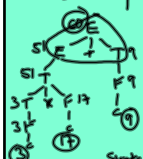
3*17+9 — Annotating a parse tree

Simple calculator: attribute "val" of the (sub)expression

Syntax directed rules: define attributes of one node based on attrs of other nodes in the parse tree.

E → E1+T	E.val := E1.val + T.val
E → T	E.val := T.val
T → T1*F	T.val := T1.val * F.val
T → F	T.val := F.val
F → c	F.val := c.val
F → (E)	F.val := E.val



Leaves (tokens) have val attrs (provided by the lexical analyzer).

Syntax-directed definition

Synthesized attribute: depends only on synthesized attributes of children (or a leaf attr).

These are the attrs that can be computed by traversing the tree bottom-up.

Inherited attribute: an attr that is not synthesized

bison Source file <name>.y (eg gram.y parse.y)

contains a syntax-directed translation scheme

annotate the grammar with actions ({ code }) which compute attributes and do other things (side-effects)

Typical Rule:

```
expr : expr '+' term { $$ = $1 + $3; }
     | term          { $$ = $1; }
     ;
```

happens by default

{ } — empty action (nothing done)

```
term : term '*' factor { $$ = $1 * $3; }
     | factor          /* default action */
     ;

factor : CONST
       | '(' expr ')' { $$ = $2; }
```

Preamble

```
%{
#include <...>
#include "..."
%}
// declare named tokens
%token CONST

%%
(Rules)

%%
Utilities
main()
{
    :
    yyparse();
    :
}

int yylex()
{
// Returns token type
// (integer code)
// sets global variable yylval
// to the attribute (int by default)
    int c;
    while((c=getchar())==' '
          || c=='\t');
    if (!isdigit(c)) // <expr>
        return c; // no variables
    // c is a decimal digit
```

yyparse(); ← the actual parse

main() ← for stand-alone parse

int yylex() ← if stand-alone (if lex is used to produce yylex() then don't need this)

Feb 9-11:01 AM

```
// c is a digit
ungetc(c);
scanf("%d", &yylval);
return CONST;
}
```

return type of yylex()

%token CONST
%token VAR

enum type
CONST ← 256
VAR ← 257

0-255 are reserved for literal one-character tokens '...'

combining bison with flex:

In flex file:

```
[0-9]+      { yylval = atoi(yytext);
              return CONST;
            }
```