



Compiler Design

Preet Kanwal

Department of Computer Science & Engineering

Teaching Assistant : Sree Pranavi G

Compiler Design

Unit 3: Syntax Directed Translation

Preet Kanwal

Department of Computer Science & Engineering

In this lecture, you will learn about -

- Syntax Directed Translation
- Design of Translation Schemes
- Types of Translation Schemes
- Problematic SDT
- Postfix schemes

- The Principle of **Syntax Directed Translation** states that the meaning of an input sentence is related to its syntactic structure, i.e., to its Parse-Tree.
- Translations for programming language constructs guided by context-free grammars.
- There are 2 kinds of attributes - **Synthesized and Inherited**.
- An SDD with only synthesized attributes is an **S-attributed** definition.
- Every S-attributed SDD is also L-attributed.

Compiler Design

Recap - Syntax Directed Translation

- We associate **Attributes** to the grammar symbols representing the language constructs.
- Values for attributes are computed by **Semantic Rules** associated with grammar productions.
- There are two notations for attaching semantic rules:
 - **Syntax Directed Definitions** : High-level specification hiding many implementation details.
 - **Translation Schemes**: More implementation oriented, indicate the order in which semantic rules are to be evaluated.

Syntax Directed Translation Schemes

- **Translation Schemes** are more implementation oriented than syntax directed definitions since they indicate the order in which semantic rules and attributes are to be evaluated.
- A Translation Scheme is a context-free grammar in which,
 - Attributes are associated with grammar symbols.
 - Semantic Actions are enclosed between braces {} and are inserted within the right-hand side of productions.
- **Yacc uses Translation Schemes.**

Design of Translation Schemes

- When designing a Translation Scheme we must be sure that an attribute value is available when a semantic action is executed.
- When the semantic action involves only synthesized attributes, the action can be put at the end of the production.
- If we have an **L-Attributed SDD** we must enforce the following restrictions:
 - An **inherited attribute** for a symbol in the right-hand side of a production must be computed in an action before the symbol
 - A **synthesized attribute** for the non terminal on the left-hand side can only be computed when all the attributes it references have been computed - The action is usually put at the end of the production.

Implementation

- Ignoring the actions, parse the input and produce a parse tree as a result.
- Then, examine each interior node N , say one for production $A \rightarrow \alpha$. Add additional children to N for the actions in α , so the children of N from left to right have exactly the symbols and the actions of α .
- Perform a preorder traversal of the tree, and as soon as a node labelled by an action is visited, perform that action.

Infix to prefix example

$L \rightarrow En$

$E \rightarrow \{\text{printf}("+");\} E + T$

$E \rightarrow T$

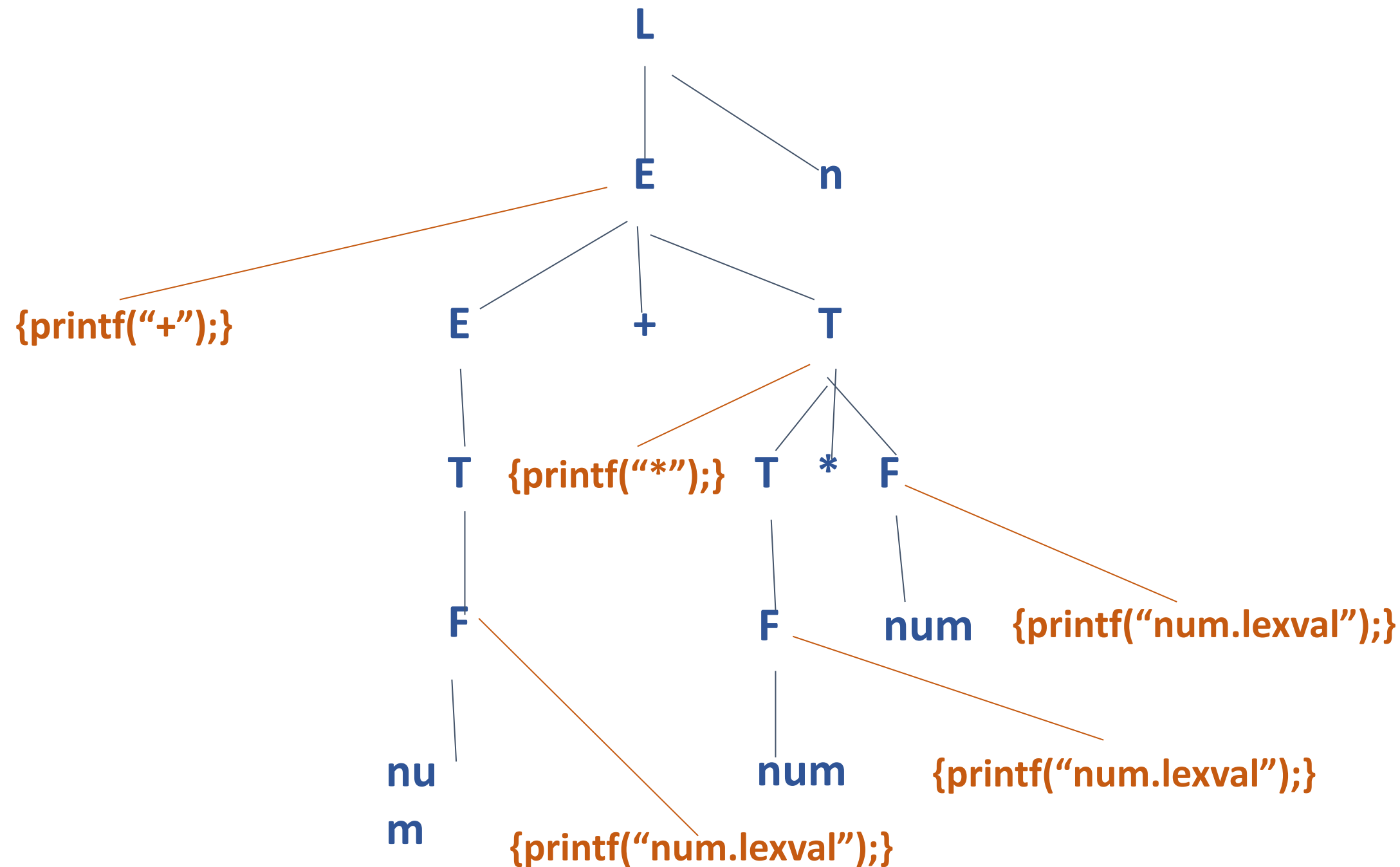
$T \rightarrow \{\text{printf}("*");\} T * F$

$T \rightarrow F$

$F \rightarrow (E)$

$F \rightarrow \text{num } \{\text{printf}(\text{"num.lexval"});\}$

Infix to prefix example (contd.)



What does the following SDT scheme print for **5 + 4 - 2**

$E \rightarrow TR$

$R \rightarrow +T \{ \text{print}("+"); \} R1$

$R \rightarrow -T \{ \text{print}("-"); \} R1$

$R \rightarrow \lambda$

$T \rightarrow F$

$F \rightarrow \text{num} \{ \text{print}(\text{num.lexval}); \}$

- S attributed to SDT.

Evaluation of S-attributed SDD

- S-attributed SDDs will have only synthesized attributes and can be evaluated by a bottom up parser.
- Since the attributes in the semantic actions are only synthesized, the actions can be placed at the end of the production.

Rules for evaluation

- Consider the following production,

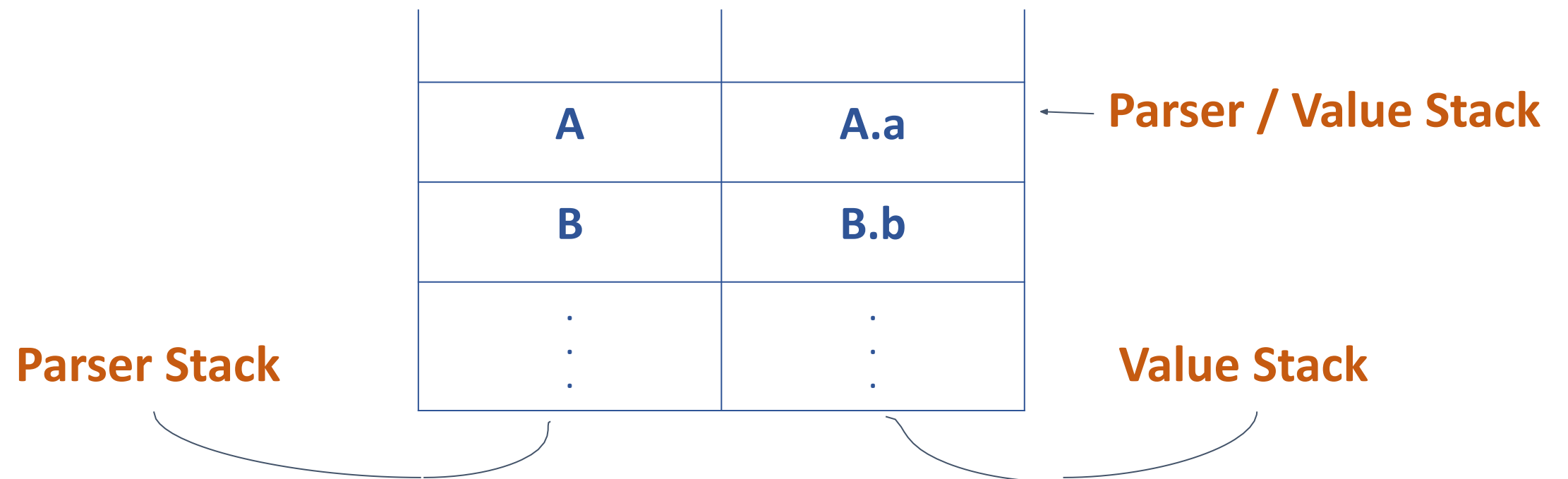
$A \rightarrow BCD$

before reducing BCD to A, the attributes of B, C and D must be computed before attribute of A which appears on the stack.

- Corresponding semantic action associated with the production must be executed.

Rules for evaluation

- The parser stack is extended to have parallel value stack.
- If the Action appears at the end of production in a SDT, such SDTs are called Postfix SDTs.





**THANK
YOU**

Preet Kanwal

Department of Computer Science & Engineering

preetkanwal@pes.edu