

Compilers

by
Marwa Yusuf

Lecture 8-2
Tues. 13-4-2021

Chapter 2 (2.3)

Syntax-Directed Translation

Syntax Directed Translation

- $\text{expr} \rightarrow \text{expr}_1 + \text{term}$
 translate expr_1 ;
 translate term ;
 handle $+$;
- Attributes: any quantity attached to a programming construct.
- Translation scheme: Attaching program fragments to the productions of a grammar, executed during syntax analysis, producing a program translation.

Postfix Notation

- No parenthesis needed

- $E_1 \text{ op } E_2 \rightarrow E_1' E_2' \text{ op}$

$$(9-5)+2 \rightarrow 95-2+$$

$$9-(5+2) \rightarrow 952+-$$

- Scan from left to right

$$952+-3* \rightarrow 97-3* \rightarrow 23* \rightarrow 6$$

Synthesized Attributes

- Syntax directed definition associates:
 - Each grammar symbol \rightarrow attributes.
 - Each production \rightarrow semantic rules to compute attributes.

- Annotated parse tree.

- Synthesized vs. inherited attributes.

- $expr \rightarrow expr_1 + term$

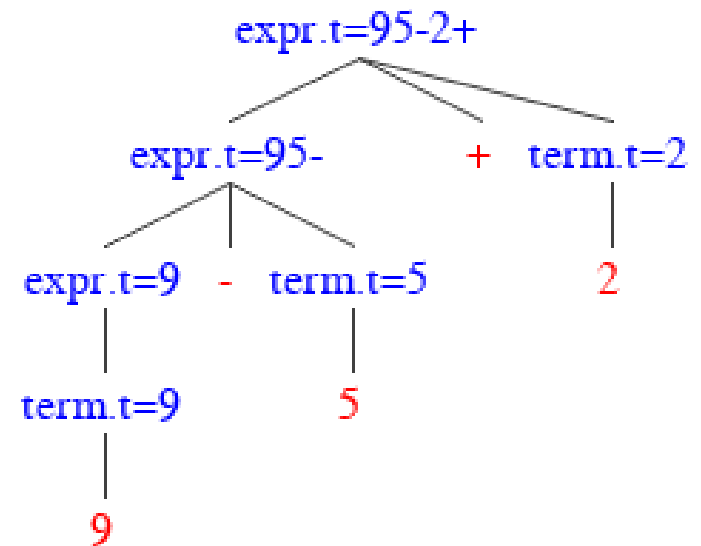
$$expr.t = expr_1.t \parallel term.t \parallel '+'$$

- $expr \rightarrow expr_1 - term$

$$expr.t = expr_1.t \parallel term.t \parallel '-'$$

- $expr \rightarrow term \quad expr.t = term.t$

- $term \rightarrow 0 \quad term.t = '0' \quad \dots \text{etc}$



Simple Syntax Directed Definitions

- String representing translation of the head = concatenation of translations of non-terminals of the body in the same order + optional additional string interleaved.

Tree Traversals

- Depth-first: visit children first.

```
procedure visit (node N) {  
    for ( each child C of N, from left to  
        right ) {  
        visit (C);  
    }  
    evaluate semantic rules at node N;  
}
```

- This is postorder (vs preorder).

Translation Schemes

- Attaching program fragments (semantic actions) to productions.

$rest \rightarrow term \{ \text{print}(' + ') \} rest_1$

- Position of semantic action in the production is significant.

- Ex:

$expr \rightarrow expr_1 + term \{ \text{print}(' + ') \}$

$expr \rightarrow expr_1 - term \{ \text{print}(' - ') \}$

$expr \rightarrow term$

$term \rightarrow 0 \{ \text{print}(' 0 ') \}$

$term \rightarrow 1 \{ \text{print}(' 1 ') \}$

...

$term \rightarrow 9 \{ \text{print}(' 9 ') \}$

