



Mar 2-11:04 AM

Recall: attributes are stored on the stack ("semantic stack")

Declaring a type for the semantic stack:

In preamble of parse.y:

```

%union {
  int yint;
  TR ytree;
}

%token {int} CONST
%token {char} VAR
%type {struct tnode} expr, term, factor

%%
expr : expr '+' term
    { $$ytree = makebinopk('%', $1.ytree, $3.ytree); }

factor : CONST
    { $$ytree = makeconstk($1.yint); }

%%
expr : expr '*' term
    { $$ytree = makebinopk('%', $1.ytree, $3.ytree); }

factor : CONST
    { $$ytree = makeconstk($1); }

In preamble:
%include "tree.h"
%}

```

including struct typedefs for TR nodes and attributes for attributes, etc.

Mar 2-11:28 AM

TR newnode, #include <assert.h>

```

newnode = (TR) malloc(sizeof(struct tnode));
assert(newnode != NULL);
newnode->tag = BINOP;
newnode->u.binop.op = '+';
newnode->u.binop.left = leftarg;
... right ...

```

Project:
tree evaluation:
int eval (TR tree)
{ int left, right; switch (tree->tag) { case CONST: return tree->u.const; case VAR: return getval(tree->u.var); case BINOP: leftval = eval(tree->u.binop.left); rightval = eval(tree->u.binop.right); ... } }

Project: Pascal compiler
Terms
3 instructions:
1. variable declarations & static storage allocation
2. Expressions, produce a few declarations, assignment statements.
3. Control flow: if, loops, etc.

Mar 2-11:48 AM

Pascal program:

```

program Hello ;
type
  var
    x : Integer;
    foo : Boolean;
  | fens & procedures
begin
  ===
end.

```

Mar 2-12:11 PM