*).

Tarefas:

- Inserir informações na tabela de símbolos
- Efectuar verificações semânticas (e.g., tipos das variáveis e expressões)
- Gerar código intermédio, para posteriormente gerar código-máquina
- Emitir mensagens de erro
- Avaliar expressões

Avaliação dirigida pela sintaxe

Vantagens

Vantagens:

- Maior rapidez – efectuada uma única passagem sobre a sequência de entrada
- Menos recursos memória – não é necessária a construção de uma representação do programa

Outline

Avaliação dirigida pela sintaxe

Gramáticas atributivas

Grafos de dependências

Abstract syntax tree

Avaliação dirigida pela sintaxe

Gramáticas atributivas

Avaliação dirigida pela sintaxe é composta por:

- Gramática livre de contexto G
- Conjunto de atributos associados aos símbolos de G
- Conjunto de acções semânticas associadas às produções de G

Avaliação dirigida pela sintaxe

Gramáticas atributivas

As acções semânticas são executadas quando a produção é seleccionada pelo analisador sintáctico.

Avaliação dirigida pela sintaxe

Gramáticas atributivas

As acções semânticas são executadas quando a produção é seleccionada pelo analisador sintáctico.

Exemplo:

Produção	Regra semântica
$E \rightarrow E_1 + T$	$E.val = E_1.val + T.val$
$E \rightarrow T$	$E.val = T.val$
$T \rightarrow T_1 * F$	$T.val = T_1.val * F.val$
$T \rightarrow F$	$T.val = F.val$
$F \rightarrow (E)$	$F.val = E.val$
$F \rightarrow \mathbf{num}$	$F.val = \mathbf{num.lexval}$

Gramáticas atributivas

Atributos

A cada produção $A \rightarrow \alpha_1 \alpha_2 \dots \alpha_n$, é associado um conjunto de regras semânticas $b = f(c_1, c_2, \dots, c_k)$, onde f é uma função, e b e c_i são atributos.

Gramáticas atributivas

Atributos

A cada produção $A \rightarrow \alpha_1 \alpha_2 \dots \alpha_n$, é associado um conjunto de regras semânticas $b = f(c_1, c_2, \dots, c_k)$, onde f é uma função, e b e c_i são atributos.

O atributo b , associado a um **símbolo não terminal**, é considerado:

- **sintetizado**
 - b é associado a A
 - c_i são atributos associados aos símbolos α_j

Gramáticas atributivas

Atributos

A cada produção $A \rightarrow \alpha_1 \alpha_2 \dots \alpha_n$, é associado um conjunto de regras semânticas $b = f(c_1, c_2, \dots, c_k)$, onde f é uma função, e b e c_i são atributos.

O atributo b , associado a um **símbolo não terminal**, é considerado:

- **sintetizado**

- b é associado a A
- c_i são atributos associados aos símbolos α_j

- **herdado**

- b é associado a um α_t
- c_i é associado a A e/ou aos símbolos α_j ($j \neq t$)

Gramáticas atributivas

Atributos

A cada produção $A \rightarrow \alpha_1 \alpha_2 \dots \alpha_n$, é associado um conjunto de regras semânticas $b = f(c_1, c_2, \dots, c_k)$, onde f é uma função, e b e c_i são atributos.

O atributo b , associado a um **símbolo não terminal**, é considerado:

- **sintetizado**

- b é associado a A
- c_i são atributos associados aos símbolos α_j

- **herdado**

- b é associado a um α_t
- c_i é associado a A e/ou aos símbolos α_j ($j \neq t$)

Nota:

Atributos associados a **símbolos terminais**, são sempre **sintetizados**. São calculados durante a análise lexical.

Avaliação dirigida pela sintaxe

Atributos sintetizados e herdados

Exemplo com atributos sintetizados:

Produção	Regra semântica
$T \rightarrow T_1 * F$	$T.val = T_1.val * F.val$
$T \rightarrow F$	$T.val = F.val$
$F \rightarrow \mathbf{digit}$	$F.val = \mathbf{digit.lexval}$

Avaliação dirigida pela sintaxe

Atributos sintetizados e herdados

Exemplo com atributos sintetizados:

Produção	Regra semântica
$T \rightarrow T_1 * F$	$T.val = T_1.val * F.val$
$T \rightarrow F$	$T.val = F.val$
$F \rightarrow \mathbf{digit}$	$F.val = \mathbf{digit.lexval}$

Exemplo com atributos sintetizados e herdados:

Produção	Regra semântica
$T \rightarrow FT'$	$T'.inh = F.syn; T.syn = T'.syn$
$T' \rightarrow *FT'_1$	$T'_1.inh = T'.inh * F.syn; T'.syn = T'_1.syn$
$T' \rightarrow \epsilon$	$T'.syn = T'.inh$
$F \rightarrow \mathbf{digit}$	$F.syn = \mathbf{digit.lexval}$

Outline

Avaliação dirigida pela sintaxe

Gramáticas atributivas

Grafos de dependências

Abstract syntax tree

Grafos de dependências

Uma definição semântica pode conter simultaneamente vários atributos herdados e sintetizados.

Grafos de dependências

Uma definição semântica pode conter simultaneamente vários atributos herdados e sintetizados.

Necessário: ordenar as acções de forma a que as avaliações sejam feitas pela ordem correcta!

Grafos de dependências

Uma definição semântica pode conter simultaneamente vários atributos herdados e sintetizados.

Necessário: ordenar as acções de forma a que as avaliações sejam feitas pela ordem correcta!

Algoritmo:

- Nó para cada atributo associado a símbolo X
- Se atributo sintetizado $A.b$ é definido em termos de $X.c$ adicionar arco de $X.c$ para $A.b$
- Se atributo herdado $B.c$ é definido em termos de $X.a$ adicionar arco de $X.a$ para $B.c$

Grafos de dependências

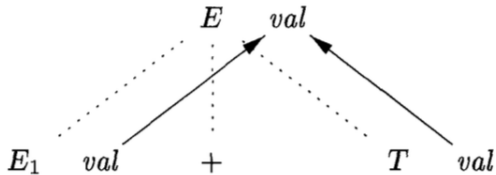
Exemplo 1

PRODUCTION

$E \rightarrow E_1 + T$

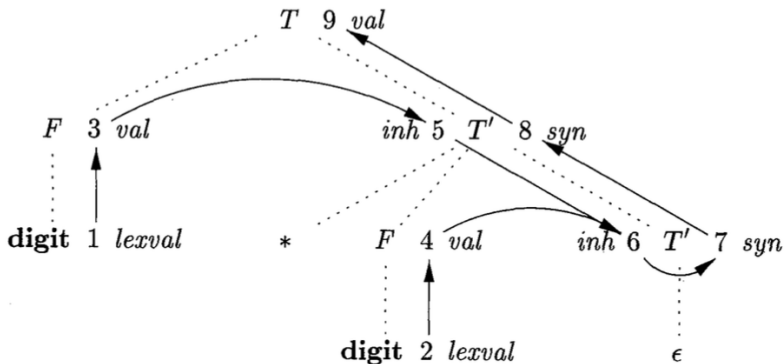
SEMANTIC RULE

$E.val = E_1.val + T.val$



Grafos de dependências

Exemplo 2



Gramáticas atributivas

Dependências circulares

Dependência circular entre atributos herdados e sintetizados:

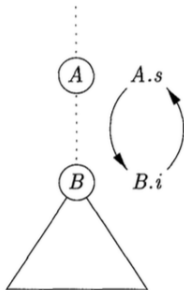
PRODUCTION

$A \rightarrow B$

SEMANTIC RULES

$A.s = B.i;$

$B.i = A.s + 1$



Impossibilidade de avaliação $A.s$ ou $B.i$!!

Gramáticas atributivas

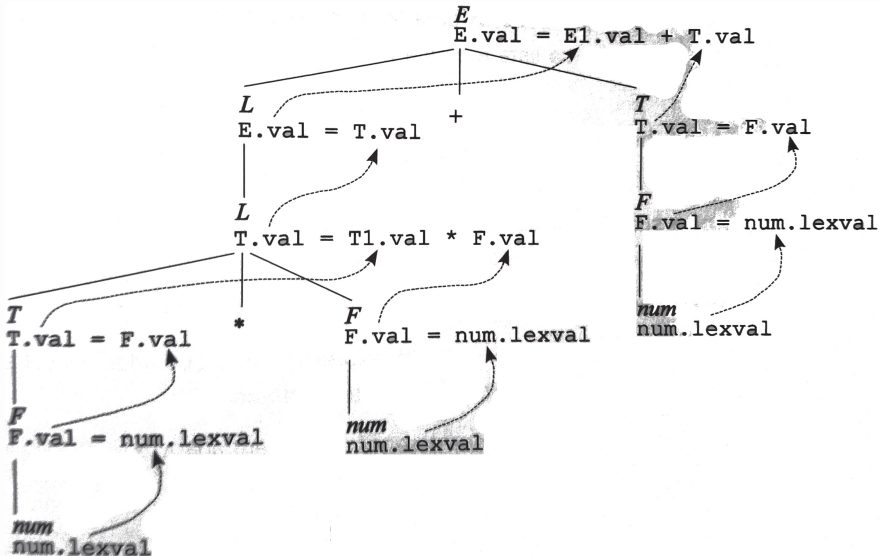
Gramática de S-atributos:

- Os valores dos atributos dependem apenas dos valores dos atributos dos nós filhos
- Atributos são avaliados de forma ascendente

Pode ser usado na análise sintáctica ascendente ([Parsing LR](#)).

Grafo de dependências

Exemplo – Gramática de S-atributos



Gramáticas atributivas

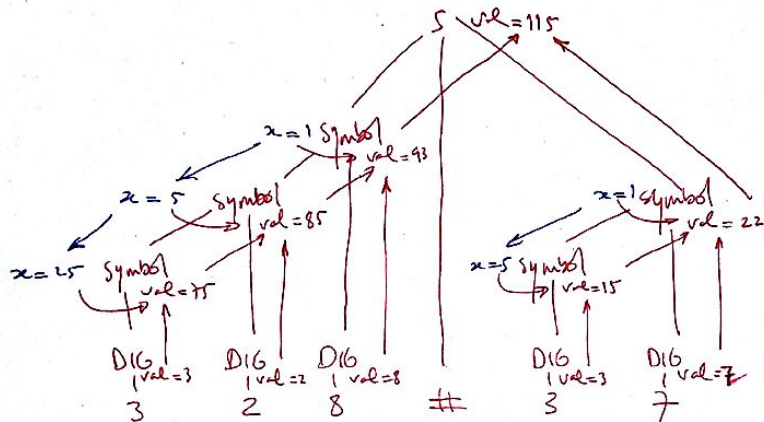
Gramática de L-atributos:

- Os valores dos atributos dependem apenas:
 - dos valores dos atributos à sua esquerda na produção
 - dos valores herdados da cabeça da regra
- Os valores dos atributos podem ser avaliados da esquerda para a direita (*left-to-right*)

Pode ser usado na análise sintáctica descendente ([Parsing LL](#)).

Grafo de dependências

Exemplo – Gramática de L-atributos



Outline

Avaliação dirigida pela sintaxe

Gramáticas atributivas

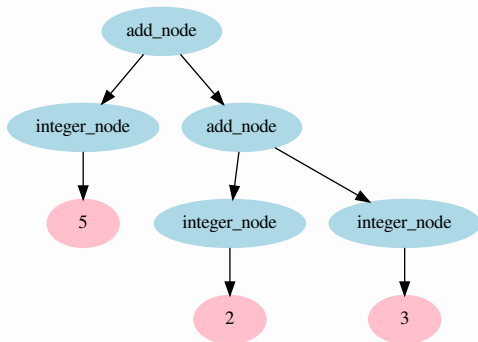
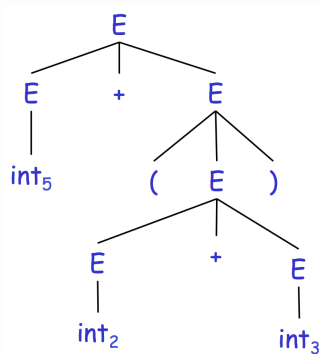
Grafos de dependências

Abstract syntax tree

Abstract Syntax Tree - AST

- Representação da estrutura do programa
 - Difere da árvore de parsing!
- Serve para análise semântica
 - Utilização de 1 ou mais passos sobre a AST
- Serve para geração de código intermédio

Árvores de parsing vs. ASTs

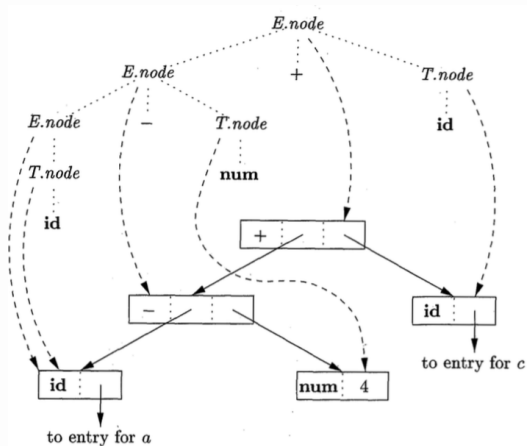


- ASTs eliminam detalhe desnecessário
- Estrutura de dados essencial em compiladores

Aplicações de SDTs – ASTs

PRODUCTION	SEMANTIC RULES
1) $E \rightarrow E_1 + T$	$E.node = \mathbf{new} \text{ Node}('+', E_1.node, T.node)$
2) $E \rightarrow E_1 - T$	$E.node = \mathbf{new} \text{ Node}('-', E_1.node, T.node)$
3) $E \rightarrow T$	$E.node = T.node$
4) $T \rightarrow (E)$	$T.node = E.node$
5) $T \rightarrow \mathbf{id}$	$T.node = \mathbf{new} \text{ Leaf}(\mathbf{id}, \mathbf{id.entry})$
6) $T \rightarrow \mathbf{num}$	$T.node = \mathbf{new} \text{ Leaf}(\mathbf{num}, \mathbf{num.val})$

Aplicações de SDTs – ASTs



- 1) $p_1 = \text{new Leaf}(\text{id}, \text{entry-}a);$
- 2) $p_2 = \text{new Leaf}(\text{num}, 4);$
- 3) $p_3 = \text{new Node}('-', p_1, p_2);$
- 4) $p_4 = \text{new Leaf}(\text{id}, \text{entry-}c);$
- 5) $p_5 = \text{new Node}('+', p_3, p_4);$