**实验二 推理技术**

**一、 实验目的**

熟悉和掌握产生式系统的运行机制，掌握基于规则推理的基本方法，利用规则演绎解决规划问题。

**二、实验内容**

1. 对已有的产生式系统(默认的例子)进行演示，同时可以更改其规则库或（和）事实库，进行正反向推理，了解其推理过程和机制。自己建造产生式系统（包括规则库和事实库），然后进行推理，即可以自己输入任何的规则和事实，并基于这种规则和事实进行推理。

2. 观察并验证简单的推理过程。对照实验过程，自己写一个简单的规则推理,再用实验环境创建一个相应的模型,并在实验中验证或修改它。

**三、 实验记录**

**产生式系统实验报告表二**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 姓名 |  | | | 年级  班级 |  | 指导老师 |  | 日期 |  | | | | |
| 实验目的 | | 熟悉和掌握产生式系统的运行机制，掌握基于规则推理的基本方法，利用规则演绎解决规划问题。 | | | | | | 推理 方法 | | □正向推理  □反向推理 | | | |
| 建立规则库 | | | | | | 建立事实库 | | | | | | | |
| 该动物是哺乳动物 <- 该动物有毛发.  该动物是哺乳动物 <- 该动物有奶.  该动物是鸟 <- 该动物有羽毛.  该动物是鸟 <- 该动物会飞&会下蛋.  该动物是食肉动物 <- 该动物吃肉.  该动物是食肉动物 <- 该动物有犬齿&有爪&眼盯前方.  该动物是有蹄类动物 <- 该动物是哺乳动物&有蹄.  该动物是有蹄类动物 <- 该动物是哺乳动物& 是嚼反刍动物.  该动物是金钱豹 <- 该动物是哺乳动物&是食肉动物&是黄褐色&身上有暗斑点.  该动物是虎 <- 该动物是哺乳动物&是食肉动物&是黄褐色&身上有黑色条纹.  该动物是长颈鹿 <- 该动物是有蹄类动物&有长脖子&有长腿&身上有暗斑点.  该动物是斑马 <- 该动物是有蹄类动物&身上有黑色条纹.  该动物是鸵鸟 <- 该动物是鸟&有长脖子&有长腿&不会飞&有黑白二色.  该动物是企鹅 <- 该动物是鸟&会游泳&不会飞&有黑白二色.  该动物是信天翁 <- 该动物是鸟&善飞. | | | | | | %------动物识别系统事实集：  %会游泳. %--该动物是企鹅  %不会飞.  %有黑白二色.  %该动物是鸟.  %-------- %--该动物是鸟  %该动物会飞.  %会下蛋.  %----该动物是金钱豹 <- 该动物是哺乳动物&是食肉动物&是黄褐色&身上有暗斑点.  %该动物有毛发.  %是食肉动物.  %是黄褐色.  %身上有暗斑点.  %----该动物是虎 <- 该动物是哺乳动物&是食肉动物&是黄褐色&身上有黑色条纹.  该动物是哺乳动物.  是食肉动物.  是黄褐色.  身上有黑色条纹.  %----该动物是长颈鹿 <- 该动物是有蹄类动物&有长脖子&有长腿&身上有暗斑点.  %该动物是有蹄类动物.  %有长脖子.  %有长腿.  %身上有暗斑点. | | | | | | | |
| 预测结果 | | | 该动物是虎 | | | | | | | | | | |
| 实验过程及结果(注意观测规则的匹配过程和方法) | | | 创建一个该动物是虎的query，然后系统根据事实（该动物是哺乳动物，是食肉动物，是黄褐色，身上有黑色条纹）匹配到规则进行推理。 | | | | | | | | 备注(原因等) | | |
|  | | |
| 学生结论 | | | 通过设定的事实和规则，可以判断一个命题是否为真。推理系统形成一棵以询问为根的有向树，每个节点的分支是或的关系，节点内部的所有条件是与的关系，因此搜索树是一个与或图。如果有一个叶子节点的条件可以满足，那么就说明询问为真，否则询问为假。 | | | | | | | | | | |
| 指导老师意见 | | |  | | | | | | | | | | |
|  |  |  |  |  |  |  |  |  | | |  |  |  |

**一个简单的产生式系统：**

规则库：

是面向对象 <- 是有接口.

是面向对象 <- 是有类.

是编译语言 <- 是需要编译运行.

是脚本语言 <- 是屎.

是脚本语言 <- 是效率较低.

是脚本语言 <- 是不需要编译运行&是逐语句运行.

是高级语言 <- 是面向对象.

是高级语言 <- 是简单&是效率较低.

是效率较高 <- 是编译语言.

是底层语言 <- 是繁琐&编译语言&直接操作内存.

是屎 <- 是代码很难看.

是golang <- 是高级语言&是屎.

是python <- 是高级语言&是脚本语言.

是java <- 是高级语言&是编译语言.

是asm <- 是底层语言.

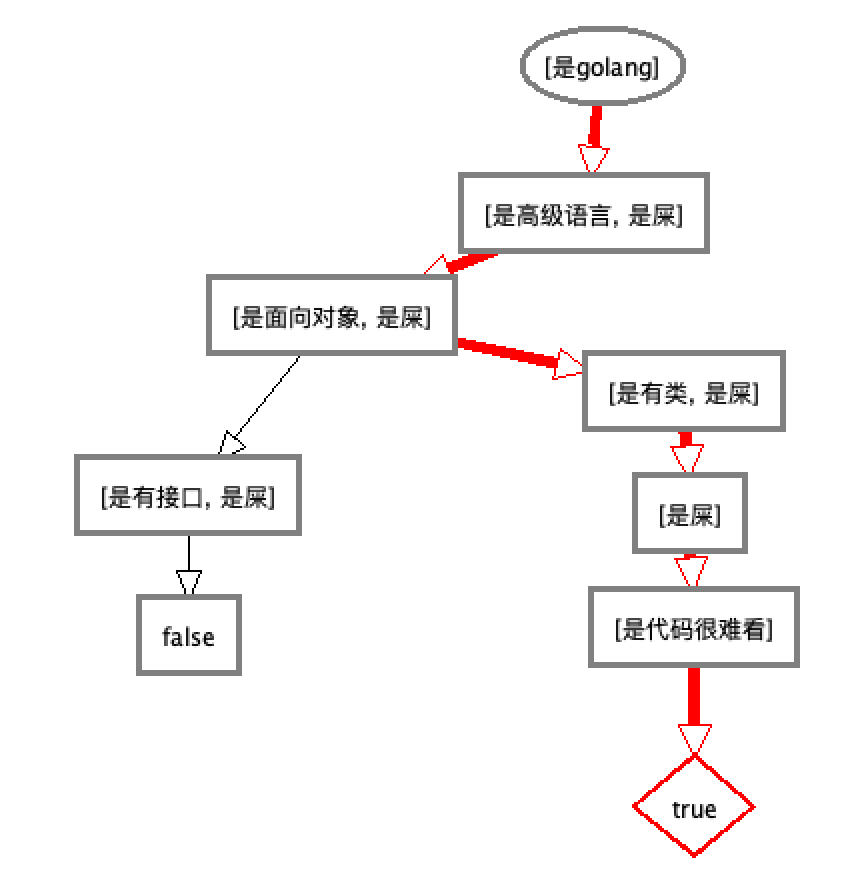
事实库:

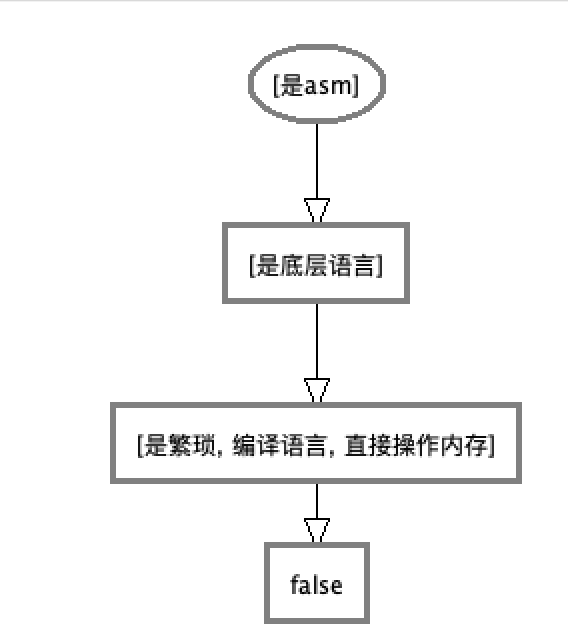
是代码很难看.

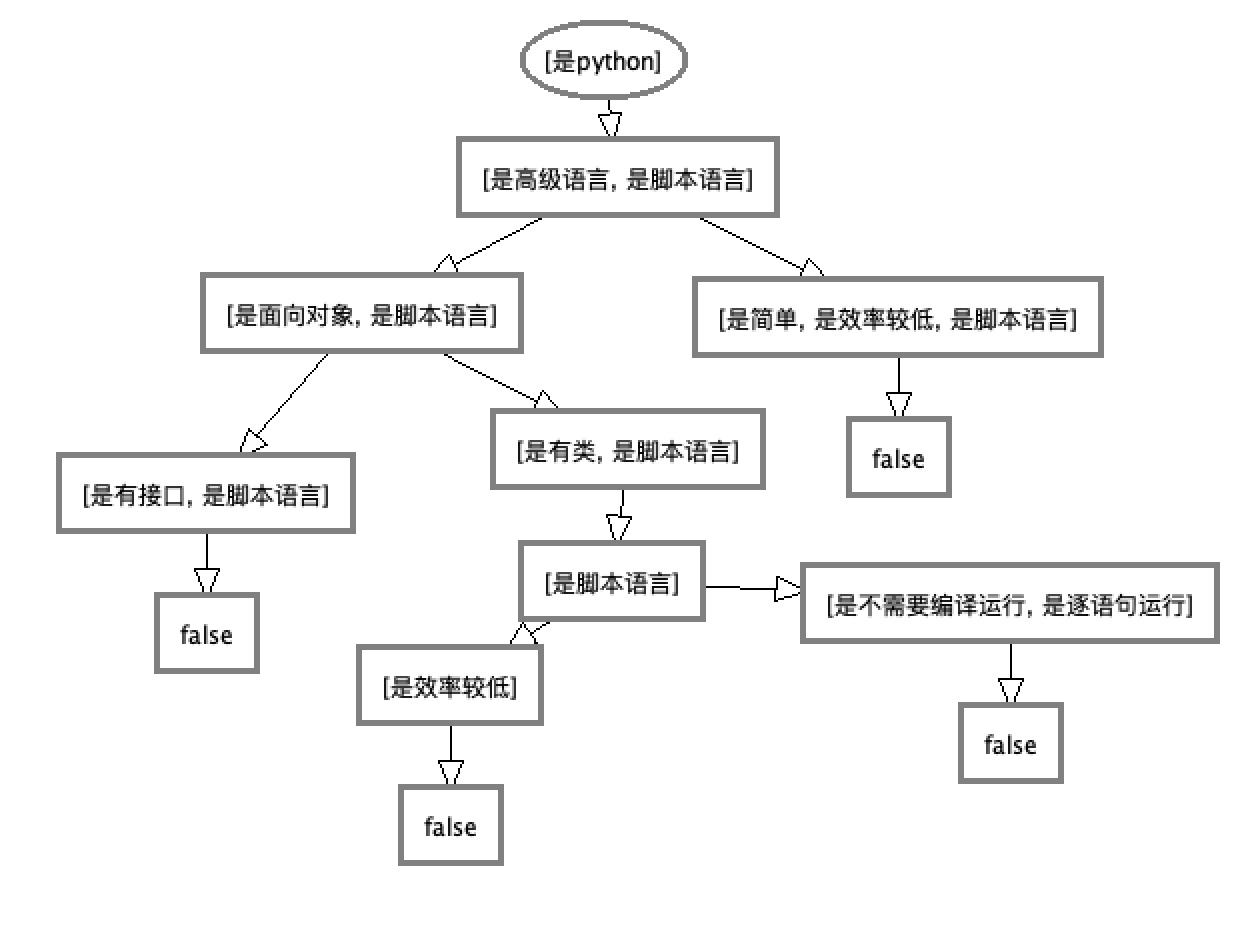
是有类.

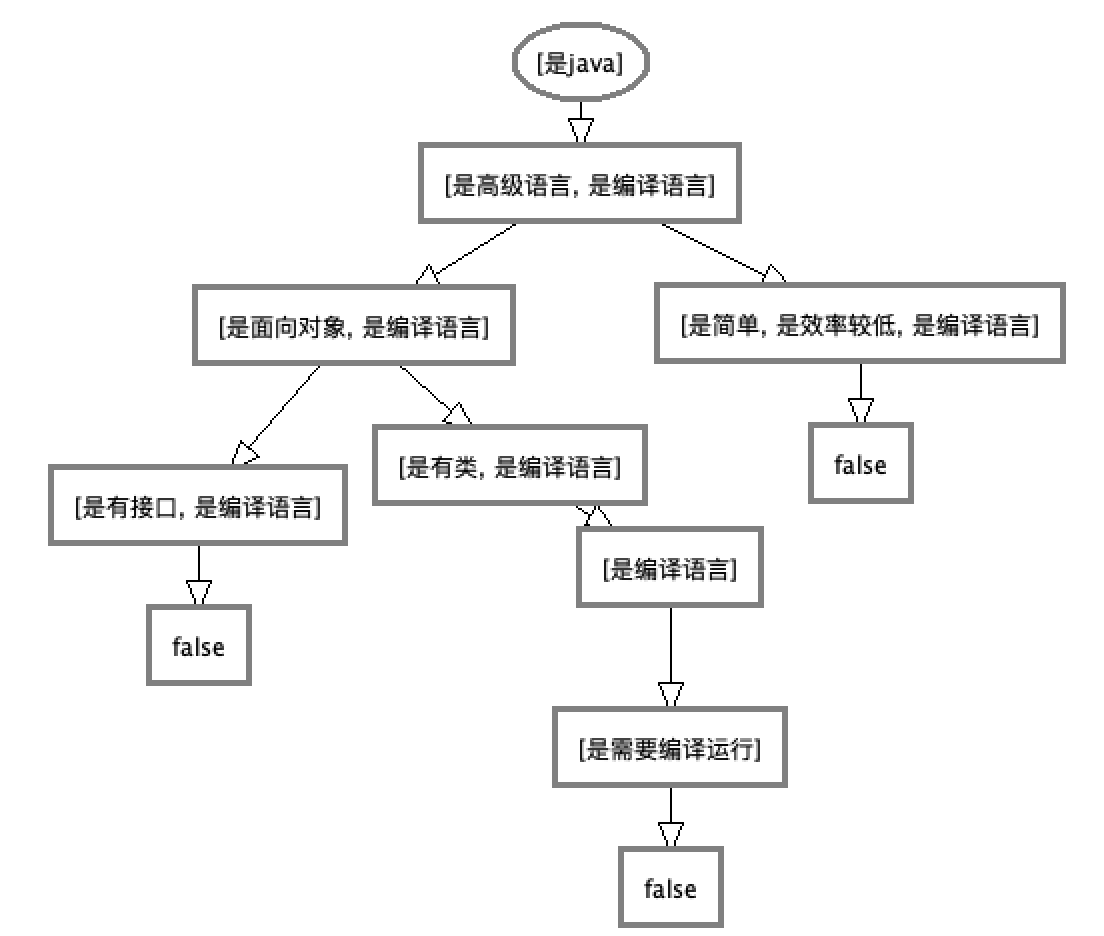
预测结果：是golang

实验验证过程如下：









反向推理过程中，假设一个询问，根据规则向上推（可能可以推出多个分支），遇到事实即“消去”，最终只有“是golang”的询问能够找到一条通路完全消除，即证明该询问为真，其他的则为假。