

实验报告

141242026 刘驭壬

1. 实验环境

python 3.5

2. 实验进度

我独立完成了作业要求的所有内容（未参考任何他人和网上的代码）：

- 实现了合一函数
- 实现了归结消解的宽度优先搜索策略
- 实现了归结消解的成组支持+单子句优先策略
- 两个策略的运行步骤与结果均记录在 txt 文件中

3. 实验结果

合一中间步骤和消解路径、最终结果在 bfs_route.txt 和 dfs_route(成组支持加单子句优先).txt 中。结果可查阅这两个文件。

4. 效率分析

由运行结果可以看出成组支持加单子句优先策略只用了 10 步就得到了最终答案，而 bfs 使用了 44 步消解。

5. 问题求解

我们要手动化简题目为如下七个表达式，：

首先，做如下定义：

拿到学位:S, 找到工作:W, 是幸福的:H, 努力学习:E, 背景好:R, 李一天:a

根据题中所给条件并对结论取反，我们得到 7 个表达式：

$$\neg S(X) \vee \neg W(X) \vee H(X)$$

$$\neg E(X) \vee S(X)$$

$$\neg R(X) \vee S(X)$$

$$\neg E(a)$$

$$R(a)$$

$$\neg R(X) \vee W(X)$$

$$\neg H(a)$$

然后定义输入文件格式：

7 表示七个表达式，以下每行均对应定义的七个表达式。

```
7
0 S X#0 W X#1 H X
0 E Y#1 S Y
0 R Z#1 S Z
0 E a
1 R a
0 R A#1 W A
0 H a
```

main 函数如下图，程序先读入所有数据，再分别用 bfs 和 dfs(成组支持加单句子优先)两种策略跑。

```
from expr import Expr, Exprs
from bfs import bfs
from dfs import dfs

if __name__ == '__main__':
    # 记录输入信息
    file = open("input.txt", "r")
    list_lines = file.readlines()
    list_exprs = []
    num = int(list_lines[0])
    for i in range(1, num+1):
        tmp = Exprs(list_lines[i])
        list_exprs.append(tmp)
    for x in list_exprs:
        x.print()
    bfs(list_exprs)
    S = [list_exprs[-1]]
    dfs(S, list_exprs)
```

文件结构如下图：



bfs.py 中实现了 bfs 策略

dfs.py 实现了 dfs(成组支持加单子句优先)策略

expr.py 中定义了表达式和表达式组（表达式的并）的数据结构

resolution.py 中定义了消解函数

unify.py 中定义了合一函数

bfs_route.txt 中记录了 bfs 的合一步骤与搜索记录

dfs_route(成组支持加单子句优先).txt 中记录了 dfs(成组支持加单子句优先)的合一步骤与搜索记录

input.txt 中记录了输入文件