

SYNTAX DIRECTED DEFINITION FOR DECLARATION STATEMENT

In translation scheme of fig.1 nonterminal P generates a sequence of declarations of the form id:T Syntax directed definition is used in putting information to the symbol table.

Terms used in SDD is listed below.

- 1. offset – it is initially set to 0 before any declaration.
- 2. proc enter(name, type, offset) creates an entry for the name in the symbol table.
- 3. type and width are the synthesized attributes of the non terminal

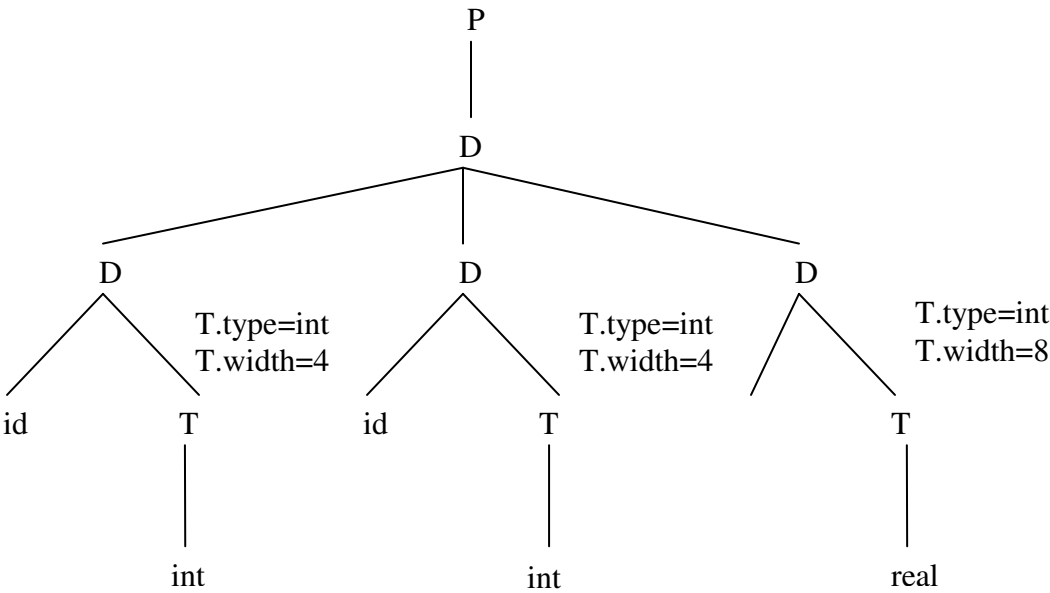
| Productions | Semantic rules |
|-------------|---|
| P -> | {offset = 0} |
| D | |
| D -> D ; D | |
| D -> id : T | {enter(id.name, T.type, offset) offset = offset + T.width} |
| T->integer | T.type = integer ; T.width = 4} |
| T->real | {T.type = real; T.width = 8} |

Fig: Computing the types and relative addresses of declared names

Example:

Sequence of declarations

- 1. var a: integer
- 2. var b: integer
- 3. var c: real



Traverse the parse tree for the declaration statement from the leaves, id.name = a, non terminal T is associated with the attributes whose SDD is given in the Fig. 1. T-> integer production corresponds to setting the T.type = integer and width = 4. considering the parent of T, procedure enter() is called to enter the id.name which is a, T.type which is integer, offset which is 0 into the symbol table. offset is also updated by adding offset with T.width. Like the above for the other two declarations also entries will be made in the symbol table and offset is updated.

| Symbol name | Type | Offset |
|-------------|---------|--------|
| a | integer | 0 |
| b | Integer | 4 |
| c | Real | 8 |