

9 BISON

Monday, March 3, 2025

12:28 PM

- SHIFT-REDUCE PARSER

BISON :- an implementation.

input :- Grammar specification

output :- Source Code the corresponding shift-reduce parser

- BISON is designed to work with FLEX

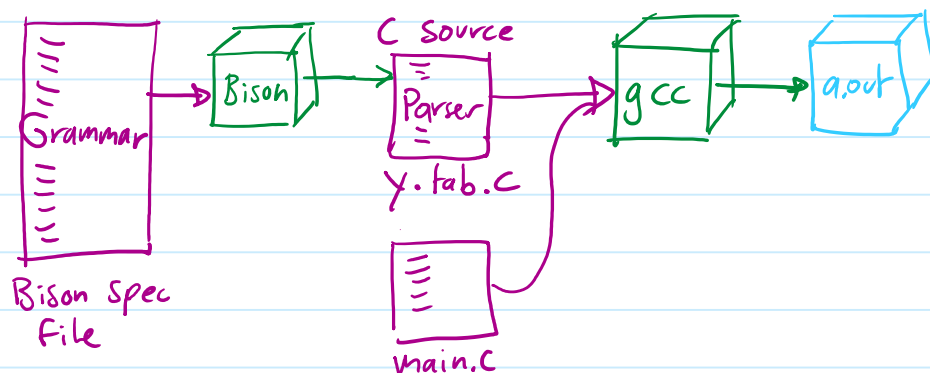
- HISTORY

'70 S. Johnson YACC B, BCPL, C

'80 R. Corbett Bison (GNU)

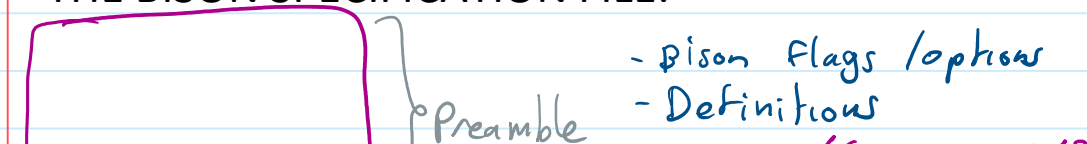
- BISON

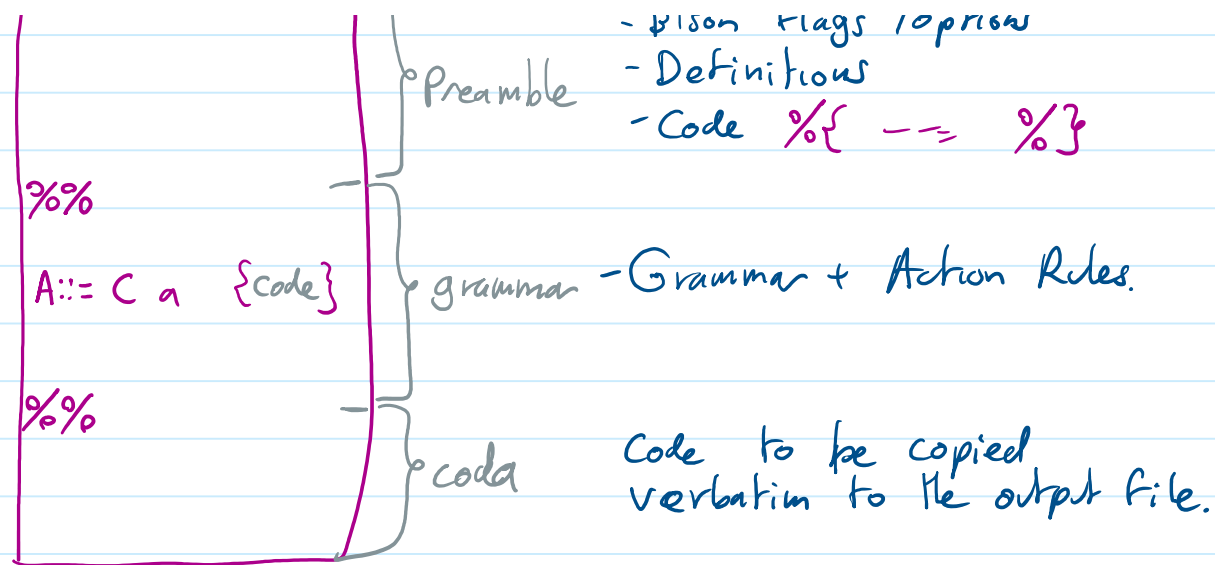
Bison is a "parser generator"



NOTE: Bison expects a function called **yylex()** to be available to read "terminal symbols" from the input.
lexemes.

- THE BISON SPECIFICATION FILE:





• THE CODE GENERATED BY BISON:

Functions:-

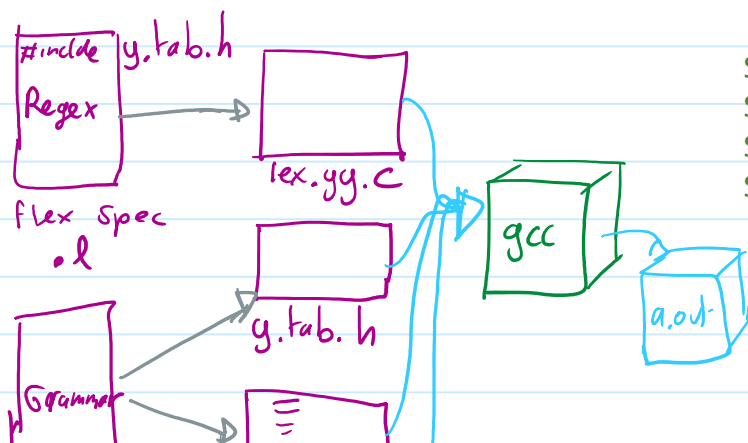
`yyparse()` - implements the shift-reduce parser

- calls `yylex()` to get terminal symbols
- returns 0 if parsing is successful, 1 otherwise

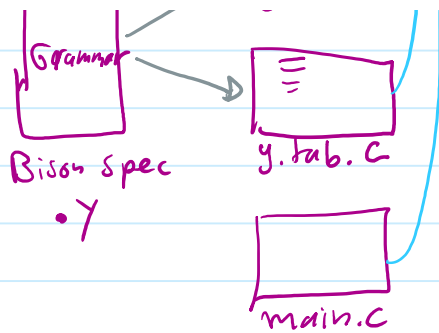
`yyerror(char *)` - called by `yyparse()` when encountering an error.

~~Also~~ you need to provide `yylex()` and `yyerror()`.

• THE BISON + FLEX WORKFLOW:



```
$ bison -d <.y file> → y.tab.h
$ flex <.l file> → lex.yy.c
$ bison <.y file> → y.tab.h
$ g++ -lfl *.c *.cpp
```

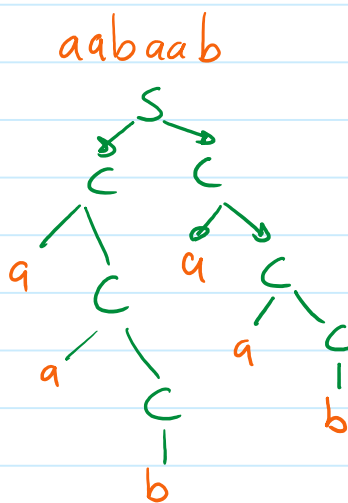


• BISON AND CONFLICTS:

- Bison will issue **warnings** when conflicts are detected in the grammar
- How Bison handles conflicts?
 - Reduce-Reduce.-
resolved in favour of first rule in file.
 - Shift-Reduce.-
resolved in favour of shift

• BISON DEMO

<https://www.gnu.org/software/bison/manual/>



Example #2 : flex + bison.

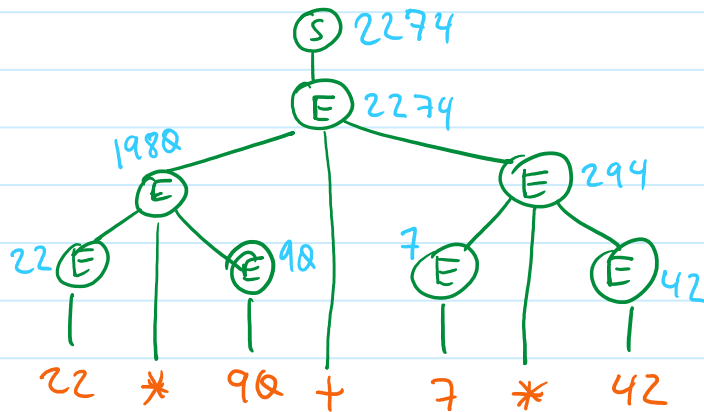
$S \rightarrow \epsilon \mid E$

flex

$S \rightarrow \epsilon \mid E$

$E \rightarrow \text{int} \mid E + E \mid E * E$

22 * 90 + 7 * 42



flex
recognize terminals
int + *
yy lex()

bison
- parse the grammar
- Evaluate expression.
yy parse()



—●— EOF