甘棠约束条件编码器

使用方法

Linux 二进制已经编译好了，直接 ./encoder 即可。

自动读取当前目录 ./data 下的三个 csv 文件作为程序输入，如果需要修改输入数据，就改那三个 csv 文件（一定要UTF-8编码的，因为文件里有中文，Office Excel 转成的 csv 是 GBK 编码的，需要额外处理一下）。

```
约束条件可以使用任意数理逻辑表达式，包括 =>、<=、()、!、~、&、|、<=>，并且可以任意嵌套
运算符默认优先级 (~|!) > (&) > (|) > (=>) = (<=) > (<=>)
```

程序的标准输出就是 CNF 格式的，并且包含合理的注释信息，直接重定向就可以：

```
./encoder > result.cnf
```

如何编译

比较懒没有写 makefile，.h 和 .cpp 文件之间互相依赖太多了，基本改了一个就要全部重新编译，所以直接用脚本，每次重新全部编译。

```
./build.sh
```

没有使用系统调用，在 Windows 上编译也可以直接编译通过，也是同样的编译命令。

正确性

在给定的示例编码文件中，最终编码文件 result.cnf 在 appmc 和 projMC 的计算中，都为 36288，和给定的目标完全一致。

projMC 的程序好像是有数组越界的 BUG，需要稍微把 p cnf 中的变量数量改的比实际变量的范围和数量要大一点才不会报错，appmc 没有问题。

```
> approxmc --epsilon 0.05 result.cnf
c Total time (this thread) : 0.67
c [appmc+arjun] Total time: 0.67
c [appmc] Number of solutions is: 4536*2**3
s SATISFIABLE
s mc 36288

> ./binary/projMC_linux -fpv=name_nbvar.var result.cnf
c Projected Model Counter Information
c Number of recursive calls: 16
c Number of decomposable AND nodes: 17
c Number of UNSAT subproblems: 3
c Number of positive hits: 2
c
c Final time: 0.015772
c
s 36288
```

编码思路

对于基本可选特征值和特征族的编码，可以用传统方式直接编码，最麻烦的一个部分是约束条件的形式是一个没有任何特殊性质的数理逻辑表达式（&、|、~、=>、<=>、()、!），并且可以任意组合，因此需要构造一个通用的语法分析器。

语法分析

使用 LR(1)/LALR(1) 文法自动机对每个约束语句进行语法分析，按照数理逻辑的符号优先级定义规约顺序，并且规则设定为左结合优先，使二义性文法符合 LR(1) 文法，利用文法构造器，去生成语法分析代码框架。

语法制导

按照自底向上的语法制导方法，每次依次合并两个变量或者取反，用新的变量等价代替它，然后将新的变量压入栈中，这样会有状态数过多的问题。

实际上连续的析取和合取子句，只需要一个等价的变量去代替就可以了，因此构建一个新的类代表若干个子句的析取或合取式子，重载该类的所有操作，遇到必须合并的时候再整体等价为一个变量操作。