云南大学数学与与统计学院 上机实践报告

课程名称: 信息论基础实验	年级: 2013	上机实践成绩:
指导教师: 陆正福	姓名: 金洋	
上机实践名称: 唯一可译码的判决算法实验	学号: 20131910023	上机实践日期:
		2016/4/28
上机实践编号: No. 3	组号:	上机实践时间: 8:34

一、实验目的

理解唯一可译码的判决算法

二、实验内容

- 1. 分析唯一可译码的判决算法 (参见英文教材第二版中的习题 5.27)
- 2. 编制程序

三、实验环境

- 1. 个人计算机,任意可以完成实验的平台,如 Java 平台、Python 语言、R 语言、Matlab 平台、Magma 平台等。
- 2. 对于信息与计算科学专业的学生,建议选择 Java、Python、R 等平台。
- 3. 对于非信息与计算科学专业的学生,建议选择 Matlab、Magma 等平台。

四、实验记录与实验结果分析

(注意记录实验中遇到的问题。实验报告的评分依据之一是实验记录的细致程度、实验过程的真实性、实验结果的解释和分析。**如果涉及实验结果截屏,应选择白底黑字。**)

五、实验体会

(请认真填写自己的真实体会)

1. 引入定义

定义:如果编码 X 的取值空间中的每个元素映射成 D^* 中不同的字符串,即

$$x \neq x' \Longrightarrow C(x) \neq C(x')$$

则称这个编码是非奇异的。

定义:编码 C 的拓展 C^* 是从 X 上的有限长字符串到 D 上的有限长字符串的映射,定义为

$$C(x_1x_2\cdots x_n)=C(x_1)C(x_2)\cdots C(x_n)$$

其中 $C(x_1)C(x_2)\cdots C(x_n)$ 表示相应码字的串联。

定义:如果一个编码的拓展编码是非奇异的,则称该编码是唯一可译的。

2.原理介绍

- (1)考察 C 中所有的码字,若 Wi 是 Wj 的前缀,则将对应的后缀作为一个悬空后缀码放入集合 Fi+1 中:
- (2)考察 C 和 Fi 俩个集合,若 Wi ∈ C 是 Wj ∈ F 的前缀或 Wi ∈ F 是 Wj ∈ C 的前缀,则将相应的后缀作为悬空后缀码放入集合 Fi+1 中:
 - (3)F=∪Fi 即为码 C 的悬空后缀集合;
- (4)若 F 中出现了 C 中的元素,算法终止,返回假(C 不是唯一可译码);否则若 F 中没有出现新的元素,则返回真。
- 3.程序实现

UDC.java

```
package IT3;
import java.util.ArrayList;

public class UDC {

    private static boolean result=true;//默认是可唯一译码

    /**

    * 比较两个字符串,若相等,即不可能是唯一可译码,使 result 为假。

    * 若满足前缀关系,则返回后缀。

    * 若无以上关系,返回真

    * @param str1

    * @param str2

    * @return

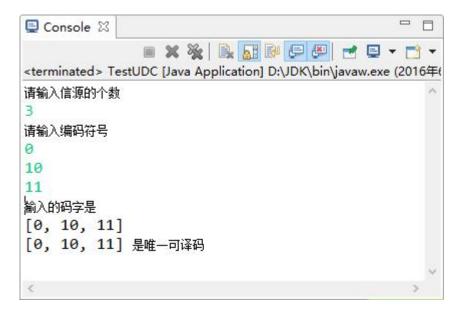
    */
    private static String compareString(String str1,String str2)
    {

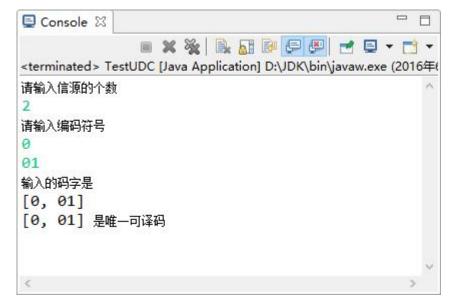
        String theResult=null;
```

```
if(str1.equals(str2)){
           result=false;
       if(result)
       {
           if(str1.startsWith(str2))
           {
              theResult=str1.substring(str2.length(),str1.length());
           if(str2.startsWith(str1))
              theResult=str2.substring(str1.length(),str2.length());
       }
       return theResult;
   }
   /**
    * 比较两个集合。若有相同的元素,则不是唯一可译码。使 result 为假。
    * 若其中有元素满足前缀关系。则提取后缀添加到 mylist 中。
    * @param a
    * @param b
    * @param endString
   private static void compareList(ArrayList<String> a,ArrayList<String>
b,String endString)
   {
       boolean flag=false;
       String aa;
       String bb;
       cp: for(int i=0;i<a.size();i++)</pre>
           for(int j=0;j<b.size();j++)</pre>
              aa=a.get(i);
              bb=b.get(j);
              endString=compareString(aa,bb);
              if(!result)
              {
                  break cp;
              if(endString!=null&&!b.contains(endString))
                  b.add(endString);
                  flag=true;
                  break cp;
              }
           }
```

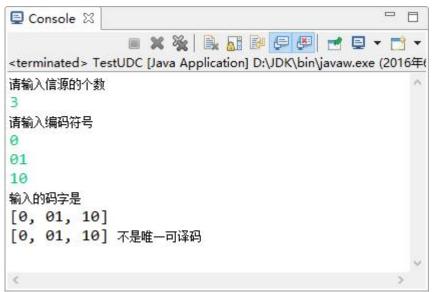
```
}
       if(flag)
       {
          compareList(a,b,endString);
       }
   }
    * 对一个 String 集是否为唯一可译码给出判决
    * @param ccString
    * @return
    */
   public static boolean compare(ArrayList<String> ccString){
       String endString=null;//用来引用要加入 <u>mylist</u>中的后缀。
       ArrayList<String> myList=new ArrayList<>();
       //比较所给字符集中任何两个字符,将满足前后缀的字符添加到 myList 中
       cp: for(int m = 0; m< ccString.size(); m++)</pre>
       {
           for(int j = m+1; j< ccString.size(); j++)</pre>
           {
              String st1=ccString.get(m);
              String st2=ccString.get(j);
              endString= compareString(st1,st2);
              if(!result)
              {
                  break cp;
              if(endString!=null&&!myList.contains(endString))
              {
                  myList.add(endString);
              }
           }
       }
       compareList(ccString,myList,endString);
       return result;
   }
}
TestUDC.java
package IT3;
import java.util.*;
```

```
public class TestUDC
{
   public static void main(String[] args){
       ArrayList<String> sourceCode=new ArrayList<String>();//存储输入的码
符号
       /*输入字符*/
       if (args.length==0){
          sourceCode.clear();
          Scanner input = new Scanner(System.in);
          System. out. println("请输入信源的个数");
          int count = input.nextInt();
          System. out. println("请输入编码符号");
          for(int i = 0; i< count; i++)</pre>
              sourceCode.add(input.next());
          System.out.println("输入的码字是\n"+sourceCode);
       }else {
          Collections.addAll(sourceCode, args);
       }
       String result= UDC.compare(sourceCode)?"是唯一可译码":"不是唯一可译
码";
       System.out.println(sourceCode+" "+result);
       System.exit(0);
   }
}
```









六、参考文献

- 1. (主讲课英文教材) 英文教材第二版中的习题 5.27
- 2. (如有其它参考文献,请列出)

(提示:参考北京邮电大学信息论方面的实验指导书)

3. Deonet. 唯一可译码判决准则 [EB/OL]. http://blog.csdn.net/u013290075/article/details/51707437, 2016.