**人工智能上机实验报告**

|  |  |  |
| --- | --- | --- |
| 课程名称：人工智能 | 班级：计科2102 | 实验日期：2023.11.06 |
| 姓名： 薛若雨 | 学号：2112190308 | 指导教师： 万庆 |
| 实验序号：二 | | 实验成绩： |
| 一、实验名称  **知识表示与推理-Resolution Theorem Prover** | | |
| 二、实验目的及要求  1、熟悉Knowledge representation/base的基本概念和表示法；  2、明确命题逻辑和一阶逻辑的运算规则和推理方法；  3、熟练使用Resolution算法自动化推理一阶逻辑问题。 | | |
| 三、实验环境  Python语言 | | |
| 四、实验内容  设计Two-Pointer resolution算法并编写一个计算机程序，该程序能探索并解决至少以下两个课堂讨论过的一阶逻辑问题：   1. Howling Hounds    1. All hounds howl at night    2. Anyone who has any cats will not have any mice    3. Light sleepers do not have anything which howls at night    4. John has either a cat or a hound   Prove: If John is a light sleeper, then John does not have any mice   1. Drug dealer and customs official    1. The customs officials searched everyone who entered the country who was not a VIP    2. Some of the drug dealers entered the country, and they were only searched by drug dealers    3. No drug dealer was a VIP    4. Some of the customs officials were drug dealers   提示:  你需要至少考虑以下一些问题：   1. 如何将一阶逻辑中使用的逻辑符号在程序中表示出来？ 2. 如何设计unification算法？ 3. 如何实现resolution算法中的substitution功能？ 4. 如何设计算法的停机标准？   实验提交：  将所有内容打包成.zip文件，要求：   1. 文件名：学号-班级-姓名.zip； 2. .zip内包含：你全部的代码和程序（关键算法步骤需注释），实验报告.docx，README.txt等。其中README.txt描述你的代码运行环境和方法； 3. .zip文件在10月27号晚上8点前提交。 | | |
| 五、实验方法与算法设计  如何定义输入：  在主程序中，定义了两个知识库clauses1和clauses2和两个查询句子query1和query2。  程序中逻辑Clauses如何表示：  对于clauses1：   1. Howl ∧ have\_hound 2. have\_cats ∧ ¬have\_mice 3. ¬light\_sleeper ∧ howl 4. ¬have\_cats ∧ have\_hound 5. have\_cats ∧ ¬have\_hound 6. john\_has\_cat ∧ ¬john\_has\_hound 7. ¬john\_has\_cat ∧ john\_has\_hound 8. john ∧ light\_sleeper   对于clauses2：   1. searched\_by\_customs ∧ ¬drug\_dealer ∧ VIP 2. ¬searched\_by\_customs ∧ drug\_dealer 3. ¬drug\_dealer ∧ VIP 4. searched\_by\_customs ∧ ¬drug\_dealer 5. ¬searched\_by\_customs ∧ drug\_dealer ∧ ¬VIP   算法原理：  主要原理是通过反证法。首先将待证明结论的否定形式加入到知识库中，然后利用合取消解规则对知识库中的子句进行合并和简化，直到得到一个空子句或者无法再简化为止。  在这段代码中，Atom类表示逻辑命题中的原子命题，Clause类表示合取范式中的一个子句。unify函数用于对两个文字进行统一，substitute函数用于对文字进行替换，resolve函数用于对两个子句进行合取消解操作，而resolution函数则是整个Resolution算法的实现。  停机准则：  当无法再产生新的子句时或者得到一个空子句时，算法停止。当没有新的子句产生时，或者新的子句为空时，算法会停止执行。 | | |
| 七、实验结果  问题一：  运算总步数：3  输出结果如图所示：    问题二：  运算总步数：2  输出结果如图所示： | | |
| 八、实验总结  优点：   1. 能够清晰地表示事实和规则，便于逻辑推理和推断。 2. 适用于处理复杂的逻辑关系和推理过程，能够应对较为复杂的推理任务。   缺点：   1. 对于过于复杂的任务，算法的效率不高，推理过程复杂度高。 2. 无法处理不确定性信息，对于模糊逻辑和概率推理能力有限。   未来改进的地方：   1. 加强在该方面的学习能力，多了解各种不同的算法 2. 提高算法推理效率，优化算法实现，减少推理过程中的冗余计算。 | | |
| 附录：  问题一：  第一步：  第二步：    第三步：      问题二：  第一步：    第二步： | | |