CSEN 1003: Compilers

Tutorial 9 - Semantic Analysis I

CFGs are not enough!

- The parse tree generated by the syntax analyzer is of minimal use to the task of semantic analysis.

CFGs are not enough!

- The parse tree generated by the syntax analyzer is of minimal use to the task of semantic analysis.
- Solution? Augment the CFGs with semantic rules.

CFGs are not enough!

- The parse tree generated by the syntax analyzer is of minimal use to the task of semantic analysis.
- Solution? Augment the CFGs with semantic rules.

Example

Production	Semantic Rule
$E \rightarrow E_1 + T$	$E.val = E_1.val + T.val$
$E \rightarrow T$	$E.val = T.val$
$T \rightarrow T_1 * F$	$T.val = T_1.val * F.val$
$T \rightarrow F$	$T.val = F.val$
$F \rightarrow$ **digit**	$F.val =$ **digit**.lexval

Today's Plan

1. Syntax-Directed Definitions
2. Syntax-Directed Translations
3. Recap

SDDs

- Symbols of the grammar have associated attributes.

SDDs

- Symbols of the grammar have associated attributes.
- Attributes carry information about the meaning of the program and how to translate it.

SDDs

- Symbols of the grammar have associated attributes.
- Attributes carry information about the meaning of the program and how to translate it.
- Computed by semantic rules.

Exercise 9-1

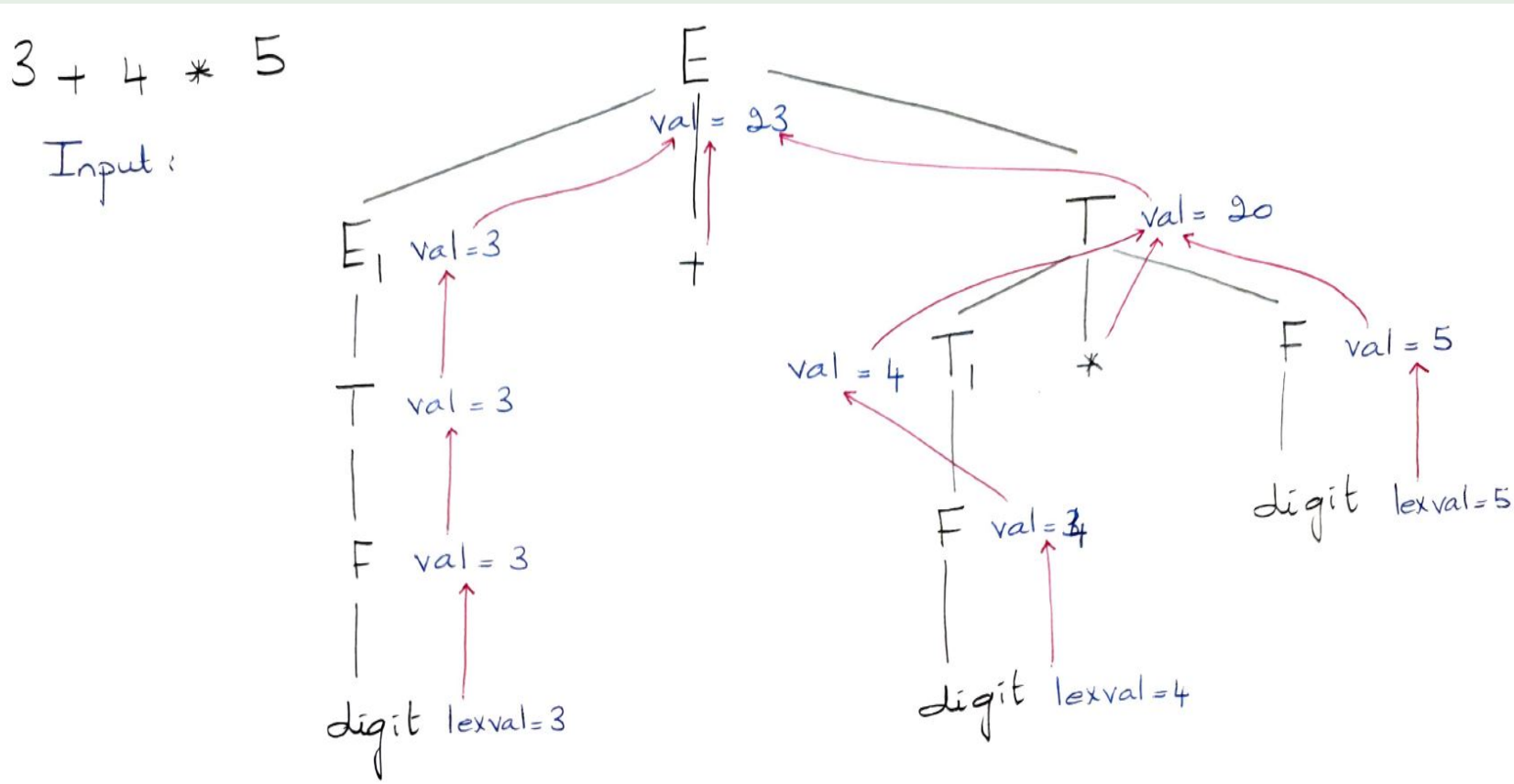
Example

Production	Semantic Rule
$E \rightarrow E_1 + T$	$E.val = E_1.val + T.val$
$E \rightarrow T$	$E.val = T.val$
$T \rightarrow T_1 * F$	$T.val = T_1.val * F.val$
$T \rightarrow F$	$T.val = F.val$
$F \rightarrow$ **digit**	$F.val =$ **digit**.lexval

Input: 3+4*5

Exercise 9-1

Example



Synthesized Attributes

- Attributes in the previous example can be computed in any bottom up traversal.

Synthesized Attributes

- Attributes in the previous example can be computed in any bottom up traversal.
- Attributes similar to the previous example are called synthesized attributes.

Synthesized Attributes

- Attributes in the previous example can be computed in any bottom up traversal.
- Attributes similar to the previous example are called synthesized attributes.
- An attribute of a node is synthesized if its value is determined from other attributes of itself or the attributes of its direct children.

Synthesized Attributes

- Attributes in the previous example can be computed in any bottom up traversal.
- Attributes similar to the previous example are called synthesized attributes.
- An attribute of a node is synthesized if its value is determined from other attributes of itself or the attributes of its direct children.
- Unfortunately, using synthesized attributes is not always possible.

Typical Example - Eliminating Left Recursion

Example

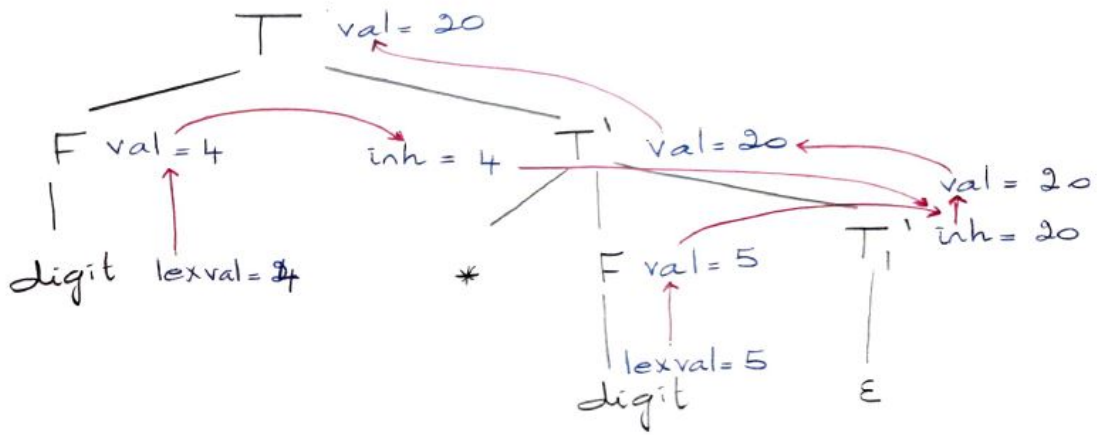
$T \rightarrow F\ T'$ $\quad T.val = T'.val \quad T'.inh = F.val$

$T' \rightarrow *\ F\ T_1'$ $\quad T_1'.inh = T'.inh * F.val \quad T'.val = T_1'.val$

$T' \rightarrow \varepsilon$ $\quad T'.val = T'.inh$

$F \rightarrow digit$ $\quad F.val = digit.lexval$

$4 * 5$



Typical Example - Eliminating Left Recursion

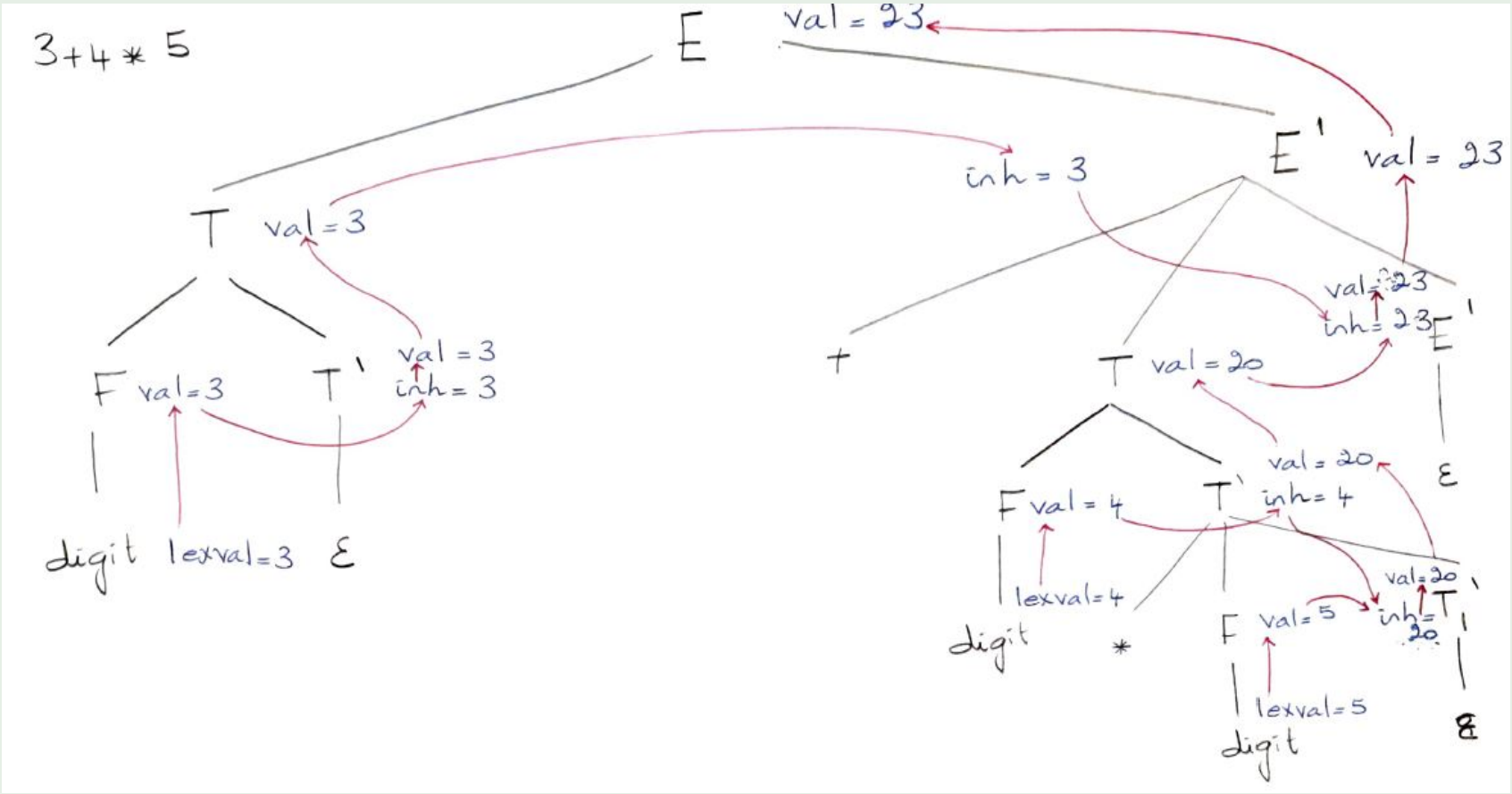
Example

Extend the previous SDD to handle the expressions given in 9-1.

Production	Semantic Rule
$E \rightarrow T\ E'$	$E.val = E'.val,\ E'.inh = T.val$
$E' \rightarrow +\ T\ E'_1$	$E'_1.inh = E.inh + T.val,\ E.val = E'_1.val$
$E' \rightarrow \varepsilon$	$E'.val = E'.inh$
$T \rightarrow F\ T'$	$T'.inh = F.val,\ T.val = T'.val$
$T' \rightarrow *\ F\ T'_1$	$T'_1.inh = T'.inh * F.val,\ T'.val = T'_1.val$
$T' \rightarrow \varepsilon$	$T'.val = T'.inh$
$F \rightarrow \mathbf{digit}$	$F.val = \mathbf{digit}.\mathrm{lexval}$

Typical Example - Eliminating Left Recursion

Example



Inherited Attributes and Order of Evaluation

- An attribute is inherited if its value is determined by its parent or sibling attributes.

Inherited Attributes and Order of Evaluation

- An attribute is inherited if its value is determined by its parent or sibling attributes.
- But how do we determine the order of evaluation when we have inherited attributes?

Inherited Attributes and Order of Evaluation

- An attribute is inherited if its value is determined by its parent or sibling attributes.
- But how do we determine the order of evaluation when we have inherited attributes?
  1. Construct the dependency graph.
  2. If the dependency graph is a DAG, an evaluation is valid. An order can be constructed by topological sorting.

Inherited Attributes and Order of Evaluation

- An attribute is inherited if its value is determined by its parent or sibling attributes.
- But how do we determine the order of evaluation when we have inherited attributes?
  1. Construct the dependency graph.
  2. If the dependency graph is a DAG, an evaluation is valid. An order can be constructed by topological sorting.
     Procedure: Use a queue
     a. Enqueue a node n with zero in-degree.
     b. Remove n from the graph together will all emanating arcs.
     c. Repeat a,b until the graph becomes empty.
     d. Start dequeuing from the queue. Evaluate a node when it is dequeued.

Some Special SDDs

Some Special SDDs

Definition

An SDD is *S-attributed* if all the attributes are synthesized.

Some Special SDDs

Definition

An SDD is *S-attributed* if all the attributes are synthesized.

Definition

An SDD is *L-attributed* if $\forall B \rightarrow \alpha A \beta$:

1. Every inherited attribute of A depends on inherited attributes of B and attributes of symbols of α.
2. Every synthesized attribute of B depends on $\alpha A \beta$ or on inherited attributes of B.

Exercise 9-4

Example

Determine if the following is consistent with the definition of S-attributed or L-attributed SDDs or neither. The grammar is $A \rightarrow BCD$.

1. $A.s = B.i + C.s$.

Exercise 9-4

Example

Determine if the following is consistent with the definition of S-attributed or L-attributed SDDs or neither. The grammar is $A \rightarrow BCD$.

1. $A.s = B.i + C.s$.
   L-attributed.

Exercise 9-4

Example

Determine if the following is consistent with the definition of S-attributed or L-attributed SDDs or neither. The grammar is $A \rightarrow BCD$.

1. $A.s = B.i + C.s$.
   L-attributed.
2. $A.s = B.i + C.s$ and $D.i = A.i + B.s$.

Exercise 9-4

Example

Determine if the following is consistent with the definition of S-attributed or L-attributed SDDs or neither. The grammar is $A \rightarrow BCD$.

1. $A.s = B.i + C.s$.
   L-attributed.
2. $A.s = B.i + C.s$ and $D.i = A.i + B.s$.
   L-attributed.

Exercise 9-4

Example

Determine if the following is consistent with the definition of S-attributed or L-attributed SDDs or neither. The grammar is $A \to BCD$.

1. $A.s = B.i + C.s$.
   L-attributed.
2. $A.s = B.i + C.s$ and $D.i = A.i + B.s$.
   L-attributed.
3. $A.s = B.s + D.s$.

Exercise 9-4

Example

Determine if the following is consistent with the definition of S-attributed or L-attributed SDDs or neither. The grammar is $A \to BCD$.

1. $A.s = B.i + C.s$.
   L-attributed.
2. $A.s = B.i + C.s$ and $D.i = A.i + B.s$.
   L-attributed.
3. $A.s = B.s + D.s$.
   S-attributed.

Exercise 9-4

Example

Determine if the following is consistent with the definition of S-attributed or L-attributed SDDs or neither. The grammar is $A \rightarrow BCD$.

1. $A.s = B.i + C.s$.
   L-attributed.
2. $A.s = B.i + C.s$ and $D.i = A.i + B.s$.
   L-attributed.
3. $A.s = B.s + D.s$.
   S-attributed.
4. $A.s = D.i$, $B.i = A.s + C.s$, $C.i = B.s$, and $D.i = B.i + C.i$.

Exercise 9-4

Example

Determine if the following is consistent with the definition of S-attributed or L-attributed SDDs or neither. The grammar is $A \to BCD$.

1. $A.s = B.i + C.s$.
   L-attributed.
2. $A.s = B.i + C.s$ and $D.i = A.i + B.s$.
   L-attributed.
3. $A.s = B.s + D.s$.
   S-attributed.
4. $A.s = D.i$, $B.i = A.s + C.s$, $C.i = B.s$, and $D.i = B.i + C.i$.
   Neither.

Exercise 9-4

Example

Determine if the following is consistent with the definition of S-attributed or L-attributed SDDs or neither. The grammar is $A \to BCD$.

1. $A.s = B.i + C.s$.
   L-attributed.
2. $A.s = B.i + C.s$ and $D.i = A.i + B.s$.
   L-attributed.
3. $A.s = B.s + D.s$.
   S-attributed.
4. $A.s = D.i$, $B.i = A.s + C.s$, $C.i = B.s$, and $D.i = B.i + C.i$.
   Neither.

Today's Plan

SDTs

- Pieces of code associated at various positions in the RHS of a production rule.

SDTs

- Pieces of code associated at various positions in the RHS of a production rule.
- The pieces of code generate the translation.

SDTs

- Pieces of code associated at various positions in the RHS of a production rule.
- The pieces of code generate the translation.
- Better suited for implementation.

SDTs

- Pieces of code associated at various positions in the RHS of a production rule.
- The pieces of code generate the translation.
- Better suited for implementation.
- To convert an L-attributed SDDs to SDTs:
  1. Insert the action that computes an inherited attribute immediately before its corresponding grammar symbol on the RHS.
  2. Insert the action that computes a synthesized attribute at the end of the rule.

Exercise 9-6

Example

Convert the following SDD to an SDT.

Production	Semantic Rule
$E \rightarrow T\ E'$	$E.val = E'.val,\ E'.inh = T.val$
$E' \rightarrow +\ T\ E'_1$	$E'_1.inh = E.inh + T.val,\ E.val = E'_1.val$
$E' \rightarrow \varepsilon$	$E'.val = E'.inh$

Exercise 9-6

Example

Convert the following SDD to an SDT.

Production	Semantic Rule
$E \rightarrow T\ E'$	$E.val = E'.val,\ E'.inh = T.val$
$E' \rightarrow +\ T\ E'_1$	$E'_1.inh = E.inh + T.val,\ E.val = E'_1.val$
$E' \rightarrow \varepsilon$	$E'.val = E'.inh$

$$
\begin{aligned}
E &\rightarrow T\ \{E'.inh = T.val\}\ E'\ \{E.syn = E'.syn\} \\
E' &\rightarrow +\ T\ \{E'_1.inh = E.inh + T.val\}\ E'_1\ \{E.syn = E'_1.syn\} \\
E' &\rightarrow \varepsilon\ \{E'.syn = E'.inh\}
\end{aligned}
$$

Today's Plan

Covered Topics

1. Syntax Directed Definitions.
2. Syntax Directed Translations.

Next Week: Semantic Analysis II