

产生式系统 实验报告

20110204 熊浩男

一、实验目的

熟悉产生式表示法，掌握产生式系统的运行机制，以及基于规则推理的基本方法

二、实验内容

设计并编程实现一个小型产生式系统（如分类、诊断等类型）。

三、实验要求

1. 具体应用领域自选，具体系统名称自定（建议不要用动物识别系统），如：<植物识别系统>
2. 用产生式规则作为知识表示，利用产生式系统实验程序，建立知识库，分别运行正、反向推理

四、实验设计

1. 设置产生式系统，包括系统名称和系统谓词，给出谓词名及其含义
2. 编辑知识库，通过输入规则或修改规则等，建立规则库
3. 建立事实库（综合数据库），输入多条事实或结论
4. 运行推理，包括正向推理和反向推理，给出相应的推理过程、事实区和规则区

本实验设计了一个产生式系统，以植物识别系统为例。首先进行事实规则与结论的预先输入。在运行此系统时，选择事实与规则。系统会进行推理，从而得出结论。

产生式系统使用方法

本实验使用 Python 语言实现，使用方法如下。

1. 输入命令 `pip install pick` 安装依赖
2. 在 `db.txt` 中输入规则（规则在下方介绍）
3. 输入命令 `python identify_system.py` 运行程序
4. 使用 `上/下 方向键` 与 `空格键` 选择所有符合要求的事实与规则，至少选择一项
5. `回车` 查看结论

`db.txt` 格式

条件编号、事实规则 和 结论 应当以特定格式写在 `db.txt` 文件中，格式规则如下。

```
<Number>: IF <Condition1> [& <Condition2>...] THEN <Conclusion>
```

举例来说： `R12: IF 被子植物 & 蔷薇科 & 木本 & 可食用 & 结果实 -> 苹果树`

行首为 `#` 符号的行会被认为是注释，不会处理。

五、实验结果

普通使用

1. 若选择的条件不对应任何一条结论，则会提示 没有通过条件找到您的结论。

python3 identify_system.py

请选择全部条件。

(x) 有刺

(x) 吸引菜粉蝶

(x) 无根

(x) 蔷薇科

(x) 仙人掌科

(x) 种子有果皮

(x) 被子植物

* (x) 喜阳

() 无茎叶

() 黄色花

() 有托叶

() 种子无果皮

() 水生

() 可食用

() 叶片针状

() 十字花科

() 木本

() 藻类植物

() 十字形花冠

() 缺水环境

() 结果实

() 裸子植物

() 药用

() 有白色粉末

littlebear@Littlebear-MacBook-Air:~/OneDrive/大学/大三上/人工智能导论/Production system

→ littlebear Production system (master) x python identify_system.py

没有通过条件找到您的结论

→ littlebear Production system (master) x

2. 若有且只有一条结论满足所有条件，则会显示结论。

```
python3 identify_system.py

请选择全部条件。

( ) 无根
(x) 仙人掌科
( ) 喜阳
( ) 木本
( ) 十字花科
( ) 叶片针状
( ) 被子植物
( ) 有托叶
* (x) 有刺
( ) 果实
( ) 黄色花
( ) 裸子植物
( ) 水生
( ) 药用
( ) 有白色粉末
( ) 无茎叶
( ) 藻类植物
( ) 十字形花冠
( ) 吸引菜粉蝶
( ) 蔷薇科
( ) 可食用
( ) 缺水环境
( ) 种子有果皮
( ) 种子无果皮
```

```
littlebear@Littlebear-MacBook-Air:~/OneDrive/大学/大三上/人工智能导论/Production system
→ littlebear Production system (master) x python identify_system.py
您输入的条件找到的结论是：仙人球！
→ littlebear Production system (master) x
```

3. 若有多项结论符合输入条件（即发生了冲突），则会提示 提供的条件对应数条结论。本系统采用了显示第一条结论的冲突消解策略。

```
python3 identify_system.py

请选择全部条件。

( ) 仙人掌科
( ) 种子无果皮
( ) 无茎叶
( ) 可食用
( ) 无根
( ) 结果实
( ) 黄色花
( ) 吸引菜粉蝶
( ) 裸子植物
( ) 被子植物
* (x) 水生
( ) 缺水环境
( ) 木本
( ) 十字花科
( ) 叶片针状
( ) 有托叶
( ) 十字形花冠
( ) 蔷薇科
( ) 种子有果皮
( ) 喜阳
( ) 有刺
( ) 有白色粉末
( ) 藻类植物
( ) 药用
```

```
littlebear@Littlebear-MacBook-Air:~/OneDrive/大学/大三上/人工智能导论/Production system
→ littlebear Production system (master) x python identify_system.py
您提供的条件对应数条结论，第一条是：荷花！
→ littlebear Production system (master) x
```

高级使用

在运行时添加 `--debug` 参数可以查看推理过程，事实区与结论区。

```
python3 identify_system.py --debug

请选择全部条件。

( ) 可食用
( ) 无茎叶
( ) 有白色粉末
( ) 种子有果皮
( ) 种子无果皮
( ) 药用
* (x) 十字花科
( ) 叶片针状
( ) 裸子植物
(x) 黄色花
( ) 十字形花冠
( ) 果实
( ) 水生
( ) 喜阳
( ) 蔷薇科
( ) 缺水环境
( ) 被子植物
( ) 无根
( ) 有托叶
( ) 木本
( ) 有刺
( ) 仙人掌科
( ) 藻类植物
( ) 吸引菜粉蝶
```

```
littlebear@Littlebear-MacBook-Air:~/OneDrive/大学/大三上/人工智能导论/Production system
→ littlebear Production system (master) x python identify_system.py --debug
选择的所有条件与规则: ['黄色花', '十字花科']
处理规则 '黄色花' 后的结论区: ['被子植物', '裸子植物', '藻类植物', '蔷薇科', '十字花科', '仙人掌科', '玫瑰', '荷花', '仙人球', '水棉', '苹果树', '油菜', '海带', '松树']
处理规则 '十字花科' 后的结论区: ['油菜']
您输入的条件找到的结论是: 油菜!
→ littlebear Production system (master) x █
```

六、结果分析与总结

本实验使用 Python 语言设计并实现了一个以植物识别系统为例的产生式系统。

在这个实验后，我对产生式系统以及基于规则的推理过程有了更深刻的认识。

本实验代码开源在：<https://github.com/littlebear0729/Production-system.git>