

形式语言与自动机实验报告

181220049 宋磊 人工智能学院

2020 年 12 月 22 日

1 实验描述

实验内容为实现多磁带确定性图灵机，能够读取和解析符合特定语法的图灵机程序，并使用解析得到的图灵机程序对输入串进行判定。

我完成了所有内容，包括编写和解析图灵机文件，对图灵机程序的合法性判断，输入的合法性判断，打印信息以及输出判定结果等。

2 分析与设计思路

具体代码见 *tm.h* 以及 *tm.cpp*。

2.1 数据结构的选择

具体见 *tm.h* 代码。

图灵机类 *Turing* 使用 *set < string >* 存储状态集合 *Q*、输入符号集合 *S*、磁带符号集合 *G* 以及终止状态集合 *F*。使用 *map* 数据结构表示转移函数，*map* 的键为当前状态、当前磁带 (将所有磁头下的符号连接组成字符串作为磁带状态) 状态的结构体，值为下一状态、写入磁带的符号以及移动方向组成的结构体。使用 *string* 表示起始状态和空字符。

对单条磁带，使用结构体 *tape_t* 记录了磁头位置，磁带第一个非空字符的位置以及最后一个非空字符的位置，并使用 *map* 存储所有的磁带上的状态 (使用 *map* 主要是考虑到磁带的索引会出现负数下标并且磁带长度无限，使用数组可能会遇到问题)。

2.2 算法设计思路

具体见 *tm.cpp* 代码。

首先，读取图灵机程序并进行解析和检查工作 (具体见 *Turing* 类中 *build_tm* 方法)。解析过程主要是按行读取，在删除分号之后的字符后，判断是否剩下非空格字符，如果有，则判断该行的类型 (可能为状态集合、输入集合、转移函数等)，判断后更新图灵机存储的内容。检查工作主要是对转移函数的合法性的检查，包括，转移函数的当前状态、下一状态是否属于 *Q*，磁带状态长度是否等于磁带数 *N*，磁带状态中的符号是否属于 *G*，是否存在同一个 *key* 映射到不同值的情况 (非确定性转移)。

之后的模拟过程中 (具体见 *Turing* 类中 *simulate* 方法)，将当前状态以及磁带状态作为键，查询值，如果不存在，则跳出模拟过程，结束模拟。如果存在，则转移，更新所有状态，其中比较重要的是对磁带状态的更新，在移动磁头后，我会根据当前的磁头坐标和磁带记录的字符串起始结束位置范 (*tape_t* 结构体中的 *range* 变量) 进行比较，如果磁头超出范围，则增大范围，如果小于范围，则判断两端是否存在空字符，如果存在则缩小范围直到不存在空字符。

3 遇到的问题及解决方案

遇到的问题主要是在编写图灵机程序时，容易漏写一些转移函数，使得在一半过程中就退出。解决方法使用了 `--verbose` 参数可视化每一步转移过程，根据退出时的状态增加新的转移函数，不断重复。在这个过程中也想起了 jyy 老师经常说编程时基础设施 (`--verbose` 可视化图灵机运作过程) 的重要性，能够节约很多调试的时间。

4 总结感想

在完成实验的过程中，我对图灵机的运作方式的理解进一步加深，并且复习了很多图灵机的知识点。

这是一门比较硬核同时也十分重要的课程，学习过程中我了解到了 FA、PDA、TM 以及它们对应的语言，以及很多计算机理论方面的知识点，希望在之后在面临实际问题时，也能想到用自动机的方式进行建模和分析，用来帮助之后的研究工作。

最后也十分感谢老师和助教的辛苦工作。