Compiler
--- Semantics Rules

Zhang Zhizheng
seu_zzz@seu.edu.cn
School of Computer Science and Engineering,
Software College
Southeast University

Attributes of Grammar Symbols I

- Denote the "real meaning" of grammar symbols, e.g.,
  - E $\rightarrow$E1$*$Digit has E.val=E1.val $\times$ Digit.val

    Digit.type=E1.type

    E.code=E1.code '$*$' Digit.code

Attributes of Grammar Symbols II

- Two kinds of attributes for nonterminals
  - **Synthesized**, whose value is defined only in terms of attributes values at the children and itself. E.g., val
  - **Inherited**, whose value is defined only in terms of attributes values at the parents, siblings and itself. E.g., type

Semantics Rules I

- Specify the evaluation of the value of the attribute
- One production is attached by several semantics rules.

➢ E.g., for E $\rightarrow$E1$*$Digit

E.val=E1.val $\times$ Digit.val

Digit.type=E1.type

E.code=E1.code '$*$' Digit.code

Semantics Rules II

- **Is Activated** when the host production is used in the reduction action.
- Is Used for type checking, intermediate code generation.

➢ E.g., for E $\rightarrow$E1$*$Digit

Digit.type=E1.type

E.code=E1.code ‘$*$’ Digit.code

Semantics Rules III

- Order the evaluation of attributions I

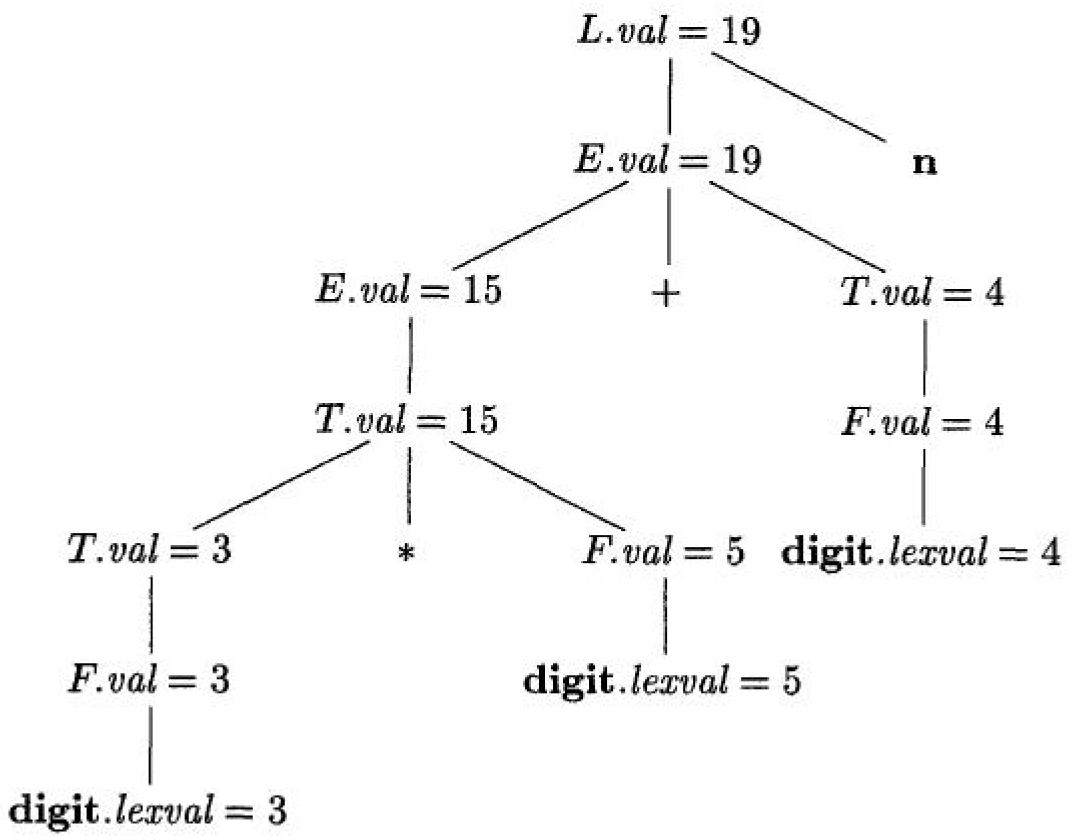
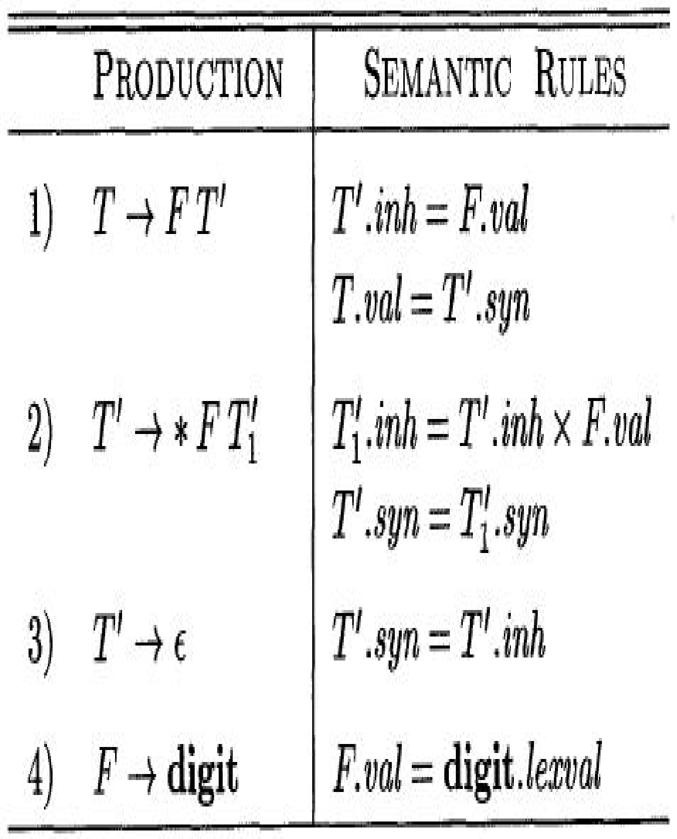


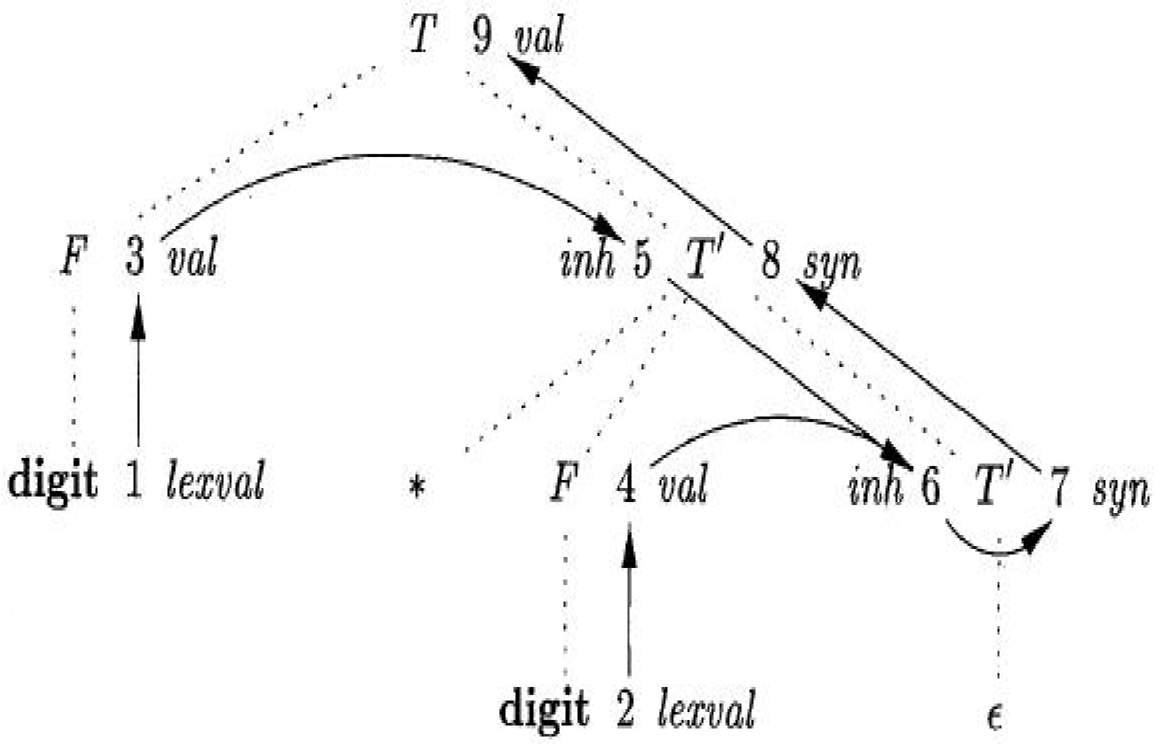
Figure 5.3: Annotated parse tree for $3 * 5 + 4\,\mathbf{n}$

Semantics Rules III

- Order the evaluation of attributions II

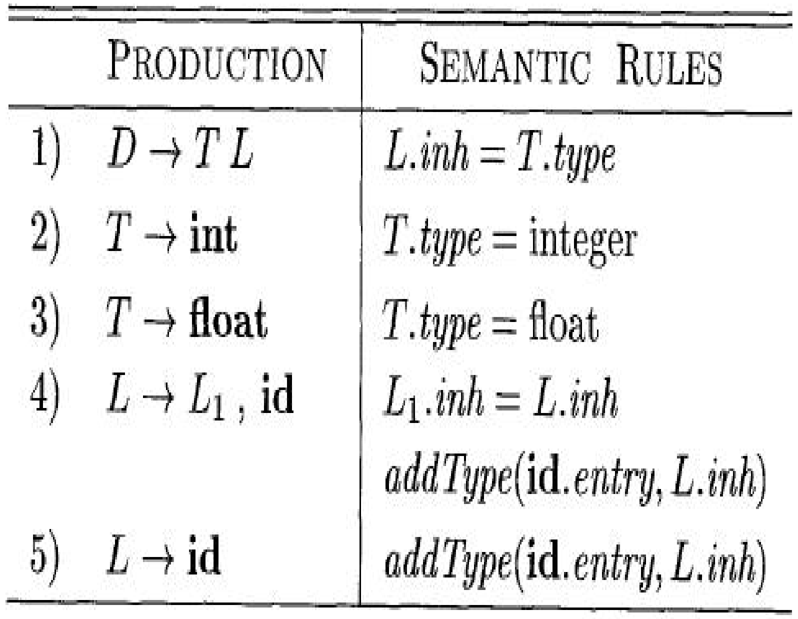


PRODUCTION	SEMANTIC RULES
1) $T \to F\, T'$	$T'.inh = F.val$ $T.val = T'.syn$
2) $T' \to *\, F\, T'_1$	$T'_1.inh = T'.inh \times F.val$ $T'.syn = T'_1.syn$
3) $T' \to \epsilon$	$T'.syn = T'.inh$
4) $F \to \mathbf{digit}$	$F.val = \mathbf{digit}.lexval$



Semantics Rules III

- Order the evaluation of attributions III



Production	Semantic Rules
1) $D \to T\ L$	$L.inh = T.type$
2) $T \to \mathbf{int}$	$T.type = \text{integer}$
3) $T \to \mathbf{float}$	$T.type = \text{float}$
4) $L \to L_1\ ,\ \mathbf{id}$	$L_1.inh = L.inh$
	$addType(\mathbf{id}.entry, L.inh)$
5) $L \to \mathbf{id}$	$addType(\mathbf{id}.entry, L.inh)$

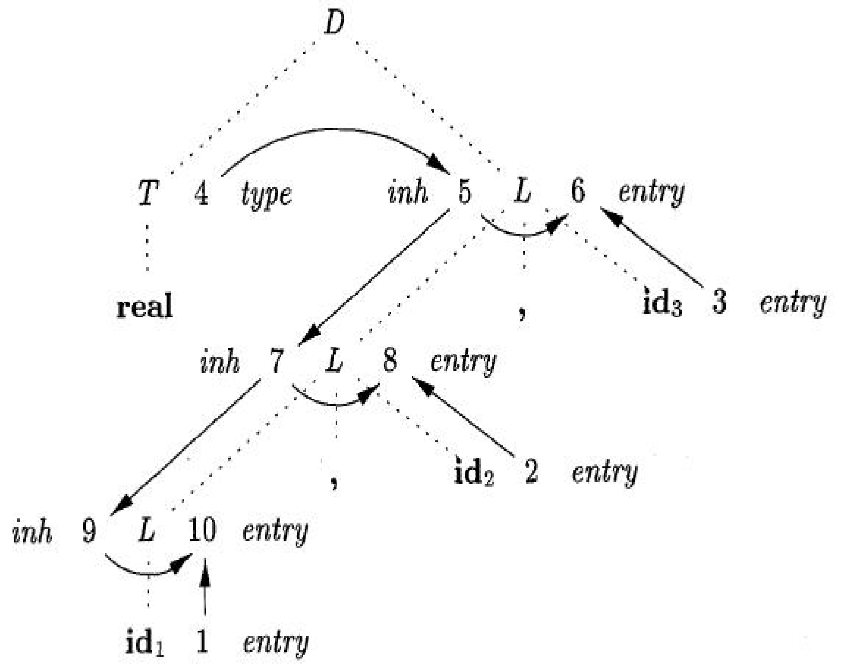


Figure 5.9: Dependency graph for a declaration **float** $\mathbf{id}_1$, $\mathbf{id}_2$, $\mathbf{id}_3$

Figure 5.8: Syntax-directed definition for simple type declarations

Syntax-directed Translation

- Because the semantics rule is activated when its associated production is used. Thus, the application of the semantics rule is syntax-directed.
- However, the order of evaluation of attributes value may not align the order of applying productions in parsing. E.g.,
  - Synthesized attributes fit bottom up.
  - Inherited attributes fit top down.

Written Assignment

- Please construct **an annotated parse tree** for the input string 4+(5*6+9)*7 where the syntax-directed definition is as following

Productions	Semantic Rules
$E \rightarrow E1 * T$	$E.val = E1.val * T.val$
$E \rightarrow T$	$E.val = T.val$
$T \rightarrow T1 + F$	$T.val = T1.val + F.val$
$T \rightarrow F$	$T.val = F.val$
$F \rightarrow (E)$	$F.val = E.val$
$F \rightarrow i$	F.val=i.**lexval**

Self Study

- Please self-study SDT and SDD according to the textbook