实验报告

## 产生式系统实验报告

**班级：10011703 姓名：李昕 学号：2017302629**

**一 实验题目**

制作小型动物识别系统进行正向推理

**二 实验目的**

1 熟悉和掌握产生式系统的构成和运行机制；

2 掌握基于规则推理的基本方法和技术，掌握正确的正向推理和逆向推理方法；

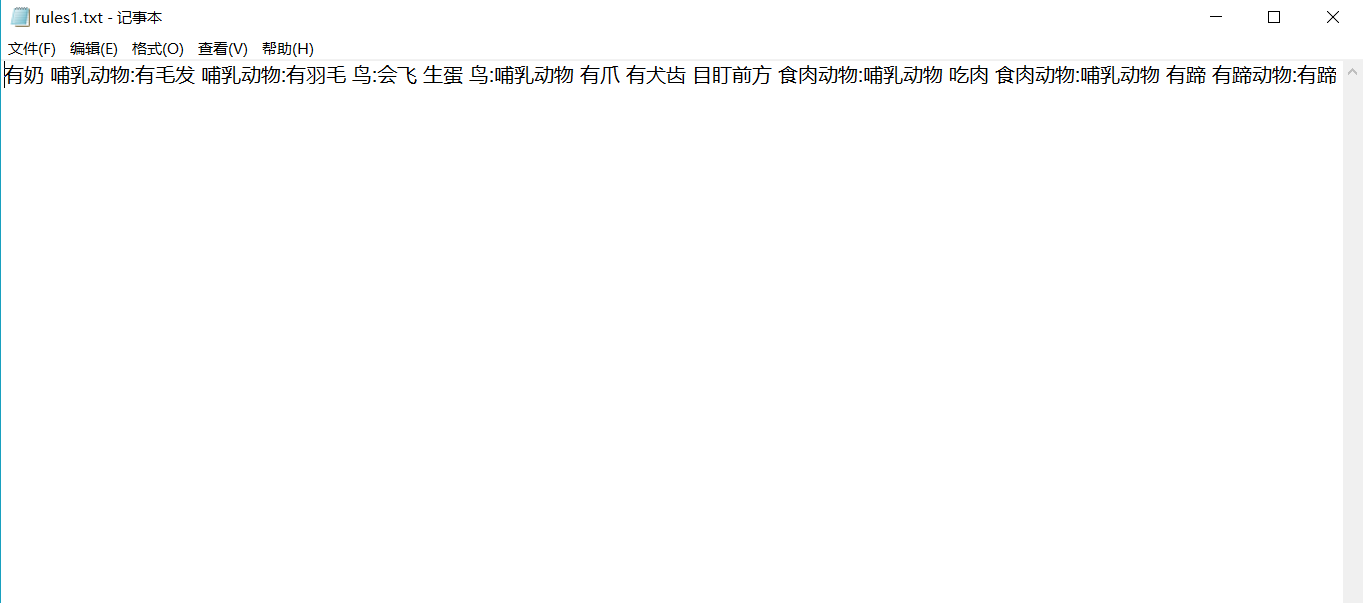
3 熟悉在具体问题中如何实现正向推理和逆向推理的求解流程。

**三 实验要求**

以动物识别系统为例，用选定的编程语言建造规则库和综合数据库，并能对它们进行增加、删除和修改操作，开发能进行正确的正向推理或反向推理的推理机。

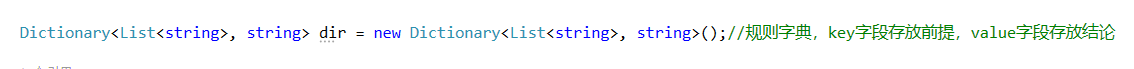
**四 数据结构**

1)规则采用文件记录



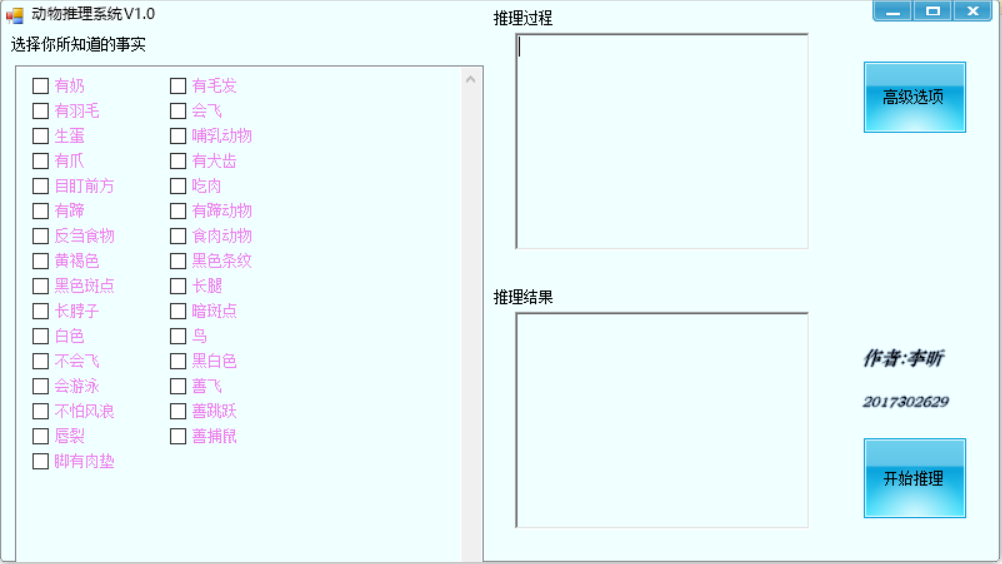
每一条规则由“：”分割，在每一条规则中，每一个事实用空格分开，最后一个为结论，当需要加入规则时可以在文件中直接加入也可以在窗体中加入。

2)在程序中所用到的数据结构是C#提供的dictionary字典，key字段我存放的是规则的前件，value字段我存放的是规则的结论

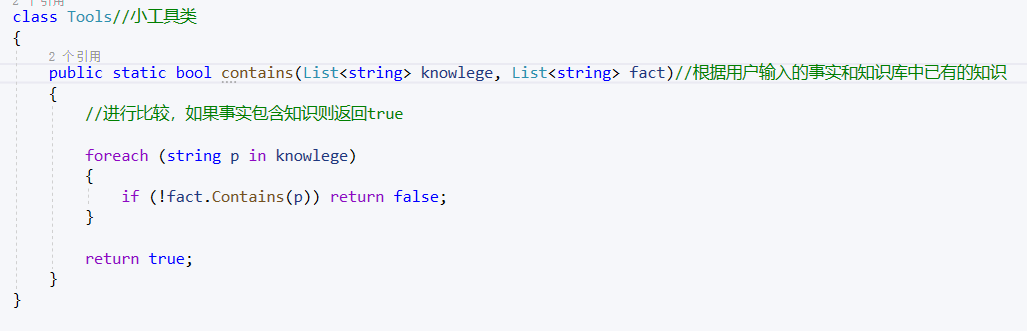


**五 实验算法**

**主界面如下：(界面的实现是通过网上下载的界面美化类，继承这个界面类得到的，压缩包是CSkin 16.1.14.3，老师如果想验证可以查看一下这个压缩包中的使用教程配置一下VS)**

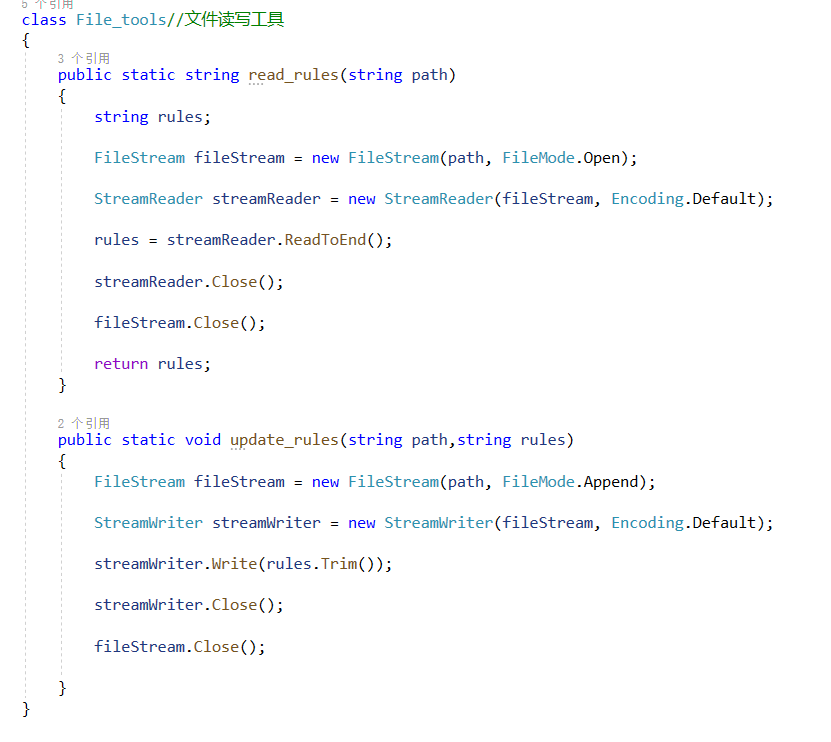


1. 我先设计了一个工具类Tools，用来支持规则匹配算法



如果用户输入的事实中包含了一条知识的所有前件则返回一个true否则返回false

2.接着我设计了一个文件读写工具类File\_tools用来简化文件读写操作



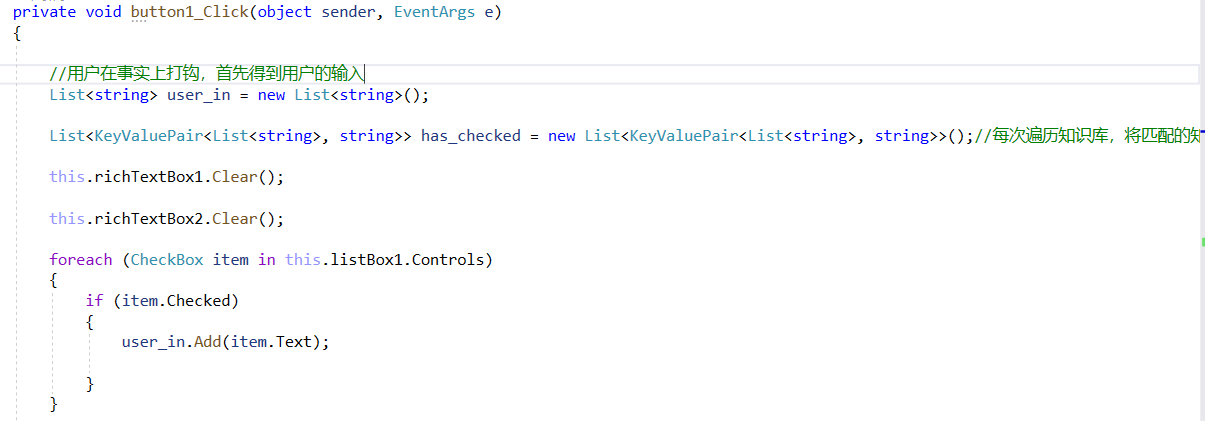
第一个方法是read\_rules功能是根据输入的路径读到文件中的所有文本

第二个方法是update\_rules功能是用来添加指定的规则到文件中

3.当主界面加载的时候需要初始化字典数据结构，从文件中读到规则然后依次将规则放入到字典中：



4.当用户勾选完了所需要的知识后，按下开始推理按钮，按钮的触发事件对应的函数的功能是，首先得到用户所有的输入事实构造一个List:



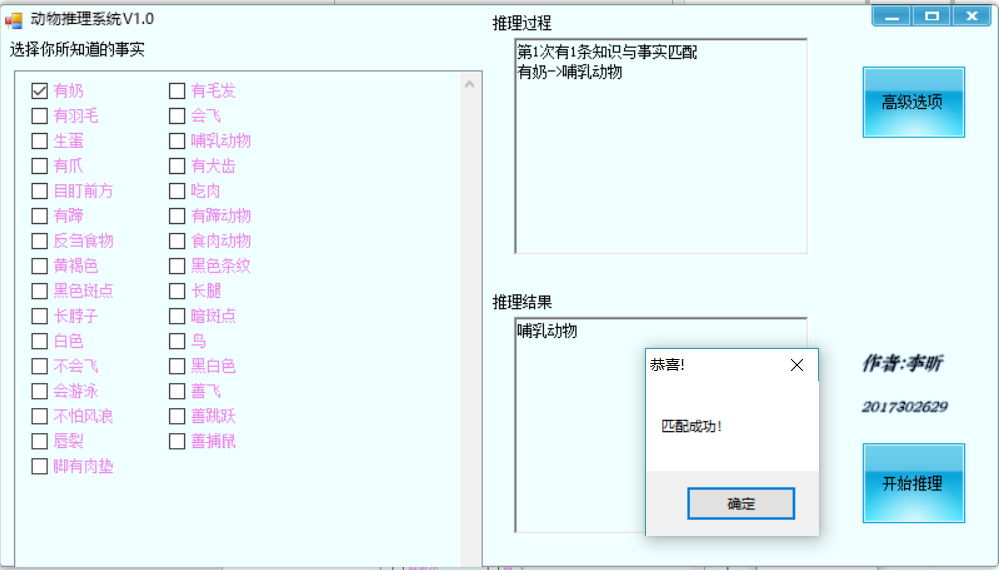
接下来便进入一个循环，每次将已有的事实和规则集中的知识前件进行匹配，即每次遍历知识库，把发生匹配的知识放到一个List中，如果说这个List不为空，说明本次有相应的知识发生了匹配，然后要做的工作是把已知事实中包含的匹配的知识的前件删除，然后把匹配的知识的结论加入到已知事实的末尾：

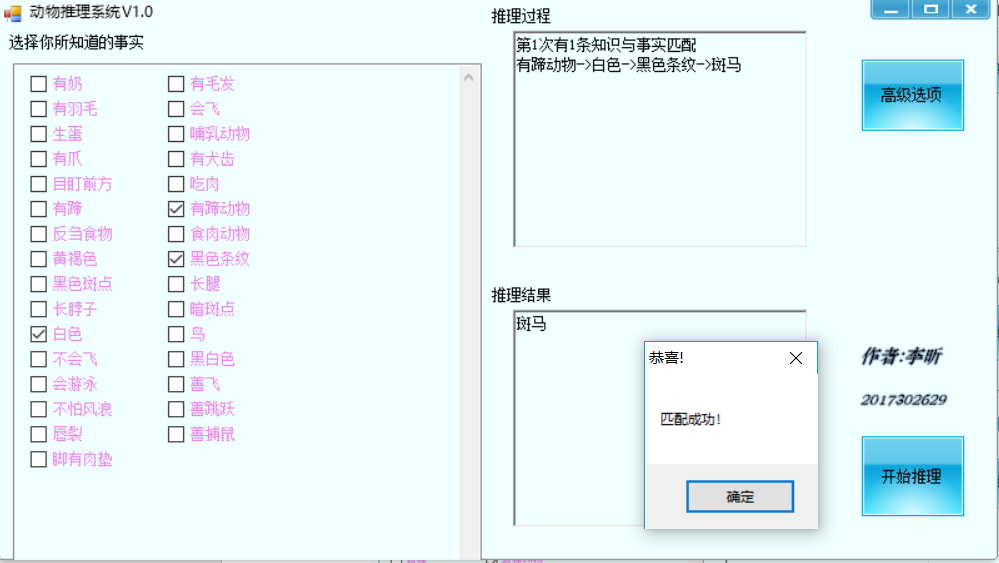


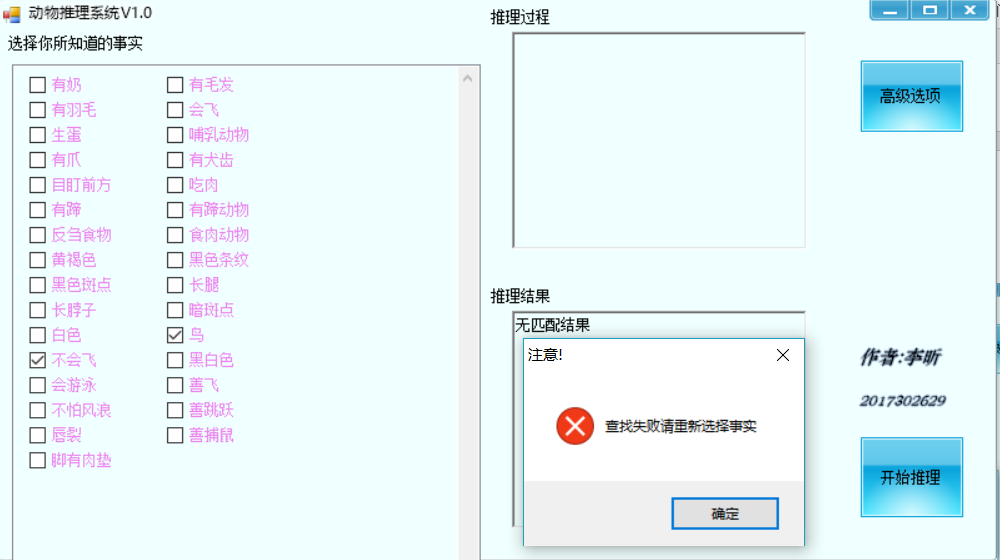
如果遍历一遍知识库，发现没有知识与之匹配了，即List为空时，并且这时是第一次推理，则推理失败，没有匹配项，而如果这是不是第一次推理，则可以输出事实集中的最后一项作为结果：



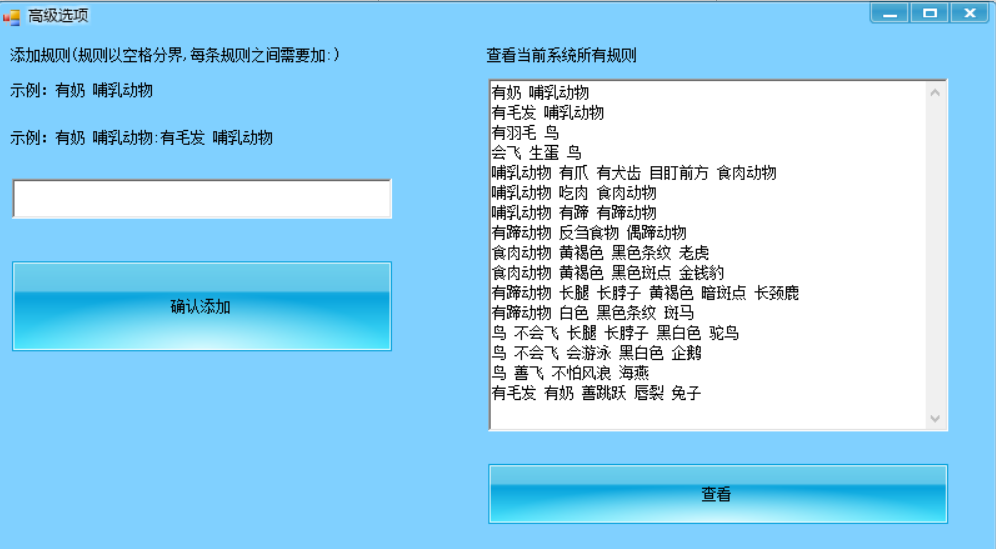
**六 实验结果**

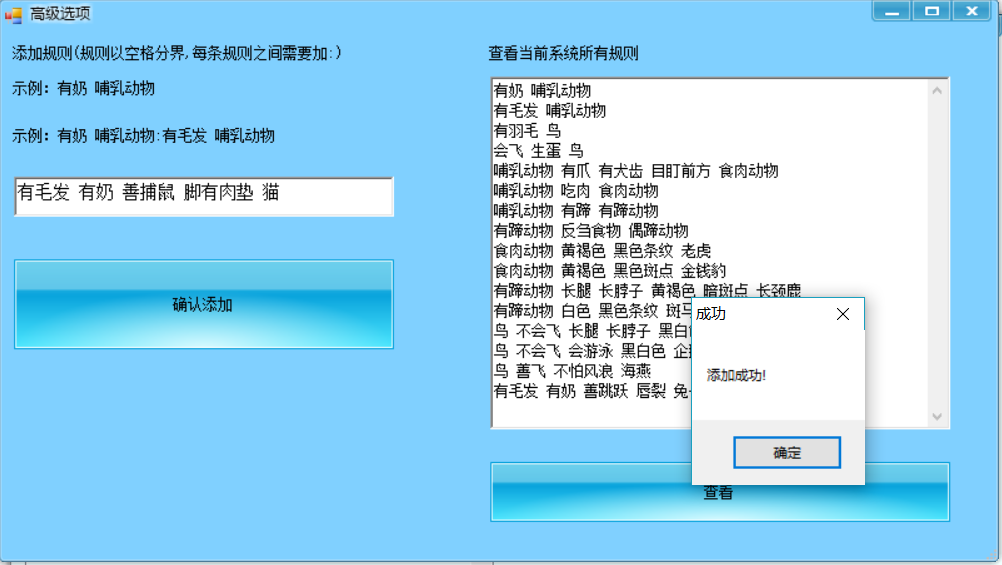
运行截图：  


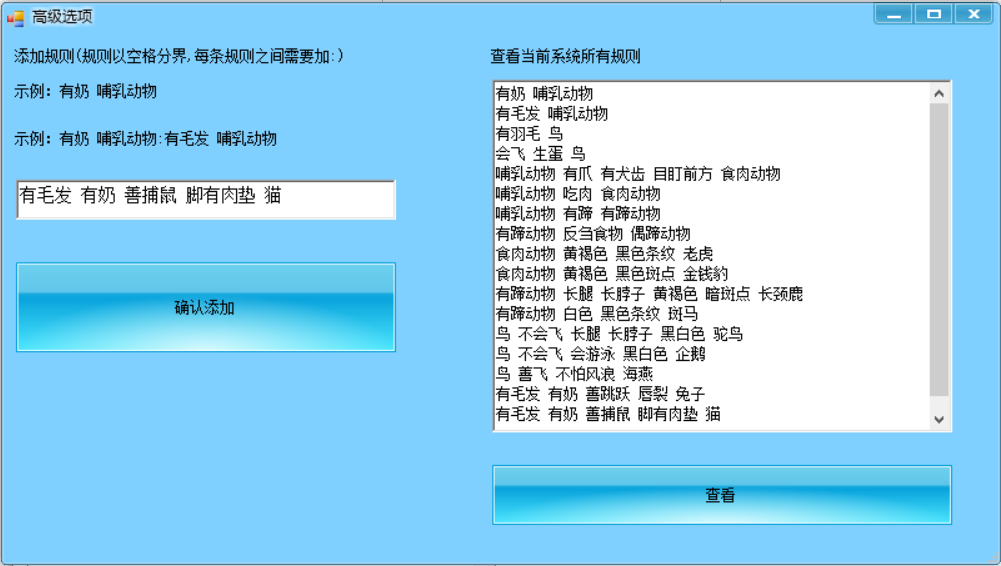


匹配失败的例子：  


点击高级选项：  


查看系统中所有的规则：  


添加一条规则：  


再次查看所有规则：  


**七 实验总结及体会**

这次的实验自己动手实践了一个小型的动物推理系统，期间遇到了不少问题，首先我存储规则时，本来规则之间是通过空行来分割的，但是我在实际代码中分割的时候出现了问题，无法得到目标的规则，一直没有找到原因，于是我换了一种方式，采用“:”来分割，这次就没有问题了可以分割出规则。在实现添加规则功能模块时，由于不熟悉C#提供的文件读写的API，一开始写文件的时候写到了文件的开始覆盖了之前的文件，后来通过查找资料，可以设置FileMode为Append即可追加到文件的末尾。