Compiler Design 11:47 AM (1 hour ago)

to bcc: me

contd... from the below email:

(1) Read Sec. 4.9 on the Textbook. It gives a very good introduction to bison. Note the following:
- The section initially has yylex() of its own, and then uses the yylex of flex (Sec.4.9.3).
- For PLUS, MULT tokens, directly the '<char>' is used in all places. We use explicit token names. Both are fine.

(2) Augment the ambiguous expression grammar (2-a to 2-d) to contain DIV, MINUS also. One point to note is, there is the unary-minus (arithmetic negation), which returns the same token (due to the same symbol: "-") as regular MINUS.
(2.a) First do, with production 2-g as
expr : MINUS expr {$$ = -$2;}
Note the number of conflicts, and the result.
(2.b) Give a different precedence to MINUS of 2-g ONLY using the %prec, as given in Fig. 4.59 of Textbook.

(3) Replace the 2-a to 2-d productions to the non-ambiguous grammar being discussed in the class. Don't have any %left, %right and see
- the # of conflicts, and
- the result.

(4) Augment (3) to have DIV, MINUS and unary-minus.

Once you do 1-9 of previous email and 2-4 above, you can move to the C-grammar. Doing so, you can focus on C language, and not be troubled bv bison.