The Structure of a Type

AST: Abstract Syntax tree

int [2] [3]


```
Class Cell {
int info; cell next;...}
```

Types

$$
\begin{array}{lcl}
D & \rightarrow & T\ \mathbf{id}\ ;\ D \mid \epsilon \\
T & \rightarrow & B\ C \mid \mathbf{record}\ \text{'}\{\text{'}\ D\ \text{'}\}\text{'} \\
B & \rightarrow & \mathbf{int} \mid \mathbf{float} \\
C & \rightarrow & \epsilon \mid [\ \mathbf{num}\ ]\ C
\end{array}
$$

1. boolean
   char
   float
   void
2. Array
3. Node
4. Record

int [num] [num] x;

float d;

SDT for variable width and offset calculation

$P \rightarrow$ D		$\{ offset = 0; \}$
$D \rightarrow T\ \mathbf{id}\ ;\ D_1$		$\{ top.put(\mathbf{id}.lexeme, T.type, offset);$ $offset = offset + T.width; \}$
$D \rightarrow \epsilon$		
$T \rightarrow B\ C$		$\{ t = B.type; w = B.width; \}$
$B \rightarrow \mathbf{int}$		$\{ B.type = integer; B.width = 4; \}$
$B \rightarrow \mathbf{float}$		$\{ B.type = float; B.width = 8; \}$
$C \rightarrow \epsilon$		$\{ C.type = t; C.width = w; \}$
$C \rightarrow [\ \mathbf{num}\ ]\ C_1$	C.type=	$\{ array(\mathbf{num}.value, C_1.type);$ $C.width = \mathbf{num}.value \times C_1.width; \}$

T.type=C.type;T.width=c.width

t=int;
w=4

offset =8

Input: float x;
int [2] [3] y;