10/11/2022

Module 2: SYNTAX DIRECTED TRANSLATION

Syntax Direction Definitions: (SDD) ⇒ implemented in conjunction with LR Parser.

→ CFG with attributes and rules
→ Attributes - assoc. with grammar symbols
→ Rules - assoc. with productions
→ 2 types of attributes:
* Inherited attribute - for a non-terminal B at a parse-tree node N is defined by a semantic rule associated with the production at the parent of N.

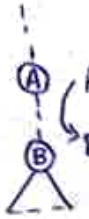
* Synthesized attribute - for a non-terminal A at a parse-tree node N is defined by a semantic rule associated with the production at N.

→ Synthesized attributes at node N can be defined in terms of inherited attribute values at node N.
→ Terminals can have synthesized attributes and cannot have inherited attributes
→ SDD with only synthesized attributes - S-attributed
→ SDD without side effects - attribute grammar.

Annotated PT:

→ A parse Tree showing values of its attributes is called annotated Parse Tree.
→ Circular dependcy: A → B

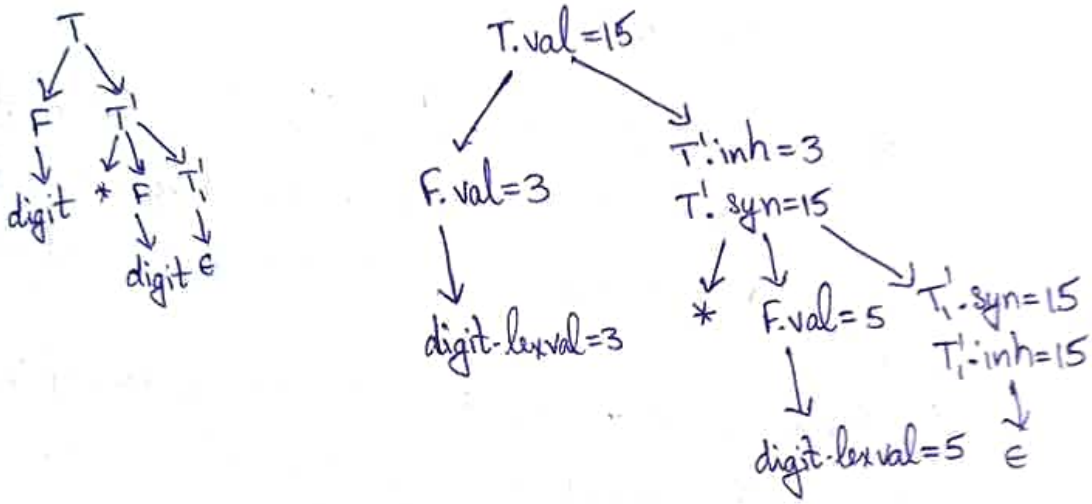
A.s = B.i
B.i = A.s + 1



Ex:

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

Annotated PT for 3*5 is:



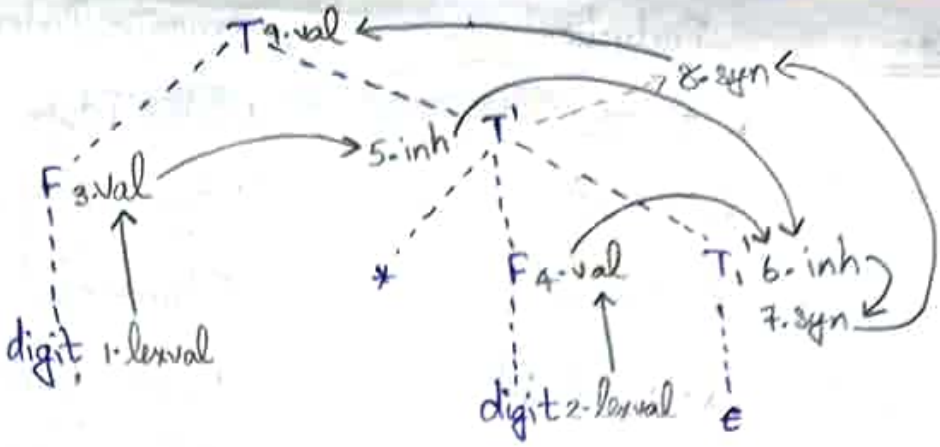
Evaluation order for SDD:

(i) Dependency Graph:

→ Depicts flow of information among attribute instances in a particular parse tree.

→ Edge from one attr. inst. to another means that the value of first needs to be computed before second.

→ Each attribute assoc. with a node X in Parse Tree is a node in Dep. Graph.



(ii) Linear ordering: Topological sort (without cycles)

{Here, 135246789}

(iii) S-Attributed Definitions:

→ No cycles in dep. graph.

→ Evaluate in postorder sequence.

(iv) L-Attributed Definitions:

→ Each attribute must be either

1. Synthesized, or
2. Inherited with rules:

$A \to X_1 X_2 \dots X_n$, inherited attr. X_i - assess

(a) inh. attr. assoc. with A

(b) inh./syn. attr. assoc. with $X_1 X_2 \dots X_{i-1}$

(c) inh./syn. attr. assoc. with X_i itself such that there are no cycles in dep. graph.

Semantic Rules with controlled side-effects:

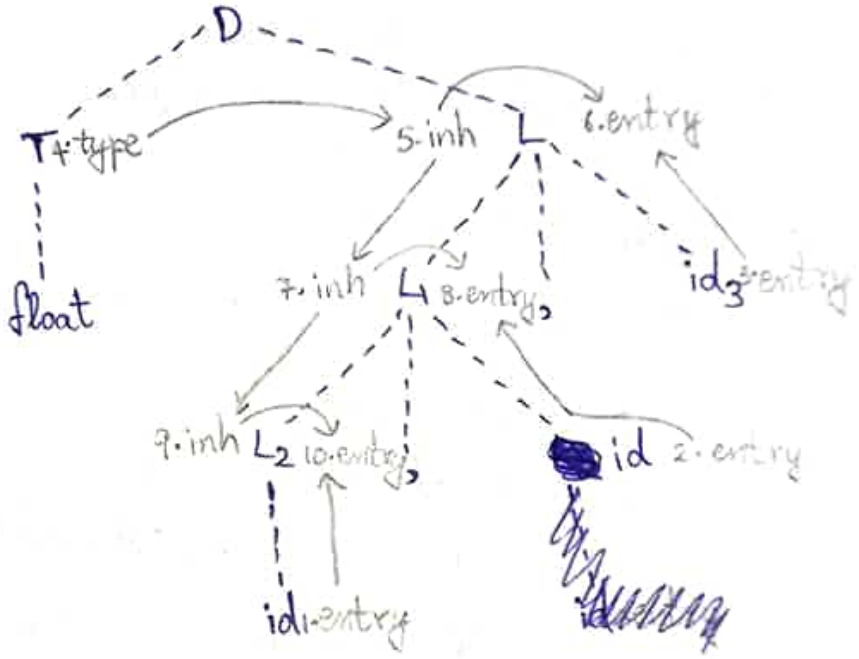
→ Attr. grammar - no side effects, allow any eval. order consistent with dep. graph.

→ Translation scheme - left-to right eval., allow semantic actions to contain any pgm-fragment

Ex:

Production	Semantic Rules
1. D → TL	L.inh = T.type
2. T → 'int'	T.type = integer
3. T → float	T.type = float
4. L → L1, id	L1.inh = L.inh addType (id.entry, L.inh)
5. L → id	addType (id.entry, L.inh)

(Dep.graph) APT for float id_1, id_2, id_3



Applications of Syntax Directed Translations:

→ Type checking, intermediate code generation

→ Const. of syntax tree. (Refer book Pg: 318 to 320).

Structure of a type:

→ Grammar for basic type/array type:

Production	Semantic Rules
1. T → BC	T.t = C.t C.b = B.t

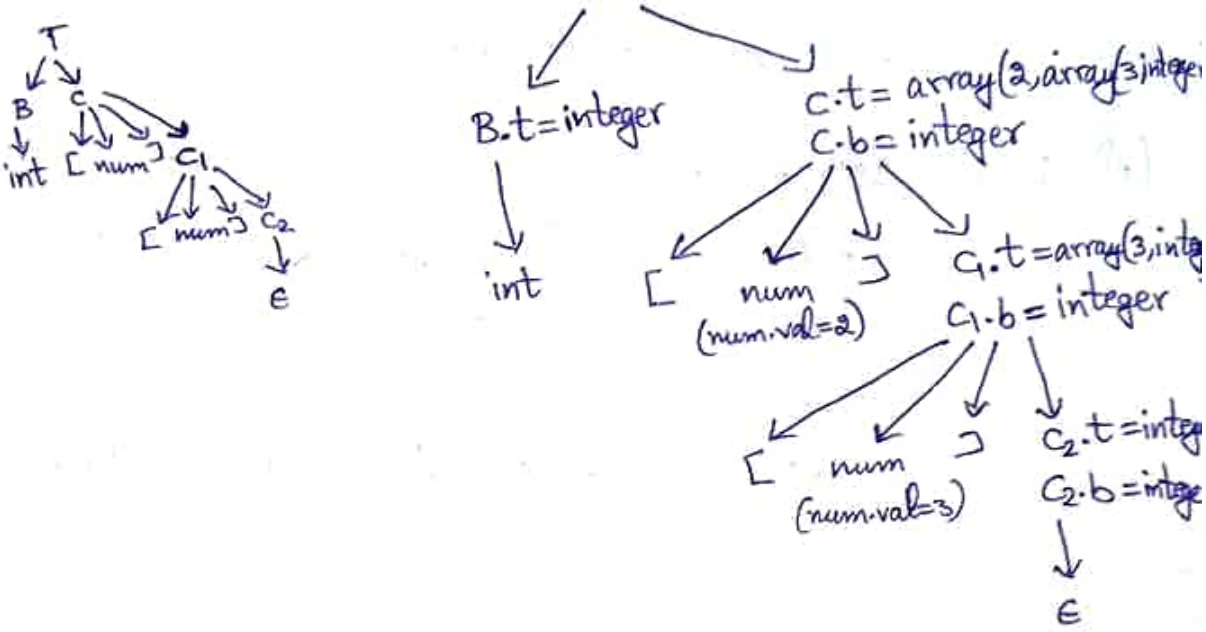
2. $B \rightarrow int$ $\quad B.t = integer$

3. $B \rightarrow float$ $\quad B.t = float$

4. $C \rightarrow [num]\, C_1$ $\quad C.t = array(num.val, C_1.t)$
$\quad C_1.b = C.b$

5. $C \rightarrow \epsilon$ $\quad C.t = C.b$

→ SDT for int [2][3]:



Syntax Directed Translation Schemes (SDT):

→ CFG with pgm fragments embedded within production bodies.

→ SDT's with all actions at right end of production bodies are called postfix SDT's.

→ SDT with action inside productions:

$B \rightarrow X\{a\}Y$ $\quad$ a is performed

- → Bottom up Parser – as soon as X appears on top of parsing stack.
- → Top down Parser – before expanding this occ. of Y.

Implementation:

(i) L-attr. SDD by Recursive Descent Parsing:

- inh. attr. - arguments of functions for nonterminals
- syn. attr. - returned by functions

(ii) L-attr. SDD on LL Grammar:

- Records to hold syn. attr. for non-terminal are place below NT on stack.
- inh. attr. for a NT - stored with NT on stack.
- Action records are also placed on stack.

(iii) L-attr. SDD on an LL Grammar, Bottom Up:

- Marker non-terminals appears on Bottom Up Parser's stack
  - inh. attr. of NT - above it on stack
  - syn. attr. - are kept with NT on stack.