软件安全实验四　恶意软件检测方法

1170300728 汤添凝

一、实验项目描述

1、基于特征匹配的误用检测技术原理和方法

（1） 掌握多模式匹配方法

（2） 掌握基于双数组-自动机原理的多模式匹配原理

2、基于双数组-AC 算法的多模式特征匹配算法实现

（1） 双数组-自动机的预处理：Next 表、Base 表、Check 表、失效函数、输出函数构建

（2） 双数组-自动机的特征扫描流程

（3） 合理的数据结构

3、利用构建的自动机扫描目标文件

（1）扫描文件发现命中某个模式，需明确给出命中的模式和该模式在文件中的位置

（2）如命中多个模式，需全部列出

二、实验要求

1、实验数据准备。实验数据应简单实用：利用 ASCII 字符集做为输入集，不考虑多字节编码的中文、英文字符集。模式和待匹配文本可以只包含英文。

2、程序的输入部分（模式、待匹配文本）需以文件方式提供或者其它非固定的输入途径。

3、实验结果和实验数据一起给出：包括设定的模式有哪些？Next 表、Base 表、Check 表是什么？待匹配文本是什么？输出结果是什么。注意：仅给出匹配结果是不够的，必须在输入多模式后，给出 Next 表、Base 表、Check 表内容，Next 表中空间占用的百分比(Next 表中状态占用的空间/Next 表大小)，然后输入待检测文本，再输出检测结果。

4、程序本身需要提交。

三、实验结果（将来需体现在实验报告中）

1、程序的流程图

2、列出构建自动机所定义的数据结构，简单说明其功能

我使用的语言是 python，自动机中包含 5 个列表，分别为 Next 表、Base 表、Check 表、失效函数、输出函数。

其中，Next 表为一个 2*(256*x)的二维列表，两个属性分别为状态值（int）和转入字符（str）；

Base 表为一个 2*n 的列表，两个属性分别为状态值（int）和 base 数值（int），n 为转向函数中非叶节点个数

Check 表为一个 2*m 的列表，两个属性分别为状态值（int）和前驱状态值（int），m 为状态数量

Failure 函数为一个 2*m 的列表，两个属性分别为状态值（int）和失效后的转向状态值（int）

Output 函数为一个 2*k 的列表，两个属性分别为状态值（int）和该状态需要输出的所有字符串（list<str>），k 为字符串个数

3、根据这些数据结构说明 Next 表、Base 表、Check 表、失效函数、输出函数的构建过程

3.1 nexts 数组

遍历每层的字符的同时，记录其对应的状态值，状态值计算如 1.2 所述。在处理子结点列表时，对第一个子结点 ch 而言，其在数组中的偏移值为数组中的最左空位 index，后续结点 ch’的偏移值为 index + ch’ - ch。

3.2 check 数组

在处理子结点列表时，第一个子结点 ch 的父字符对应的结点即为所有子结点的父结点，因此该列表中的结点的 check 值均为父结点状态。

3.3 base 数组

若父结点状态为 s，第一个子结点的输入字符为 ch，状态为 t，设 t 在数组中的偏移为 t_index。根据 AC 算法，计算式为 t = next[s + base[s] + ch]。

则可得到 base[s] = t_index – s – ch。因此，每当处理子结点列表时，可计算父结点的 base 值。

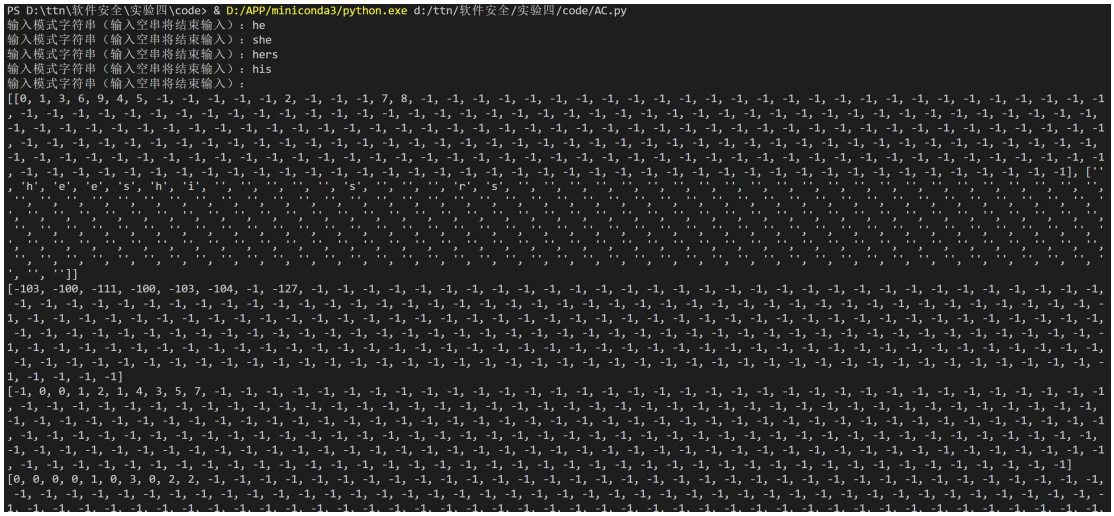
3.4 failure 函数

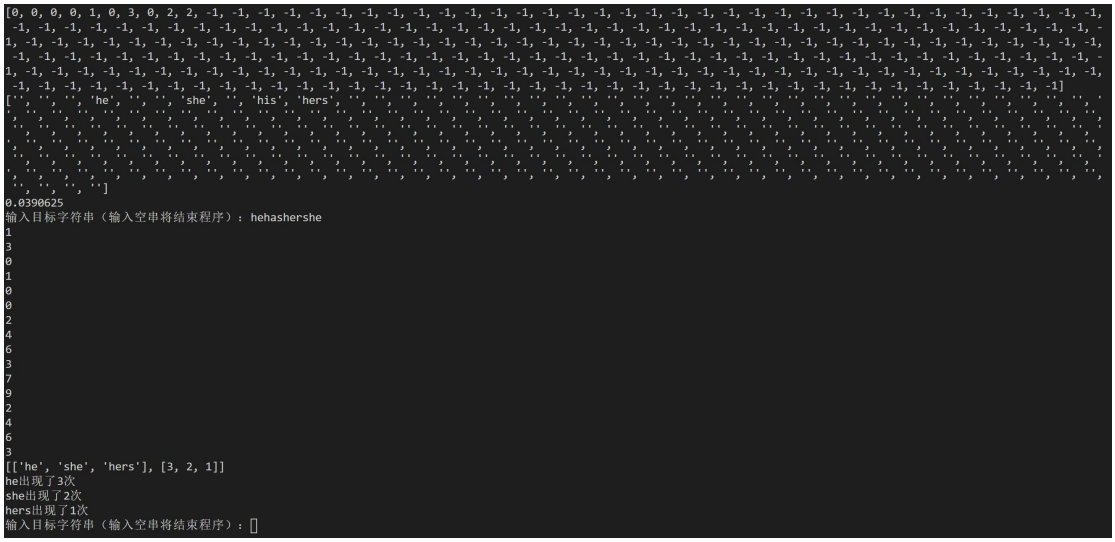
深度为 1 时，结点的 fail 值为 0。深度大于 1 时，设结点为 t，其父结点为 s，输入字符为 ch。则若 goto(fail[s], ch) != -1，fail[t] = goto(fail[s], ch)；否则 fail[t] = 0。

3.5 output 函数

当到达模式串的最后一个状态结点时，记录此时的状态-模式串。构建 fail 数组完毕后，若存在某个接收状态 t 满足 fail[t]也是接收状态，则将 fail[t]对应的模式串集合合并到 t 的模式串集合中。

4、实验结果





利用率为 0.0390625 = 10/256，由于数据量较小，利用率很低，但如果数据量提升的话，利用率将大幅提高。