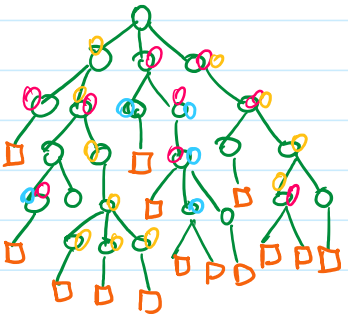
16 Operational Semantics

Monday, April 21, 2025 12:27 PM

- Main Idea
  - Let us add semantic information to our parse tree.
  - Add extra information to the intermediate nodes



- ATTRIBUTE GRAMMAR :

- expand the grammar with rules for semantic information
  - D. Knuth and D. Wagner

idea:
- each node will have attributes
- for each grammar symbol X a set of attributes Att(X)
- for each grammar rule R a collection of attribute rules that assign values to attributes of symbols in R

E.g. #1

Gramar Symbols A B C

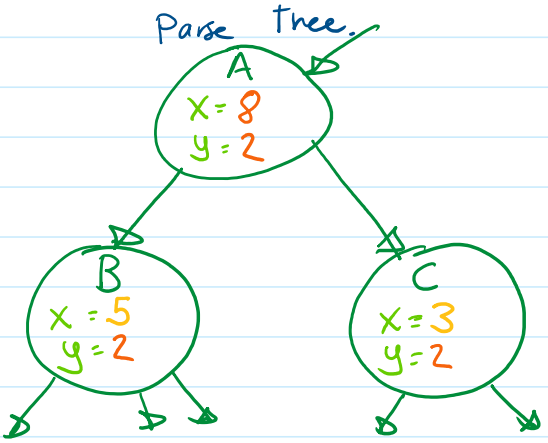
attributes x y

Grammar Rule: A → BC

Attribute rules:
- A.y ← 2
- A.x ← B.x + C.x
- B.y ← A.y
- C.y ← B.y

Synthesized: A.x A.y

Inherited: B.y C.y



- Types of Attributes

Synthesized Attributes
- value depends on the values of a nodes children

Synthesized Attributes
- value depends on the values of a node's children (info. flows from bottom to top)

Inherited Attributes
- value depends on the values of a node's parent or siblings. (info. flows from top to bottom or sideways)

E.g. #2

Attributes type exptype.

Grammar Rules:

$A \rightarrow$ var := E $\quad E.exptype \leftarrow A.type$

$E_0 \rightarrow E_1 + E_2$

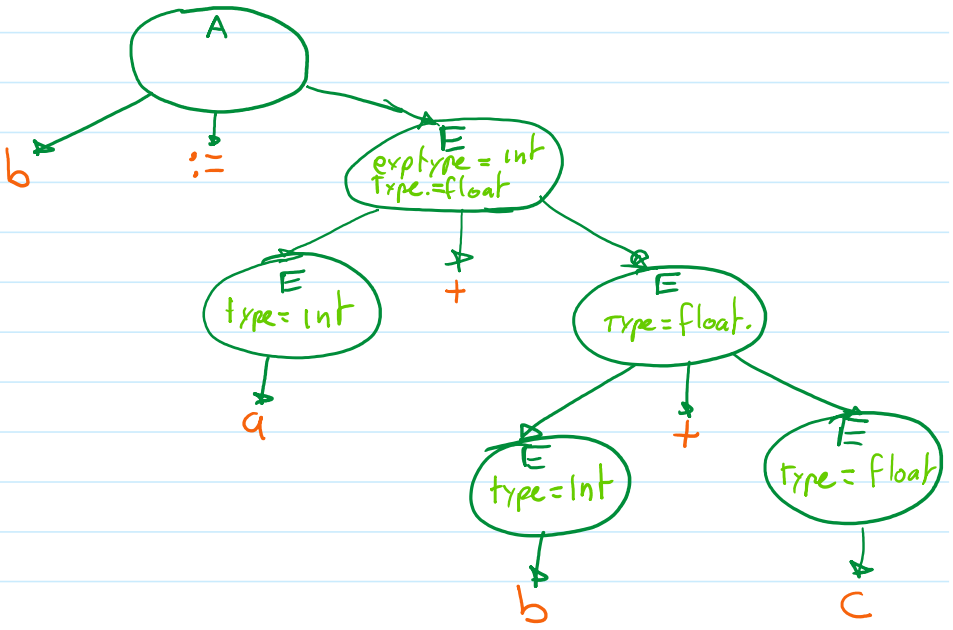
$$E_0.type = \begin{cases} int & \text{if } E_1.type = int \text{ AND } E_2.type = int \\ float & \text{otherwise} \end{cases}$$

$E \rightarrow$ var $\quad E.type = var.type.$

Symbol table

	type.
a	int
b	int
c	float.

b := a + b + c

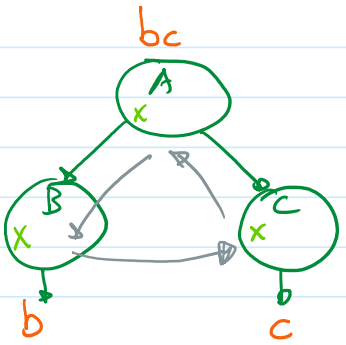


- COMPUTING ATTRIBUTE VALUES
  - Synthesized:

- COMPUTING ATTRIBUTE VALUES
  - Synthesized: traverse tree bottom-up.
  - Inherited: Traverse tree top-to-bottom
  - Both
    Multiple traversals bottom-up & top-to-bottom.

E.g. Degenerate attribute grammar.

$A \rightarrow BC$	$A.x \leftarrow C.x$
$B \rightarrow b$	$B.x \leftarrow A.x$
$C \rightarrow c$	$C.x \leftarrow B.x$



EOF