基于Bison的Pascal语言语法及语义分析器

设计要点说明：

对于整数类型Integer，为简单，采用32位整数处理。

支持设置label并使用goto语句。

支持设置整形和实数型常量，并支持引用之前的常量作为新的常量。

支持类型重命名，并支持嵌套使用类型重命名。

支持指针类型，但指针必须是基类型的指针

支持任意维任意类型（除记录类型）的数组，访问数组时可以使用含有变量的表达式作为数组下标。

支持记录类型，及记录类型中嵌套数组和记录类型。

支持设置任意定义过类型及整形、实型、字符型、布尔型的变量。

常量、变量、过程与函数间不允许重名，但可以与类型名相同。

支持定义过程及函数，支持嵌套定义，支持递归过程/函数，支持传入任意数量的值参，函数返回值可以为基类型。支持函数作为表达式的一部分。

对于每个Block，记录Label，Const，Type，Array，Record，Var，Procedure/Function表项的入口。

支持赋值语句，将同类型的表达式的结果赋给变量。

支持条件语句，支持if-then，if-else-then，case语句，且case语句中支持else。

支持循环语句，for-to，for-downto，while语句及repeat语句。

表达式可以包含四则运算，取模运算，位运算，变量，函数，及布尔类型的逻辑运算。

为实现额外功能，对于类Pcode解释器增加额外功能：

对于现有的四则运算，当l域为0时进行整数运算，当l域为1时进行浮点数运算

对于STO及LOD操作，当offset为-1时，从栈顶取一个值作为offset（为支持数组动态下标）。

对STO及LOD操作进行扩展，使其支持浮点数的读取和存入：STOD，LODD。

支持对于指针类型变量的存入及读取操作：LODP，LODPD，STOP，STOP，STOFP，STOFPD。

增加额外的OPR操作，支持与，或，非，异或操作，当l域为0时为整数位运算，当l域为1时进行布尔类型逻辑运算。

存在的问题及未来的改进目标：

函数不支持传入形式参数。

不支持字符串变量及常量。

不支持文件操作。

编译选项：

-d，生成flex需要的符号表定义头文件。

-i，词法分析时不区分大小写。

--yylineno，生成行号信息

语法EBNF范式：

Pascal EBNF Definition（<https://www.fit.vutbr.cz/study/courses/APR/public/ebnf.html>）

文件列表：

code.h 用于类Pcode解释器的指令结构定义。

interpret.c 类Pcode解释器。

pascal.h 定义语法分析器的成分属性类型。

pascal.l 词法分析器源文件。

pasca.y 语法及语义分析器源文件。

Makefile 编译脚本。

pl0\_data/ 样例pl0程序。

pascal\_data/ 样例pascal程序。

输入及输出：

编译成功后，运行”./compiler filename”，其中filename为待编译pascal源文件。

在终端输出“start pl0”之后正常输入输出。

语法成分分析在“syntax.out”文件中。