9 Bison

Monday, March 2, 2026 3:06 PM

- SHIFT-REDUCE PARSER:

Bison :- an implementation

input:- grammar specification

output:- source code for the corresponding shift-reduce parser.

Bison is designed to work with flex

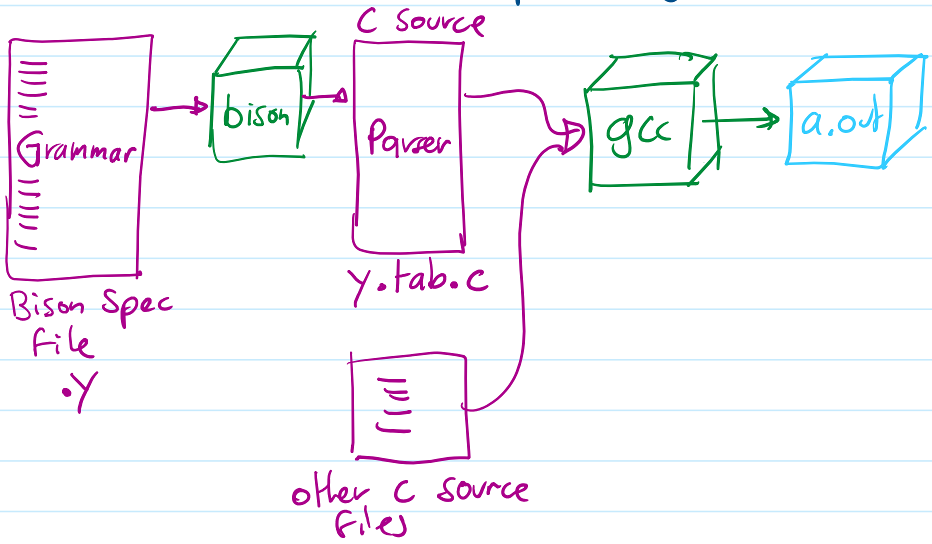
- HISTORY:

`70 S. Johnson YACC at ATT B, BCPL, C

`80 R. Corbett GNU Bison.

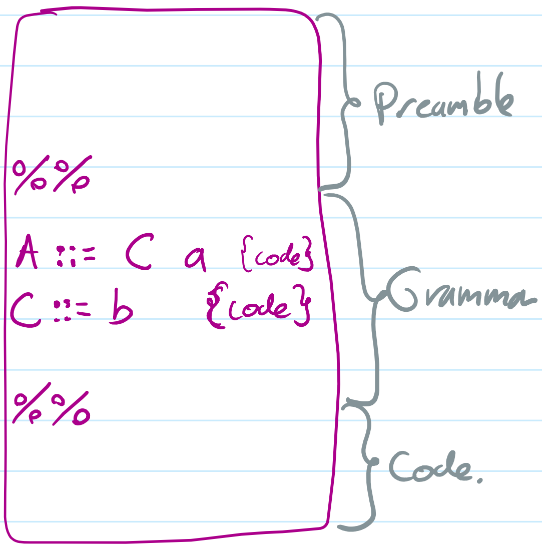
- BISON:

Bison is a "parser generator"



- Bison speds a function called yylex() to be available to read terminal symbols from the input.

- THE BISON SPECIFICATION FILE:



Bison options & directives
Code %{ ≡ ≡ %}

Grammar Rules
+ Action Code.

Code to be copied verbatim
to the output file.

- THE CODE GENERATED BY BISON:

Functions

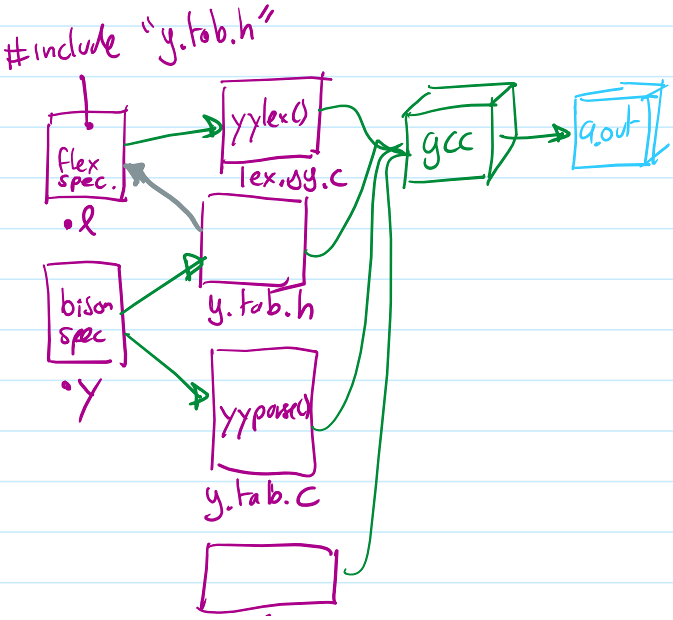
yyparse() - implements the shift-reduce parser
- calls yylex() to read terminal symbols
- returns 0 if parsing is succesful
1 otherwise.

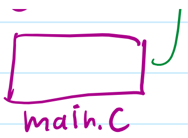
yyerror(char *) - called by yyparse() if an error is encountered.

🚩 you need to provide yylex() and yyerror()

- BISON + FLEX Workflow:

```
$ bison -d <.y file>
$ flex <.l file>
$ bison <.y file>
$ g++ -lfl *.c *.cpp
```





- BISON AND CONFLICTS:
  - Bison will issue warnings when conflicts are detected in the grammar.

  How Bison "resolves" conflicts:

  - Reduce-Reduce Conflict:
    Bison resolves using the first rule in file.
  - Shift - Reduce Conflict:
    Bison resolves in favor of shift

- LINK:

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

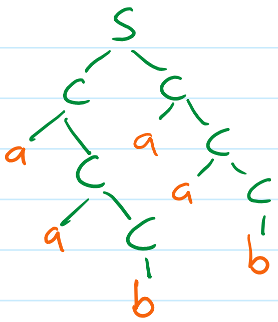
- BISON DEMO:

  Example #1

  $S \to CC$
  $C \to aC$
  $C \to b$

  two Terminals = {a, b}
  Non-terminals {NT_S, NT_C}

  adbaab



Example #2

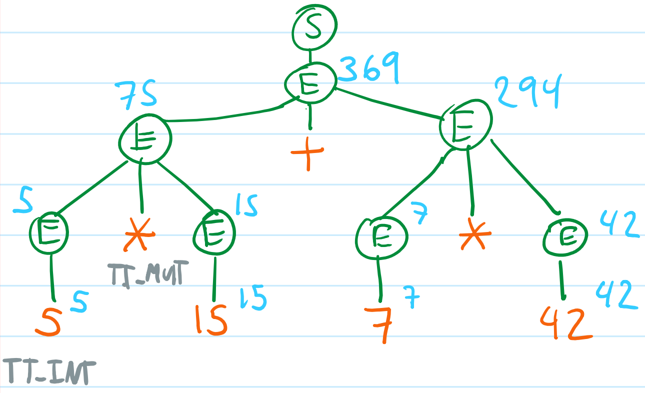
flex + bison.

$S \to E \mid \lambda$
$E \to E + E \mid E * E \mid \underline{int}$

5*15 + 7 * 42

flex: .l
recognize terminals RegEx.
int + *
TT_INT TT_PLUS TT_MULT

5 * 15 + 7 * 42



recognize terminals RegEx.

int + *

TT_INT TT_PLUS TT_MULT

bison: .y

- parse the grammar.

$1 $2 $3

- Compute "values" of intermediate nodes.