lex负责词法解析，而yacc负责语法解析，其实说白了就是lex负责根据指定的正则表达式，将输入的字符串匹配成一个一个的token，同时允许用户将当前匹配到的字符串进行处理，并且允许返回一个标识当前token的标识码。而yacc则负责进行语法解析，将一个个的token最终形成一个完整的语法。

lex与yacc之间的通信就是靠%union里定义的联合体来完成。其实%union里定义的联合体最终会被生成一个叫yylval的全局变量，这个全局变量可以在lex和yacc之间传递变量。

%type<xxx> 这里的xxx必须是%union里定义的一个成员变量，那么%type<xxx> 里定义的类型就会被存储这个成员变量里

%token<xxx> 与%type类似，只是它所代表的是token而已

$$代表当前的type所计算的最终结果，而$1代表type组成式中的第一个变量的值

为使一元’－’的优先级与’\*’相同，我们使用了子句

％prec’\*’

它说明它所在的语法规则中最右边的运算符或终结符的优先级与％Prec后面的符号的优先级相同，注意％Prec子句必须出现在某语法规则结尾处分号之前，％prec子句并不改变’－’作为二元运算符时的优先级。