Syntax-Directed translation

Syntax-directed definition: (SDD)

Production	semantic rule.
$A \rightarrow S ; A_1$	$A.code = S.code \parallel A_1.code$

nonterminals have abitrary attributes

$$\underset{\text{nonterminal}}{A}.\ \underset{\text{attribute}}{type}$$

Syntax-Directed Translation Schemes:

allows you to put actions in the middle of grammar

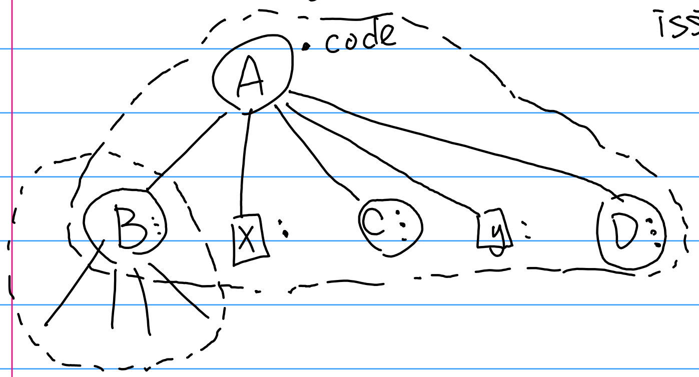
$$A \rightarrow \overbrace{\underbrace{\{\ \text{print "hello world"};\ \}}_{\text{semantic action}}}^{N} S ; \overbrace{\underbrace{\{\text{print "newS"}\}}}^{M} A$$

$A \rightarrow NS ; MA$

$N \rightarrow \varepsilon$ ≡ print "hello world" ;

$M \rightarrow \varepsilon$ ≡ print "newS" ;

$A \rightarrow BxCyD$



locality issues ?

Rule

SYNTHESIZED : just going up

→ attributes at the same node or children node

A.attr = ∼
A.∼
B.∼
C.∼
D.∼

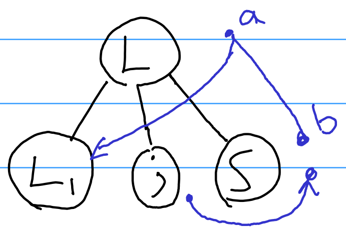
inherited attribute : can go down or between the sublings

C.attr = ∼
A.∼
B.∼
C.∼
D.∼

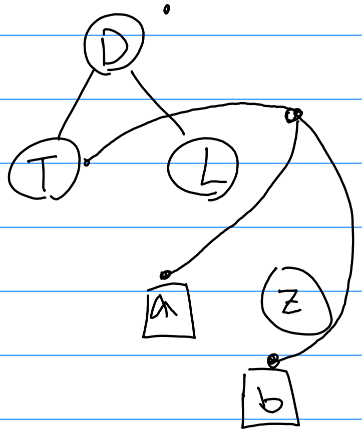
An SDD (for any grammar) is called S-attributed if all attributes are synthesized. → easy to evaluate.

$L \to L_1 ; S$ $\quad$ $L.a = S.b$

$S.b = L_1.a + 1$



Dependency Graph edge from S.b to L.a if L.a = f(S.b)



$V \rightarrow \{ L \}$
$L \rightarrow S$
$L \rightarrow S\ L$

$S \rightarrow \underline{return}$
$S \rightarrow \underline{if}\ \underline{(}\ cond\ \underline{)}\ S$
$S \rightarrow \underline{while}\ \underline{(}\ cond\ \underline{)}\ \underline{\{}$

$V = \{ L \}$ — $\begin{cases} v.returnTarget = newLabel() \\ L.returnTarget = v.returnTarget \end{cases}$