

形式语言与自动机 大作业

“

丁豪 人工智能学院

181220010@smail.nju.edu.cn

实验完成

完成了全部任务，必做以及选座

- 任务一

- 任务分析：解析器的任务主要是文件读取和构建图灵机，这主要涉及到 按行读取 操作以及图灵机 结构体 的设计。
- 图灵机结构体：由两部分组成。第一部分是根据文件产生的 $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ 。第二部分是运行时ID，包括状态、每一条纸带上的内容以及读写头位置。
- 初始化图灵机：
 - 第一步，根据main函数从命令行读取的图灵机文件名称，找到相应的图灵机文件，并按行进行扫描，使用switch对每一行不同情况进行不同操作，最终在读取完成时完成图灵机的构建。
 - 第二步，依据图灵机的初始状态和main函数读到的输入串，构建起图灵机初始ID。
- 错误处理
 - 在第一步中，如果检测到图灵机不合语法，则打印"syntax error"并退出。
 - 在第二步中，如果输入串不合法，则按照"verbose"与否分别进行相应处理。

- 任务二

- 任务分析：模拟器，即将任务一中的图灵机run起来
- Run：循环执行以下内容（根据是否"verbose"，选择性输出相应内容）
 - 检查当前状态+纸带字符组合，是否存在与之匹配的转移函数
 - 如果存在，进行此次转移，continue
 - 如果不存在，halt
- 错误处理：这一步不应存在错误，顶多就是提前Halt

- 任务三

- L_1 : 需要维护 i 和 j 两个值, 统计后面出现的次数是否等于前面出现的次数, 很容易想到用两个栈来实现, 于是我们采用三条纸带, 第一条记录“剩余输入串”, 第二条用作 a 的栈, 第三条用作 b 的栈。在扫描的过程中, 使用不同状态记录当前位于 $[a^i][b^j][a^i][b^j]$ 四段中的哪一段, 然后分别对不同的纸带做“push / pop”操作, 最终检查是否纸带为空, 来作为 accept 还是 reject 的依据。
- L_2 : 需要实现乘法, 一个很简单的方式是使用双重 for 循环。我们使用三条纸带, 第一条用于记录输入串, 第二条记录 m , 第三条记录 n 。在扫描过程中, 使用不同状态记录当前位于 $[11] \times [111] = [111111]$ 三段中的哪一段, 前两段用于记录 mn , 最后一段用于在双重 for 循环的过程中不断向后扫描。如果最终 for 循环结束与 1 串同时, 则接受, 否则拒绝。

问题 & 解决方案

- 如何判断图灵机不合法: 经过群内同学的讨论, 发现图灵机的定义格式较为自由, 每种并不需要严格按照样例中的先后顺序 (当然自己设计的是严格按照这个顺序的), 因此不能在构建的同时进行正确性检查, 最终改为在全部构建完成后进行正确性检查。
- 如何判断输入串不合法: 输入串的不合法情况千奇百怪, 需要全部考虑到十分困难。解决的方法是首先在每一个状态下把合法转移全部写完, 然后利用“补”操作的思想, 把其余所有情况全部转移到拒绝处理状态“reject”, 最终可以实现能够识别非法串并正确地在“reject”状态停机的图灵机。

总结感想

面向图灵机的编程, 需要考虑的内容比面向编程语言的编程要多的多, 任何可能出现的状况都要严格定义以避免出现超出预期的行为。这提示我们在日常编程过程中, 也要养成良好的错误、异常处理习惯, 只实现正确的执行流是远远不够的!

课程和实验建议与意见

课程内容较为丰富, 结合实验可以加深理解。建议每次作业减少一些书面题, 增加一些编程题。