Lab 2：一阶逻辑归结算法

实现一阶逻辑归结算法，在给定知识库（KB），对查询（QUERY）应用归结算法。在给出的3个`test{x}.txt`文件中，以如下格式记录了知识库和查询的子句：

```
KB:
Clause1
Clause2
...
QUERY:
Clause query
```

文本`KB:`表示此行以后，`QUERY:`以前的每一行表示一个子句。`QUERY:`以后的子句代表一个查询。

注意：查询已经转换为其否定的形式，不需要做否定转换，就可直接与数据库内容进行归结。

子句形式

本次实验中所有子句已转化为析取式，不包含∀和∃。其中所有子句以字符串方式存储，单一谓词子句直接记录其表达式，多谓词子句以元组形式存储并以字符串形式记录，例如：

- 子句：$A(tony)$ 表示该子句包含1个1元谓词：`A(tony)`。
- 子句：$(A(x), S(x), C(x))$ 表示该子句包含3个1元谓词，含以上且均以析取符号连接，该子句在数学上的表达式是：$A(x) \vee S(x) \vee C(x)$。

谓词、变量与常量

本次实验中，我们约定：

1. 谓词符号第一个字符采用大写字母，其余字符为英文字符：如$A(x), S(x), GradStudent(x)...$等。
2. 变量符号均用**单个**小写字母表示，如$x, a, b...$等。
3. 常量符号均用**多个**小写字母表示，如$tony, aa...$等。
4. 否定符号使用'~'表示。

实验要求：

1. 使用单独的python文件实现一个归结算法类，在给出的`main.py`文件中调用该类，文件名和类名任意，下面我们将以`my_Predicate.py`作为文件名，`Sentences`作为类名为例进行进一步描述，可供参考。
2. `Sentences`类实现`__init__(self, path)`：根据传入的文件路径，读取知识库和查询，并保存到类内部，查询以转换为其否定形式，可直接加入知识库。
3. `Sentences`类除去`__init__()`外必须要实现2个类方法：
    1. `resolution()`：归结当前类存储的知识库，记录每一步归结的结果。
    2. `reindex()`：从知识库归结后的最终状态出发，逆推`resolution()`中需要用到的步骤，并按顺序打印归结步骤。
4. 打印归结步骤格式：
    - 例子：

```
# 假设数据库内有子句
1. (P(x),Q(g(x)))
2. (R(a),Q(z),~P(aa))

# 归结时要打印的内容
R[1a,2c](x=aa) (Q(g(aa)),R(aa),Q(z))
```

其中：“1a” 表示第一个子句(1-th)中的第一个 (a-th)个原子公式，即P(x)；“2c”表示第二个子句(1-th)中的第三个(c-th)个原子公式，即~P(a)。做这一步归结时，需要使用的MGU算法，找到两个子句的最一般合一项。

5. 具体采用的编程方法（面向过程/面向对象）不作限制，允许略微修改main.py的内容（包名、类名、函数名等）。要求在完成归结推导后，可以显示推导过程，且结果正确。
6. 在实验报告中需要描述MGU算法的伪代码、归结过程的伪代码，以及归结过程中使用到的数据结构。**禁止直接打印每一步的推导结果！！！**在实验报告中展示归结的运行过程，并粘贴你认为有必要的**核心代码。**

> HINTS：
>
> 1. 程序运行的主程序main.py已经给出，具体调用Sentence类的过程可参考该文件。
> 2. code.zip文件中包含了main.py和需要完成的3个推导作业。
> 3. 实验报告模板请参考超算习堂-参考材料-《人工智能实验报告模板.docx》

提交：

1. 将所有文件打包成一个压缩包，压缩包命名为：“学号_姓名_作业编号”，例：20240312_张三_实验2。
2. 压缩包内包含：code文件夹和实验报告PDF文件。
    - code文件夹：存放实验代码；
    - PDF文件格式参考超算习堂上的模板，主要描述实现类及其类方法的过程。
3. 截至日期：**2025年3月23日晚24点。**
4. 提交地址：超算习堂