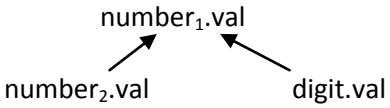
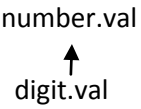
Dependency Graph

- It represents the flow of information among the attributes in a parse tree.
- It is useful for determining order for attributes in the parse tree (How are the values computed).

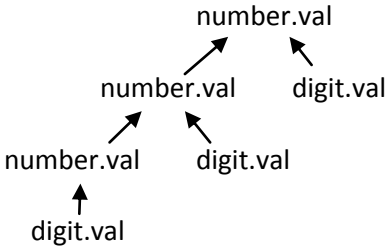
$number_1.val = number_2.val * 10 + digit.val$



number.val = digit.val



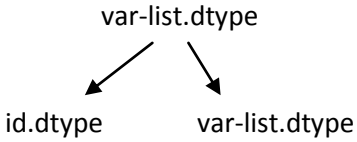
String: 345



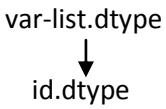
decl → type var-list example: int a, b, c;
type → **int** | **float**
var-list → **id**, var-list | **id** example: int a, b, c; int a;

Grammar Rule	**Semantic Rule**
decl → type var-list	var-list.dtype = type.dtype
type → **int**	type.dtype = integer
type → **float**	type.dtype = real
$var\text{-}list_1$ → **id**, $var\text{-}list_2$	**id**.dtype = $var\text{-}list_1$.dtype
	$var\text{-}list_2$.dtype = $var\text{-}list_1$.dtype
var-list → **id**	**id**.dtype = var-list.dtype

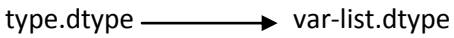
Production: $var\text{-}list_1$ → **id**, $var\text{-}list_2$
id.dtype = $var\text{-}list_1$.dtype
$var\text{-}list_2$.dtype = $var\text{-}list_1$.dtype



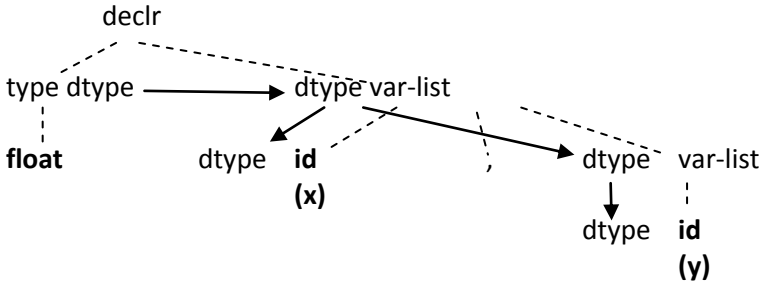
Production: var-list → **id**
id.dtype = var-list.dtype



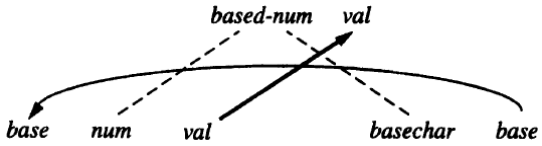
Production: decl → type var-list
var-list.dtype = type.dtype



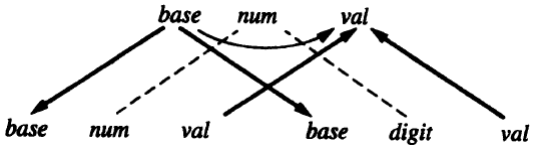
Superimpose dependency graph on the parse table
Expression: float x,y



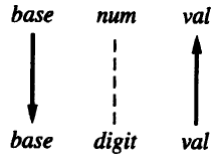
based-num → num basechar



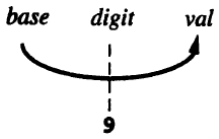
num → num digit



num → digit



digit → 9



Attribute Grammar:

Grammar Rule	Semantic Rule
based-num → num basechar	based-num.val = num.val num.base = basechar.base
basechar → **o**	basechar.base = 8
basechar → **d**	basechar.base = 10
num_1 → num_2 digit	num_1.val = if digit.val=error or num_2.val=error then error else num_2.val * num_1.base + digit.val num_2.base = num_1.base digit.base = num_1.base
num → digit	num.val = digit.val digit.base = num.base
digit → **0**	digit.val = 0
digit → **1**	digit.val = 1
digit → **2**	digit.val = 2
digit → **3**	digit.val = 3
digit → **4**	digit.val = 4
digit → **5**	digit.val = 5
digit → **6**	digit.val = 6
digit → **7**	digit.val = 7
digit → **8**	digit.val = **if** digit.base=8 **then** error **else** 8
digit → **9**	digit.val = **if** digit.base=8 **then** error **else** 9

Production: num → digit
num.val = digit.val
digit.base = num.base

String: 345o