

– INF01147 –
Compiladores

Análise Sensível ao Contexto (Análise Semântica 2/2)

Prof. Lucas M. Schnorr
– Universidade Federal do Rio Grande do Sul –

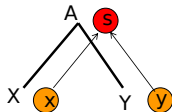


Plano da Aula de Hoje

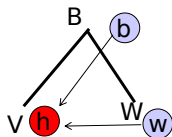
- ▶ Revisão
- ▶ Implementação do esquema S-Atribuído
- ▶ Implementação do esquema L-Atribuído
- ▶ Exercícios
- ▶ Sumário da Análise Semântica

Revisão

- ▶ Anotação da árvore de derivação
 - ▶ Atributos sintetizados
 - ▶ Atributos herdados
- ▶ Atributos Sintetizados
 - Depende somente de atributos dos filhos



- ▶ Atributos Herdados
 - Depende de atributos do pai ou dos irmãos



Revisão

- ▶ Para calcular os valores dos atributos
 - ▶ Varre a árvore de derivação já criada
 - ▶ Utiliza-se esquemas de tradução
- ▶ Esquema de Tradução
 - ▶ Regras de produção + Ações Semânticas
 - ▶ Tradução Orientada pela Sintaxe

D	→	T L	L.h = T.tipo
T	→	int	T.tipo = inteiro
T	→	float	T.tipo = flutuante
L	→	L ₁ , id	L ₁ .h = L.h
			adicionaTipo (id.key, L.h)
L	→	id	adicionaTipo (id.key, L.h)

Revisão

► Esquema S-Atribuído

- Apenas atributos sintetizados
- Ações semânticas estão sempre à direita das produções
- Calculados somente no momento da redução
- Exemplo

$$L \rightarrow E \{ L.val = E.val \}$$

► Esquema L-Atribuído

- Atributos sintetizados e herdados
- Restrições: $A \rightarrow X_1 X_2 \dots X_n$ e um atributo herdado $X_i.h$
- Calculados a medida que o reconhecimento avança
- Exemplo

$$D \rightarrow T \{ L.in = T.tipo + D.val \} L$$

- Atributos que serão herdados por L são calculados antes

Revisão

- Considerando o esquema de tradução

$$\begin{array}{lll} S & \rightarrow & A_1 A_2 \quad \{ A_1.h = 1; A_2.h = 2; \} \\ A & \rightarrow & a \quad \{ \text{print}(A.h) \} \end{array}$$

- Vamos ver o cálculo de atributos para a entrada **aa**

- Re-escrevendo o esquema em **L-Atribuído**

$$\begin{array}{lll} S & \rightarrow & \{ A_1.h = 1; \} A_1 \{ A_2.h = 1; \} A_2 \\ A & \rightarrow & a \{ \text{print}(A.h) \} \end{array}$$

- Como fica a árvore com a entrada **aa**

Implementação de Esquemas L-Atribuído

Implementação L-Atribuído

- ▶ Focado em analisadores **descendentes**
 - ▶ Possibilitam o uso de atributos herdados e sintetizados (dentro dos limites do esquema L-Atribuído)

- ▶ Supondo a seguinte gramática

$$E \rightarrow T R$$
$$R \rightarrow + T R$$
$$R \rightarrow - T R$$

- ▶ Analisador **descendente recursivo**
 - ▶ Uma função reconhecedora para cada não-terminal
- ▶ Idéia simples
 - ▶ Atributos herdados são os **argumentos** da função
 - ▶ Atributos sintetizados fazem parte do **retorno** da função

Implementação L-Atribuído Método

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

- Um esquema em **L-Atribuído**

$$\begin{aligned} S &\rightarrow \{ A_1.h = 1; \} A_1 \{ A_2.h = 1; \} A_2 \\ A &\rightarrow a \{ \text{print}(A.h) \} \end{aligned}$$

Implementação L-Atribuído Recursividade à Esquerda

- ▶ Analisadores descendentes
 - ▶ Não admitem recursividade à esquerda
- ▶ Supondo o esquema de tradução seguinte
 - ▶ Entrada: **3 4 5**

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

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

- ▶ Removendo a recursão a esquerda

$$A \rightarrow digit \quad \{ X.h = digit.lexval \} X \quad \{ A.val = X.s \}$$

$$X \rightarrow digit \quad \{ X_1.h = X.h + digit.lexval \} X_1 \quad \{ X.s = X_1.s \}$$

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

- ▶ Como fica a implementação descendente recursiva?

Implementação L-Atribuído

Removendo Recursão à Esquerda

- Considerando o esquema de tradução

$$\begin{array}{lcl} A & \rightarrow & A_1 Y \quad \{ A.a = \mathbf{g}(A_1.a, Y.y) \} \\ A & \rightarrow & X \quad \quad \{ A.a = \mathbf{f}(X.x) \} \end{array}$$

- Algoritmo que remove a recursão, temos

$$\begin{array}{lcl} A & \rightarrow & X R \\ R & \rightarrow & Y R \\ R & \rightarrow & \epsilon \end{array}$$

- Com ações semânticas

$$\begin{array}{lcl} A & \rightarrow & X \{ R.h = \mathbf{f}(X.x) \} R \{ A.a = R.s \} \\ R & \rightarrow & Y \{ R_1.h = \mathbf{g}(R.h, Y.y) \} R_1 \{ R.s = R_1.s \} \\ R & \rightarrow & \epsilon \{ R.s = R.h \} \end{array}$$

- R contém os atributos R.s e R.h

Implementação L-Atribuído Utilizando a Pilha

- Considerando o esquema de tradução

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

- Entrada **3 4 5** e uma Análise Preditiva Tabular com pilha

Implementação L-Atribuído Utilizando a Pilha

► Comportamento da Pilha e Entrada

Pilha	Entrada
\$ A	3 4 5 \$
\$ { A.val = X.s } X { X.h = digit.lexval } digit	3 4 5 \$
\$ { A.val = X.s } X { X.h = 3 }	4 5 \$
\$ { A.val = X.s } X	4 5 \$
\$ { A.val = X.s } { X.s = X ₁ .s } X ₁ { X ₁ .h = X.h + digit.lexval } digit	4 5 \$
\$ { A.val = X.s } { X.s = X ₁ .s } X ₁ { X ₁ .h = X.h + digit.lexval }	5 \$
\$ { A.val = X.s } { X.s = X ₁ .s } X ₁ { X ₁ .h = 3 + 4 }	5 \$
\$ { A.val = X.s } { X.s = X ₁ .s } X ₁	5 \$
\$ { A.val = X.s } { X.s = X ₁ .s } { X ₁ .s = X ₂ .s } X ₂ { X ₂ .h = X ₁ .h + digit.lexval } digit	5 \$
\$ { A.val = X.s } { X.s = X ₁ .s } { X ₁ .s = X ₂ .s } X ₂ { X ₂ .h = X ₁ .h + digit.lexval }	\$
\$ { A.val = X.s } { X.s = X ₁ .s } { X ₁ .s = X ₂ .s } X ₂ { X ₂ .h = 7 + 5 }	\$
\$ { A.val = X.s } { X.s = X ₁ .s } { X ₁ .s = X ₂ .s } X ₂	\$
\$ { A.val = X.s } { X.s = X ₁ .s } { X ₁ .s = X ₂ .s }	\$
\$ { A.val = X.s } { X.s = X ₁ .s } { X ₁ .s = X ₂ .s } { X ₂ .s = X ₂ .h }	\$

Implementação L-Atribuído – Ascendente?

- ▶ Uma análise **ascendente** em L-Atribuído? Possível?
- ▶ Problema
 - ▶ Atributos herdados vêm dos pais, que serão avaliados depois dos filhos, no momento da redução

$$A \rightarrow \{ B.h = f(A.h); \} B C$$

- ▶ Solução
 - ▶ Introduzir um **marcador** no lugar da ação semântica

$$A \rightarrow M B C$$

$$M \rightarrow \epsilon \{ M.h = A.h; M.s = f(M.h); \}$$

- ▶ Coordenar a presença de A.h e M.h (igual a B.h) na pilha

Exercício

► Considerando o esquema de tradução

```
E  →  T { R.h = T.ptr; } R { E.ptr = R.s; }
R  →  + T { R1.h = geraNo('+', R.h, T.ptr); } R1 { R.s = R1.s; }
R  →  - T { R1.h = geraNo('-', R.h, T.ptr); } R1 { R.s = R1.s; }
R  →  ε { R.s = R.h; }
T  →  ( E ) { T.ptr = E.ptr; }
T  →  id { T.ptr = geraFolha(id, id.nome); }
T  →  enum { T.ptr = geraFolha(num, num.val); }
```

► O que acontece quando temos a entrada $x - 2 + y$?

Implementação de Esquemas S-Atribuído

Implementação S-Atribuído

- ▶ Focado em analisadores **ascendentes**
 - ▶ Somente atributos sintetizados
 - ▶ Implementação
 - ▶ Atributos são mantidos na pilha de análise
 - ▶ No momento da redução
 - ▶ Atributos sintetizados da cabeça são calculados a partir dos atributos que estão na pilha (do corpo da produção)
 - ▶ Considerando o esquema de tradução
- $A \rightarrow X Y Z \quad \{ A.a = f(X.x, Y.y, Z.z); \}$
- ▶ Pilha

Análise	Atributo	
Z	Z.z	← Topo
Y	Y.y	
X	X.x	
...	...	

Implementação S-Atribuído – Exemplo

► Esquema de tradução de uma calculadora

L	→	E n	{ L.val = E.val }
E	→	E ₁ + T	{ E.val = E ₁ .val + T.val }
E	→	T	{ E.val = T.val }
T	→	T ₁ * F	{ T.val = T ₁ .val * F.val }
T	→	F	{ T.val = F.val }
F	→	(E)	{ F.val = E.val }
F	→	digit	{ F.val = digit.lexval }

► Possível implementação da calculadora

L	→	E n	{ print Atr(topo) }
E	→	E ₁ + T	{ att = Atr(topo-2) + Atr(topo) topo = topo-2; Atr(topo) = att; }
E	→	T	
T	→	T ₁ * F	{ att = Atr(topo-2) * Atr(topo) topo = topo-2; Atr(topo) = att; }
T	→	F	
F	→	(E)	{ att = Atr(topo-1) topo = topo-1; atr(topo) = att; }
F	→	digit	{ Atr(topo) = digit.lexval }

► Testar com a entrada 2*3+4n

Implementação S-Atribuído – Exemplo

Atributo	Pilha	Entrada	Produção
	\$	2 * 3 + 4 n	
	\$ 2	* 3 + 4 n	F → digit
2	\$ F	* 3 + 4 n	T → F
2	\$ T	* 3 + 4 n	
2	\$ T *	3 + 4 n	
2 _	\$ T * 3	+ 4 n	
2 _	\$ T * F	+ 4 n	F → digit
2 _ 3	\$ T	+ 4 n	T → T * F
6	\$ E	+ 4 n	E → T
6	\$ E +	4 n	
6 _	\$ E + 4	n	
6 _	\$ E + F	n	F → digit
6 _ 4	\$ E + T	n	T → F
6 _ 4	\$ E	n	
10	\$ E n		
10	\$ L		L → E n

Exercício 1

- O esquema de tradução dirigida pela sintaxe que segue **traduz** uma linguagem com terminais a, b e c em uma linguagem cujos terminais são 2, 3, 4, 5, e 6. Usando um analisador **ascendente** que executa as ações semânticas imediatamente após reduzir a regra correspondente, diga o resultado da tradução da entrada **aaadbc**?
- Esquema de Tradução

S	→	AB	{ print "1" }
S	→	AS	{ print "2" }
A	→	a	{ print "3" }
B	→	bC	{ print "4" }
B	→	dB	{ print "5" }
C	→	c	{ print "6" }

Exercício 2

- O esquema de tradução dirigida pela sintaxe que segue **traduz** uma linguagem com terminais a , b e c em uma linguagem cujos terminais são 2, 3, 4, 5, e 6. Usando um analisador **ascendente** que executa as ações semânticas imediatamente após reduzir a regra correspondente, diga o resultado da tradução da entrada **aaadbc**?
- Esquema de Tradução

S	\rightarrow	AB	$\{ S.\text{node} = \text{new node } (A.\text{node}, B.\text{node}); \}$
S	\rightarrow	AS_1	$\{ S.\text{node} = \text{new node } (A.\text{node}, S_1.\text{node}); \}$
A	\rightarrow	a	$\{ A.\text{node} = \text{new node } ("a"); \}$
B	\rightarrow	bC	$\{ B.\text{node} = \text{new node } ("b", C.\text{node}); \}$
B	\rightarrow	dB	$\{ B.\text{node} = \text{new node } ("d", B.\text{node}); \}$
C	\rightarrow	c	$\{ C.\text{node} = \text{new node } ("c"); \}$

Análise Semântica

Resumo

Análise Semântica – Sumário

- ▶ Esquemas de tradução
 - ▶ Regras de produção + ações semânticas
- ▶ Sem atributos herdados
 - ▶ Esquema S-Atribuído
 - ▶ Implementação fácil
 - ▶ Descendente, Ascendente
- ▶ Com atributos herdados
 - ▶ Esquema L-Atribuído
 - ▶ Implementação complexa
 - ▶ Descendente
 - ▶ Ascendente possível com mecanismos artificiais
- ▶ Resultado da Análise Semântica
 - ▶ Cálculo de valor de atributos
 - ▶ Expressões aritméticas, tipos de sequências de variáveis
 - ▶ Uma representação mais compacta de parte das entradas

Conclusão

- ▶ Leituras Recomendadas
 - ▶ Livro do Dragão
 - ▶ Seção 5.5
 - ▶ Série Didática
 - ▶ Seções 4.4 e 4.6

- ▶ Próxima Aula

Geração de Código Intermediário