CS3005D Compiler Design

Winter 2024

Lecture #22

SDD for setting Types in Symbol Table, Synthesized/Inherited attributes

Saleena N

CSED NIT Calicut

March 2024

Symbol Tables

- Records information regarding the identifiers in the program
  - variable name - type, size and other relevant attributes
  - procedure name - number and types of its arguments, return type
- Entries created during lexical/syntax analysis phases

Setting type in Symbol Table

Production	Semantic Rules
$D \rightarrow T\ \mathbf{id}$	$addType(\mathbf{id}.entry,\ T.type)$
$T \rightarrow \mathbf{int}$	$T.type = int$
$T \rightarrow \mathbf{float}$	$T.type = float$

id.*entry* points to the Symbol Table entry for **id**

Setting type in Symbol Table

Production	Semantic Rules
$D \rightarrow T\ L$	$L.type = T.type$
$T \rightarrow$ **int**	$T.type = int$
$T \rightarrow$ **float**	$T.type = float$
$L \rightarrow L_1,$ **id**	$L_1.type = L.type,\ addType(\textbf{id}.entry,\ L.type)$
$L \rightarrow$ **id**	$addType(\textbf{id}.entry,\ L.type)$

Draw the annotated parse tree for $int\ id_1,\ id_2,\ id_3$

Synthesized / Inherited attributes

- Synthesized attribute: Attribute value at node N defined only in terms of attribute values at the children of N and at N itself.
- Inherited attribute: Attribute value at node N is defined only in terms of attribute values at N's parent, at N itself and N's siblings

Setting type in Symbol Table

Production	Semantic Rules
$D \rightarrow T\ L$	$L.type = T.type$
$T \rightarrow$ **int**	$T.type = int$
$T \rightarrow$ **float**	$T.type = float$
$L \rightarrow L_1,$ **id**	$L_1.type = L.type,\ addType(\mathbf{id}.entry,\ L.type)$
$L \rightarrow$ **id**	$addType(\mathbf{id}.entry,\ L.type)$

Attribute $T.type$ is *synthesized*

Attribute $L.type$ is *inherited*

Order of evaluation of attributes: Dependency Graph

Dependency Graph depicts the dependency among attributes.

- for each grammar symbol X, the graph has a node for each attribute associated with X
- a directed edge from attribute $X.a$ to attribute $Y.b$ to indicate that $Y.b$ is dependent on $X.a$ (value of $X.a$ is needed to compute value of $Y.b$)
  - synthesized attribute - dependency edge goes from child to parent.
  - inherited attribute - dependency edge goes from parent to child or from sibling to sibling
- dummy attributes corresponding to the application of functions (like $addType()$ in the example)

Dependency Graph

Production	Semantic Rules
$D \rightarrow T\ L$	$L.type = T.type$
$T \rightarrow$ **int**	$T.type = int$
$T \rightarrow$ **float**	$T.type = float$
$L \rightarrow L_1,$ **id**	$L_1.type = L.type,\ addType(\mathbf{id}.entry,\ L.type)$
$L \rightarrow$ **id**	$addType(\mathbf{id}.entry,\ L.type)$

Draw the dependency graph for $int\ id_1,\ id_2$

SDD: S-attributed / L-attributed

S-attributed SDD: involves only synthesized attributes

L-attributed SDD: attributes can be synthesized or inherited, but with the restrictions that dependency graph edges between attributes of symbols in a production body go from left to right

L-attributed SDD: Precise definition

Each attribute must be either

1. Synthesized, or
2. Inherited, but with the restriction that in a production $A \rightarrow X_1 X_2 \dots X_n$ any inherited attribute of X_i is computed using only
   - inherited attributes of A
   - either inherited or synthesized attributes of $X_1, X_2, \dots, X_{i-1}$
   - inherited or synthesized attributes of X_i such that there are no cycles in the dependency graph

Order of evaluation of attributes

- If there is an edge from attribute $X.a$ to attribute $Y.b$, evaluate $X.a$ before evaluating $Y.b$
- If the dependency graph has no cycles, attributes can be evaluated in a topological sort order of the graph.

Attributes: evaluation order

Production	Semantic Rules
$D \to \ T\ L$	$L.type = T.type$
$T \to \ \textbf{int}$	$T.type = int$
$T \to \ \textbf{float}$	$T.type = float$
$L \to \ L_1,\ \textbf{id}$	$L_1.type = L.type,\ addType(\textbf{id}.entry,\ L.type)$
$L \to \ \textbf{id}$	$addType(\textbf{id}.entry,\ L.type)$

Exercise:Draw the dependency graph for $int\ id_1,\ id_2$ and find a possible order of evaluation of attributes.

References

References:

- Aho A.V., Lam M.S., Sethi R., and Ullman J.D. Compilers: Principles, Techniques, and Tools (ALSU). Pearson Education, 2007.

Further reading:

- ALSU Chapter 2-sections 2.3, Chapter 5-section 5.1, 5.2