

Homework # 4

due 2008/2/19

1 Predictive Parsing

Compute FIRST and FOLLOW sets for each nonterminal for the following grammar and then compute PREDICT sets for each production. What conflicts do you get? What is/are the cause(s)? How can the conflicts be fixed by changing the grammar?

```

expr : LET decls IN exprs END
      | ID
      | '(' expr ')'
      | expr '+' expr;
decls : decl more_decls;
more_decls :
           | decls ;
decl : ID ':' TYPE opt_init ';' ;
opt_init :
           | ASSIGN expr ;
exprs : expr more_exprs ;
more_exprs :
            | ';' exprs

```

NB: This grammar will not work for PA3.

2 LR Parsing

Run bison on this grammar. How many states do you get? Are there any conflicts? Describe any conflicts and how they could be fixed.

3 Using Parse Tables

Use the “Dangling ELSE Parse Tables” attached to this assignment to parse the following program:

```
A = B * X + 1;
```

Your answer should be similar to the form shown in Figure 2.22 on page 83 (Figure 2.29 on page 91 in the new textbook) except that this figure combines a shift or goto with a following reduce (confusing). You may combine a reduce with the following goto (standard).

In the grammar, we used precedence declarations to resolve some shift-reduce conflicts. Suppose we hadn't used the precedence declarations. Then the rules labeled **IGNORED** would be back in the table. All the shift-reduce conflicts would be resolved in favor of the shift rules. Supposing this case, how would your answer to the first part of this question differ? Please give the part of the trace that differs.

What is the problem with this result?