Semantic Analysis I

Tutorial (9)

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 CFG with semantic rules.
- Example:

$$
\begin{array}{ll}
E \rightarrow E_1 + T & \{E.val = E_1.val + T.val\} \\
E \rightarrow T & \{E.val = T.val\} \\
T \rightarrow T_1 * F & \{T.val = T_1.val + F.val\} \\
T \rightarrow F & \{T.val = F.val\} \\
F \rightarrow digit & \{F.val = digit.lexval\}
\end{array}
$$

Syntax Directed Definitions (SDDs)

- Syntax Directed Definitions:
  - 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.
  - Better suited for specification.

Exercise 9-1

$$
\begin{array}{ll}
E \to E_1 + T & \{E.val = E_1.val + T.val\} \\
E \to T & \{E.val = T.val\} \\
T \to T_1 * F & \{T.val = T_1.val + F.val\} \\
T \to F & \{T.val = F.val\} \\
F \to digit & \{F.val = digit.lexval\}
\end{array}
$$

Input: 3+4*5

- Here all attributes are synthesized.
- An attribute is synthesized if its value is determined from other attributes of itself or the attributes of its direct children.
- A post-order traversal will be appropriate to compute all attributes.

Synthesized attributes and Inherited attributes

- Sometimes, synthesized attributes are not sufficient.
- Typical Example: Left Recursion Elimination

$$
\begin{array}{ll}
T \to FT' & \{T.val = T'.val \\
 & \quad T'.inh = F.val\} \\
T' \to *FT'_1 & \{T'.val = T'_1.val \\
 & \quad T'_1.inh = T'.inh * F.val\} \\
T' \to \varepsilon & \{T'.val = T'.inh\} \\
F \to digit & \{F.val = digit.lexval\}
\end{array}
$$

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

Order of Evaluation

- How to determine the order of evaluation of the attributes if the SDD contains inherited attributes?
    a. Construct the dependency graph.
    b. If the dependency graph is a DAG, an evaluation is valid. An order can be constructed by topological sorting.

    Procedure: Use a queue

    1. Enqueue a node n with zero in-degree.
    2. Remove n from the graph together will all emanating arcs.
    3. Repeat 1,2 until the graph becomes empty.
    4. Start dequeuing from the queue. Evaluate a node when it is dequeued.

Exercise 9-2

Extend the below SDD to match the grammar in Exercise 9-1.

$$
\begin{aligned}
T \to FT' \quad & \{T.val = T'.val \\
& \ T'.inh = F.val\} \\
T' \to *FT'_1 \quad & \{T'.val = T'_1.val \\
& \ T'_1.inh = T'.inh * F.val\} \\
T' \to \varepsilon \quad & \{T'.val = T'.inh\} \\
F \to digit \quad & \{F.val = digit.lexval\}
\end{aligned}
$$

Some Special SDDs

- An SDD is S-attributed if all attributes are synthesized.
- An SDD is L-attributed if $\forall$ B$\rightarrow\alpha$Aβ:
  a. Every inherited attribute of A depends on inherited attributes of B and any attributes of symbols in α.
  b. Every synthesized attribute of B depends on αAβ or inherited attributes of B.
- Exercise 9-4: Suppose we have a production A$\rightarrow$BCD. Determine whether the following rules are consistent with the definition of S-attributed SDDs or L-attributed SDDs or neither.
  a. A.s = B.i+C.s
  b. A.s = B.i+C.s D.i = A.i+B.s
  c. A.s = B.s+D.s
  d. A.s = D.i B.i = A.s+C.s C.i = B.s D.i = B.i + C.i

Syntax Directed Translations (SDTs)

- Syntax Directed Translations:
  - 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.

From SDDs to SDTs

- Given an **L-attributed SDD**:
  a. Insert the action that computes an inherited attribute immediately before its corresponding grammar symbol on the RHS.
  b. Insert the action that computes a synthesized attribute at the end of the rule.
- Exercise 9-6: Apply this to the SDD in 9-2.

Writing SDDs

- Exercise 9-5: Write an SDD that converts binary numbers with decimal points to decimal numbers (base 10).