动物识别系统-设计文档

设计思路

P19 例2.7的动物识别规则库中，有如下知识：

R1： IF 动物有毛 THEN 动物是哺乳动物

R2： IF 动物有奶 THEN 动物是哺乳动物

R3： IF 动物有羽毛THEN 动物是鸟

R4： IF 动物会飞 AND 动物会下蛋 THEN 动物是鸟

R5： IF 动物吃肉 THEN 动物是食肉动物

R6： IF 动物有犬齿 AND 有爪 AND 眼盯前方THEN动物是食肉动物

R7： IF 动物是哺乳动物 AND有蹄 THEN 动物是有蹄类动物

R8： IF 动物是哺乳动物 AND该动物是反刍动物 THEN 动物是有蹄类动物

R9： IF 动物是哺乳动物 AND是食肉动物 AND黄褐色 AND 暗斑点 THEN 动物是豹

R10：IF 动物是哺乳动物 AND 食肉动物 AND黄褐色 AND 黑色条纹 THEN 动物是虎

R11：IF 动物是有蹄类动物 AND 有长脖子 AND 有长腿 AND 有暗斑点 THEN 动物是长颈鹿

R12：IF 动物是有蹄类动物 AND 有黑色条纹 THEN 该动物是斑马

R13：IF 动物是鸟 AND 不会飞 AND 有长脖子 AND 有长腿 AND 有黑白二色 THEN 动物是鸵鸟

R14：IF 动物是鸟 AND 不会飞 AND 会游泳 AND 有黑白二色 THEN 动物是企鹅

R15：IF 动物是鸟 AND 善飞 THEN 动物是信天翁

数据结构：

每个规则包含了一个或多个动物特征，以及一个由这些特征组合得出的结论。

于是我设计的规则库格式为：文本文档当作规则库，文档内的一行视为一条规则，动物特征和结论都简化后视为词条，结论永远放在规则最后面，词条之间用英文逗号分隔，例如：

会飞,会下蛋,鸟

经整理后，规则库文档如下：

文本

描述已自动生成

程序设计：

使用环境：python3.10 ，Windows10

专家系统要求除了进行识别以外，还能对规则库进行增、删、改操作，这些操作都很简单，在程序中读取规则文档时，使用readlines()函数读取全部规则，规则会以列表类型按行分隔，再使用for语句就可以逐个规则进行遍历，单个规则可以使用split(“,”)来分割特征和结论。

增加规则：打开规则文档，缓存读取规则，待接收新的规则后，使用append将新规则增补到老规则后方，再对规则文档进行覆盖写入。

修改规则：打开规则文档，缓存读取规则，待接收到需要修改的规则序号后，使用列表指针对被改规则进行赋值，再把缓存信息覆盖写入到文档中。

删除规则：打开规则文档，读取规则，接收到要删除的规则序号后，使用pop函数移除掉该规则，最后覆盖写入文档。

规则匹配：

从文档中读取完规则后，切割特征和结论，按相同顺序分开存储，待接收到用户输入的特征后，将用户特征与单条规则特征进行集合比较，若用户特征中完全包含该条规则的特征，则表示匹配该规则，将该条规则的结论增补到用户特征中，将作为新的特征继续后续集合比较，直到规则库中不存在还没有比较的规则。

由于是按照顺序进行比较，在规则库中较后的结论总是会被较后增补到用户特征中，当规则库中规则顺序严格正向时，最后增补到用户特征的结论也就肯定是最终结论，即使规则库没有确切的动物名字，也能匹配到库内最恰当的结论。

但如果规则的顺序比较混乱时，得出正确结论后，也可能会被后加的规则给覆盖，所以规则库内理应有个特定的排序标准；但这只是个简单系统的设计，我就假定规则库已排序了。

结：

最后完成了一个十分简单的动物识别系统，同时也十分的粗略，因为在编写代码时是先完成查增删改功能，再去做匹配的功能，导致我在前期就没有考虑好程序效率的问题，最后这个程序其实有很多我能想到的、可优化的地方，例如规则文件反复开启关闭的问题、代码冗余的问题，但实在是牵一发动全身，不好改动。