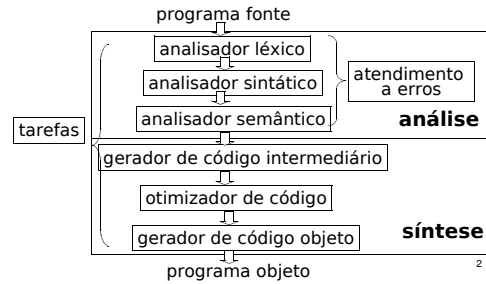


Análise semântica

Tradução orientada pela sintaxe

Estrutura de um compilador



Análise Semântica

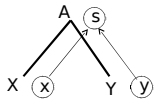
- Análise semântica:
 - Tipos (& Declarações)
 - Escopo
- Checagens estáticas:
 - Tipos: verificar se operadores são aplicados a operandos errados.
 - Ex. somar uma variável a um nome de procedimento
 - Fluxo de controle: verificar se comandos que alteram fluxo de execução estão dentro de um contexto apropriado.
 - Ex. Break
 - Unicidade: Existem situações onde objetos devem ser definidos uma única vez. Ex: rótulos do comando case
 - Relacionados ao nome: Em algumas linguagens, um mesmo nome deve aparecer duas ou mais vezes. Ex. Laços em Ada precisam ter um nome que apareça no começo e fim da construção

Anotação da árvore sintática

- Idéia: aproveitar o cálculo da árvore para propagar atributos que definem a semântica.
 - Cada não-terminal tem um atributo associado
 - Exemplo: T.tipo = real, L.tipo = T.tipo...
- Os atributos vão ser comunicados entre os nós da árvore, a medida que vai sendo construída.

Tipos de comunicação de atributos?

- Atributos Sintetizados

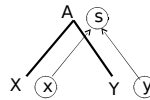


somente depende de atributos dos filhos

$A \rightarrow XY$
 $A.s := f(X.x, Y.y)$

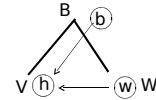
Tipos de comunicação de atributos?

- Atributos Sintetizados
- Atributos Herdados



somente depende de atributos dos filhos

$A \rightarrow XY$
 $A.s := f(X.x, Y.y)$



depende pelo menos um atributo do pai (ou de um dos irmãos)

$B \rightarrow VW$
 $V.h := g(B.b, W.w)$

5

6

Como definir atributos semânticos?

1. Regras Semânticas
2. Esquemas de Tradução

7

Como definir atributos semânticos?

1. Regras Semânticas:

Produção	Regras Semânticas
$D \rightarrow T L$	$L.tipo := T.tipo$
$T \rightarrow int$	$T.tipo := inteiro$
$T \rightarrow real$	$T.tipo := real$
$L \rightarrow L, id$	$L.tipo := L.tipo$ $incluirTipo(id.entrada, L.tipo)$
$L \rightarrow id$	$incluirTipo(id.entrada, L.tipo)$

8

Como definir atributos semânticos ?

1. Regras Semânticas:

Produção	Regras Semânticas
$D \rightarrow T L$	$L.tipo := T.tipo$
$T \rightarrow int$	$T.tipo := inteiro$
$T \rightarrow real$	$T.tipo := real$
$L \rightarrow L, id$	$L.tipo := L.tipo$ $incluirTipo(id.entrada, L.tipo)$
$L \rightarrow id$	$incluirTipo(id.entrada, L.tipo)$

confuso !

9

Como definir atributos semânticos ?

1. Regras Semânticas:

Produção	Regras Semânticas
$D \rightarrow T L$	$L.tipo := T.tipo$
$T \rightarrow int$	$T.tipo := inteiro$
$T \rightarrow real$	$T.tipo := real$
$L \rightarrow L_1, id$	$L_1.tipo := L.tipo$ $incluirTipo(id.entrada, L.tipo)$
$L \rightarrow id$	$incluirTipo(id.entrada, L.tipo)$

índice é usado para diferenciar atributos iguais ◦

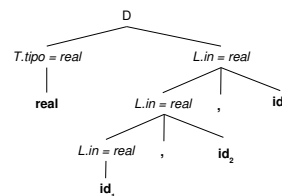
Como fazer a avaliação

- Em tempo de compilação, obter uma ordem de classificação a partir da classificação topológica do grafo de dependências
 - Ordenação topológica = que respeita a ordem parcial definida pelo grafo de dependências.
- Esta abordagem pode falhar
 - quando este grafo possui ciclos

11

Como avaliar atributos

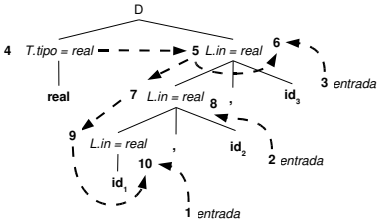
- Dependência define ordem de avaliação



12

Como avaliar atributos

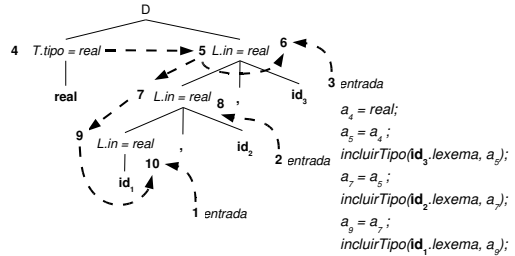
- Dependência define ordem de avaliação
árvore de derivação + grafo de dependências



13

Como avaliar atributos

- Dependência define ordem de avaliação
árvore de derivação + grafo de dependências

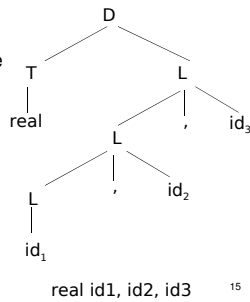


Tradução Dirigida Pela Sintaxe

- Análise Semântica e Geração de Código coordenada pela árvore sintática produzida

Produção

$D \rightarrow T L$
 $T \rightarrow \text{int} \mid \text{real}$
 $L \rightarrow L, \text{id} \mid \text{id}$



real id1, id2, id3

15

Como definir atributos semânticos?

- Regras Semânticas
- Esquemas de Tradução

16

Como definir atributos semânticos?

2. Esquemas de tradução:

- Regras semânticas são expressadas como ações sobre não-terminais no lado direito de uma produção (similar ao YACC)

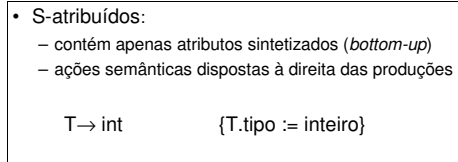
- $T \rightarrow \text{int} \{T.\text{tipo} := \text{inteiro}\}$
 - $D \rightarrow T \{L.\text{in} := T.\text{tipo}\} L$

17

Como definir atributos semânticos?

Como implementar os esquemas de tradução?

- Esquemas gerais são difíceis de construir
- Restringe-se a classes de esquemas:



Implementação de esquemas S-atribuídos

- Bottom-up Parser
- Atributos sintetizados são avaliados à medida que a cadeia de entrada é reconhecida pelo analisador
- Usa duas pilhas para reconhecer a sentença de entrada :
 - estado + atributo

Somente atributos sintetizados

19

Implementação de esquemas S-atribuídos

- Bottom-up Parser
- Atributos sintetizados são avaliados à medida que a cadeia de entrada é reconhecida pelo analisador
- Usa duas pilhas para reconhecer a sentença de entrada :
 - estado + atributo

Estado	Atributo
...	...
X	X.x
Y	Y.y
Z	Z.z
...	...

Somente atributos sintetizados

20

Exemplo (Implem. S-atribuídos)

Produções	Código
$L \rightarrow E =$	<code>print(Atributo[topo])</code>
$E \rightarrow E_1 + T$	<code>Atributo[topo] := Atributo[topo-2] + Atributo[topo]</code>
$E \rightarrow T$	
$T \rightarrow T_1 * F$	<code>Atributo[topo] := Atributo[topo-2] * Atributo[topo]</code>
$T \rightarrow F$	
$F \rightarrow (E)$	<code>Atributo[topo] := Atributo[topo-1]</code>
$F \rightarrow \text{digit}$	<code>Atributo[topo] := digit.lexval</code>

$ntopo = topo - r + 1$
onde a produção com r símbolos a direita é reduzida

21

Como definir atributos semânticos?

Como implementar os esquemas de tradução?

- Esquemas gerais são difíceis de construir
- Restringe-se a classes de esquemas:

- L-atribuídos:
 - restringem o uso de atributos herdados
$$D \rightarrow T \{L.in := T.tipo\} L$$

22

Implementação de Esquemas L-atribuídos

- Podem ser avaliados por uma busca em profundidade
- Um esquema de tradução é L-atribuído se:
 - cada atributo herdado de X_j com $1 \leq j \leq n$ do lado direito de $A \rightarrow X_1 X_2 \dots X_n$ depende somente:
 - dos atributos dos símbolos X_1, X_2, \dots, X_{j-1} à esquerda de X_j na produção,
 - dos atributos herdados de A
- Todo o esquema S-atribuído é L-atribuído pois as restrições acima aplicam-se somente a atributos herdados

23

Esquemas de tradução L-atribuídos

Procedure dvisit (n : node);

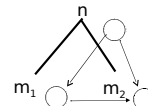
Begin

Para cada filho m_i de n, da esquerda para direita, faça

Avalie os atributos herdados de m_i
dvisit(m_i)

Avalie os atributos sintetizados de n

End



24

Exemplo

Produção

$A \rightarrow BC$ $B.h := f_1(A.h)$ L-atribuído ?
 $C.h := f_2(B.s)$
 $A.s := f_3(C.s)$

$A \rightarrow DE$ $E.h := f_4(A.h)$ L-atribuído ?
 $D.h := f_5(E.s)$
 $A.s := f_6(D.s)$

25

Exemplo

Produção

$A \rightarrow BC$ $B.h := f_1(A.h)$ L-atribuído ?
 $C.h := f_2(B.s)$ *Sim, todas as dependências estão satisfeitas.*
 $A.s := f_3(C.s)$

$A \rightarrow DE$ $E.h := f_4(A.h)$ L-atribuído ?
 $D.h := f_5(E.s)$ *Não, o atributo herdado de D depende do atributo sintetizado de E e E está à direita de D.*
 $A.s := f_6(D.s)$

26

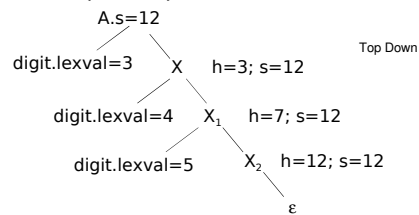
Implementação de esquemas L-atribuídos

- Facilmente realizada através da análise descendente (top-down)
- Não admite recursividade à esquerda
- Possibilita o uso conjunto de atributos herdados e sintetizados, com alguns cuidados:
 - Um atr. herdado para um símbolo à direita de uma produção deve ter sido computado antes.
 - Uma regra não pode referenciar um atr. sintetizado de um símbolo mais à direita.
 - Um atr. sintetizado de um símbolo à esquerda só pode ser computado após o cálculo de todos os atributos que referencia.

27

Árvore de derivação para 3 4 5

$A \rightarrow \text{digit}$ $\{X.h := \text{digit.lexval}\} X \{A.s := X.s\}$
 $X \rightarrow \text{digit}$ $\{X_1.h := X.h + \text{digit.lexval}\} X_1 \{X.s := X_1.s\}$
 $X \rightarrow \epsilon$ $\{X.s := X.h\}$



28

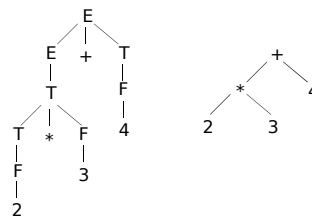
Árvores de sintaxe abstrata

- Nem sempre se calcula diretamente o valor final do atributo!
- Pode-se calcular, com as regras semânticas, uma representação intermediária:
 - Árvore de sintaxe abstrata
- Forma condensada da árvore de derivação
- Somente operandos da linguagem aparecem como folhas
- Operadores são nós interiores
- Desaparecem produções do tipo $A \rightarrow B$, $B \rightarrow C$

29

Exemplo: Sentença $2*3+4$

Árvore de derivação Árvore de sintaxe



30

Construção da árvore de sintaxe

Produções	Regras semânticas
$E \rightarrow E_1 + T$	$E.ptr := \text{geranodo}("+", E_1.ptr, T.ptr)$
$E \rightarrow E_1 - T$	$E.ptr := \text{geranodo}("-", E_1.ptr, T.ptr)$
$E \rightarrow T$	$E.ptr := T.ptr$
$T \rightarrow (E)$	$T.ptr := E.ptr$
$T \rightarrow id$	$T.ptr := \text{gerafolha}("id", id.nome)$
$T \rightarrow num$	$T.ptr := \text{gerafolha}("num", num.lexval)$

31

Esquema de tradução para expressões

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

Eliminando a recursão e modificando as regras

$E \rightarrow T$	$\{R.h := T.s\}$	R	$\{E.s := R.s\}$
$E \rightarrow + T$	$\{R_1.h := R.h + T.s\}$	R_1	$\{R.s := R_1.s\}$
$E \rightarrow - T$	$\{R_1.h := R.h - T.s\}$	R_1	$\{R.s := R_1.s\}$
$E \rightarrow \epsilon$	$\{R.s := R.h\}$		
$T \rightarrow (E)$	$\{T.s := E.s\}$		
$T \rightarrow num$	$\{T.s := num.lexval\}$		

Exemplo 2 – esquema L-atribuído

$S \rightarrow A \{B.h = A.s\} B \{C.h = B.s\} C \{S.s = A.s + B.s + C.s\}$
 $A \rightarrow (num_1 + num_2) \{A.s = num_1.val + num_2.val\}$
 $B \rightarrow A \{B_1.h = A.s\} B_1 \{C.h = B_1.s\} C \{B.s = A.s + B_1.s + C.s\}$
 $B \rightarrow (num_1 * num_2) \{B.s = B.h + num_1.s * num_2.s\}$
 $C \rightarrow (num_1 - num_2) \{C.s = C.h + num_1.s - num_2.s\}$

Efetuar o cálculo referente ao string de entrada:
(5+3)(7+2)(1*6)(2-4)(5-2)

33

Uso de análise *bottom-up* com gramáticas L-atribuídas

- Problema com os atributos herdados...
 - Eles vêm dos pais, que serão avaliados depois dos filhos... (ação efetuada ao fazer um Reduce!)
- Deve-se executar uma ação na parte “do meio” de uma regra:
 - $A \rightarrow X \{ Y.h = f(X.s) ; \} Y$
- Para isso, basta introduzir um não-terminal artificial na GLC:
 - $A \rightarrow X \text{ Xacao } Y$
 - $\text{Xacao} \rightarrow \epsilon \{ Y.h = f(X.s) ; \}$
 - (obs: é preciso memorizar um contexto para saber o que são $Y.h$ e $X.s$ na nova regra !)

34

Implementação de esquemas de tradução L-atribuídos

```

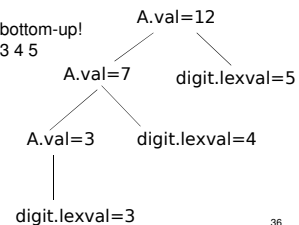
Procedure dfvisit (n : node);
Begin
  Para cada filho mi de n, da esquerda para direita, faça
    Avalie os atributos herdados de mi
    Dfvisit(mi)
  Avalie os atributos sintetizados de n
End
    
```

35

Exemplo de implementação top-down

$A \rightarrow A_1 \text{ digit} \{A.val := A_1.val + \text{digit.lexval}\}$
 $A \rightarrow \text{digit} \{A.val := \text{digit.lexval}\}$

Possibilita uma análise bottom-up!
 Exemplo: derivação de 3 4 5



36

Transformação em esquema L-atribuído

$A \rightarrow A_1 \text{ digit} \quad \{A.val := A_1.val + \text{digit.lexval}\}$

$A \rightarrow \text{digit} \quad \{A.val := \text{digit.lexval}\}$

- Eliminando a recursividade...

– Cria-se dois tipos de atributos (s e h)

$A \rightarrow \text{digit} \{X.h := \text{digit.lexval}\} X \{A.s := X.s\}$

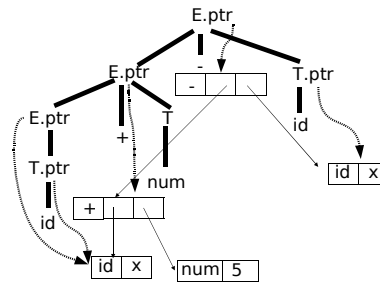
$X \rightarrow \text{digit} \{X_1.h := X.h + \text{digit.lexval}\}$

$X_1 \{X.s := X_1.s\}$

$X \rightarrow \epsilon \quad \{X.s := X.h\}$

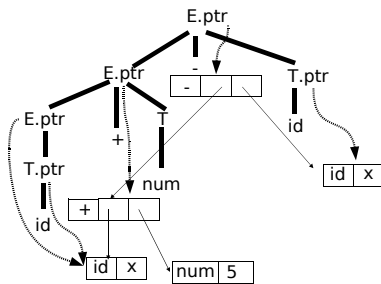
37

Árvore sintática expressão $x + 5 - y$



38

Árvore sintática expressão $x + 5 - y$



39

Bibliografia e Exercícios

Livro Price e Toscani. Capítulo 4. Todos txcrcios.

Aho, Sethi, Ullma. Capítulo 5, seções 5.1 a 5.6. Exercícios 5.1 a 5.17

40