Semantic Analysis

Sudakshina Dutta

IIT Goa

1^{st} April, 2022

Attribute Translation Grammar

Two types of attributes at a parse-tree node N for non-terminal A are considered

- Synthesized attributes is defined only in terms of attribute values at the children of N and at N itself. Production must have A as its head.
  — Terminals can have synthesized attributes, not inherited attributes

  — Attributes for terminals have lexical values supplied by lexical analyzer; no semantic rules for terminals
- Inherited attribute is defined only in terms of the attribute values at N's parent, N itself and N's siblings. Production must have A as a symbol in its body.

- An attribute can be either synthesized or inherited, not both
- A symbol can have both the attributes
- A dependency graph depicts the order of evaluation of attributes
  - The graph has attributes as vertices
  - Edges depict order of evaluation
- Circularity is avoided

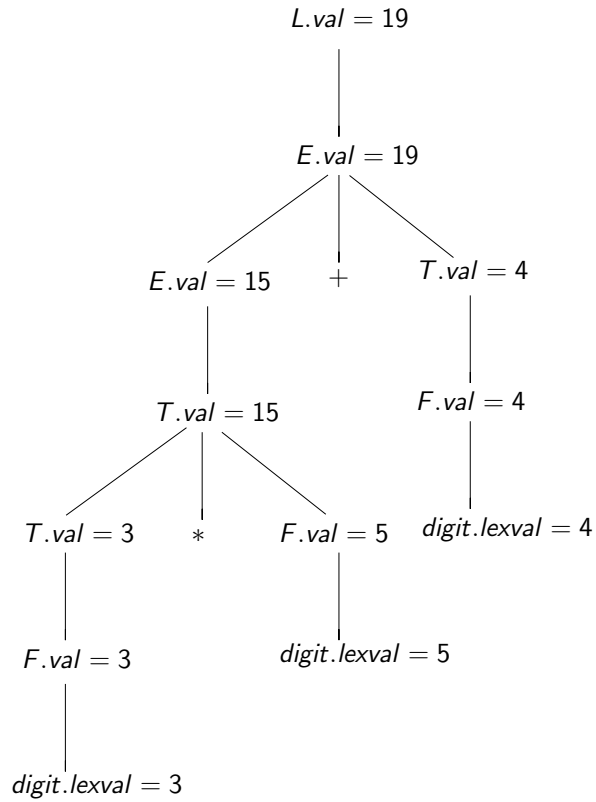
- A definition with only synthesized attributes is called $S-$attributed
- Can be combined with a LR parser e.g., Bison
- For grammars having only synthesized attributes, the evaluation of attributes can proceed in bottom-up manner

  — Post-order traversal
- For grammars with both inherited and synthesized attributes, there is no even order

Example - Synthesized attributes

- Bottom-up evaluation

PRODUCTION	SEMANTIC RULES
1) $L \rightarrow E$	$L.val = E.val$
2) $E \rightarrow E_1 + T$	$E.val = E_1.val + T.val$
3) $E \rightarrow T$	$E.val = T.val$
4) $T \rightarrow T_1 * F$	$T.val = T_1.val \times F.val$
5) $T \rightarrow F$	$T.val = F.val$
6) $F \rightarrow (E)$	$F.val = E.val$
7) $F \rightarrow digit$	$F.val = digit.lexval$



Note that:
T, T1
E, E1
are same.
Just for the sake of distinguishing while understanding the order of computation, we named differently.

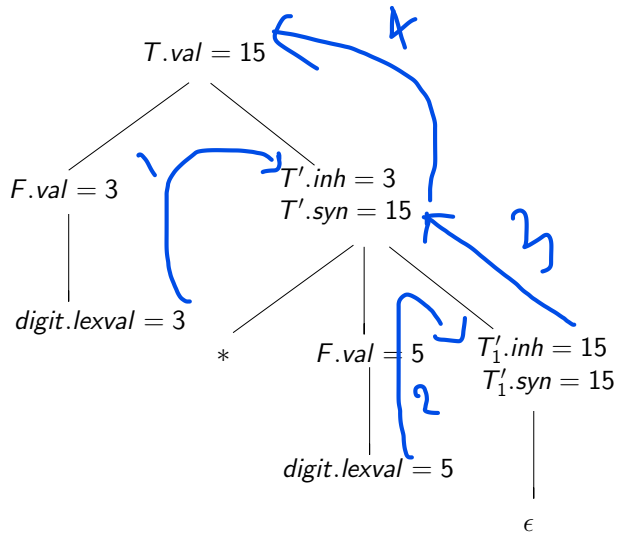
Synthesized and inherited attributes

- Inherited attributes are used to pass operands to the operator
- No particular evaluation order

Example - Synthesized and inherited attributes

Here also T' & T'1 are same.
But T & T' are different.

PRODUCTION	SEMANTIC RULES
1) $T \rightarrow FT'$	$T'.inh = F.val$ $T.val = T'.syn$
2) $T' \rightarrow *FT'_1$	$T'_1.inh = T'.inh \times F.val$ $T'.syn = T'_1.syn$
3) $T' \rightarrow \epsilon$	$T'.syn = T'.inh$
4) $F \rightarrow digit$	$F.val = digit.lexval$



We want to multiply two numbers 3 & 5 using the above grammar.