

正则表达式的动态集合扩展与实现^①

程元斌

(江汉大学 数学与计算机科学学院, 武汉 430056)

摘 要: 正则表达式是表达复杂检索条件的有力工具, 但是现有的正则表达式表达规则仍有不少不足之处, 例如不能用简单的形式表达“任何一种水果”这样的意思. 提出一种基于集合扩展的架构, 使得可以在正则表达式中包含用户自定义集合, 用以表示“此处可以是集合中的任一成员”这样的含义. 提出的这种集合不仅是用户自定义的, 而且是动态的、按名称访问的.

关键词: 正则表达式; 集合扩展; 动态扩展

Extension and Implementation of Regular Expression's Dynamic Set

CHENG Yuan-Bin

(School of Mathematics & Computer Science, Jiangnan University, Wuhan 430056, China)

Abstract: Regular expression is a powerful tool to express a complex retrieve condition. However, some lacks that still exist in the current regular expression. For example, we can't express such a meaning like “any one kind of fruit” in a simple form. In this paper, an extension based on set which can be defined dynamic by user is proposed. By this extension, this paper can express the meaning “here can be any member of the set” simply in a regular expression.

Key word: regular expression; set-based extension; dynamic extension

1 引言

正则表达式在计算机领域的应用十分广泛, 而且随着计算机应用领域的不断扩大与深入, 正则表达式越来越倍受重视, 成为信息检索、信息安全等领域不可或缺的有力工具^[1-3].

最早的正则表达式定义了其最简形式及三种运算. 最简形式就是字母表中的一个字母, 包括空符号 ϕ 及空串 ϵ ; 三种运算包括连接运算“.”、或运算“|”及闭包运算“*”. 理论上, 由最简形式及三种运算就可得到任意一个字符串, 但在实用中表达起来很不方便. 例如, 超级链接的正则表达式就需要由一百多个符号组成. 因此, 几乎所有的正则表达式实现方案都对其表达方法作了基本相似的扩展. 例如, 美国微软公司的 .net framework 开发平台就额外定义了 12 种字符类表达方法(其中包含字符集合), 11 种用以表达匹配前导元素出现次数的限定符, 10 种关于子表达式运用的“分组构造”, 以及用来指定匹配必须出现在字符串

中的位置的“原子零宽度断言”等一系列表达方法^[4-7].

对正则表达式语法的这些扩展大大简化了正则表达式的运用, 可以说在实际上大大提高了正则表达式的威力. 例如, 超级链接的正则表达式现在只需要这样表达:

`href\s*=\s*"[\^"]*"`

总共只有 23 个符号!!!

除此之外, .net framework 的正则表达式系统包中还加入了一些相关功能, 例如替换功能, 在检索的同时, 可以顺带进行更改日期格式、从字符串中剥离无效字符等常见工作^[8,9].

然而即便如此, 在实际应用中, 仍然常常感到很不方便. 其中尤其是 .net framework 定义的集合仅限于 UNICODE 字符集中的子集, 不能表达日常事务的集合. 例如, 当我们想表达“任何一种水果”这样的意思时, 还是只能在正则表达式中一个个的枚举出每种水果的名称. 而且实际中类似情况出现的机会是非常

^① 收稿时间:2012-02-23;收到修改稿时间:2012-03-27

多的. 针对这种情况, 本文提出了一种进一步的扩展方案, 以进一步增强正则表达式的实际效能.

2 用于正则表达式扩展的集合

正则表达式的扩展主要是表达形式的扩展. 上一节已经对目前已有的对正则表达式的扩展及其不足作了一个简要的说明. 本文将主要针对这种不足, 探讨如何将一般意义上的字符串集合——而不是简单的连续字符的集合——引入正则表达式中.

就集合的一般概念而言, 集合的元素可以是任意数据类型, 既包括各种简单数据类型, 也包括文件、对象、子集等等. 不过, 由于目前正则表达式一般都用于与字符串数据的匹配检索相关的领域, 所以, 本文所述的集合, 仅限定在字符串为元素的集合范围内.

另外, 本文所述的集合, 应当具备以下特性:

- a) 每个集合有一个在作用域范围内唯一的名称;
- b) 可用户自定义;
- c) 可动态定义及修改;

特性 a 保证对集合的“按名访问”, 特性 b 保证用户可以按需要定义自己的集合, 特性 c 保证集合可以在程序运行过程中进行动态定义、运算, 例如可以根据程序运行需要, 动态的定义一个集合, 并从数据库中获取数据添入集合, 也可以让用户从已有的集合中取出一部分元素或者将几个集合合并到一起, 等等.

在上述集合定义的基础上, 正则表达式将可以对其“按名引用”. 这样, 当我们想在正则表达式中表达“任何一种水果”这样的意思时, 就可以先依据相关的数据库, 定义(静态地或动态地)出水果集合, 然后就可以十分方便得给出相应的正则表达式了(参见第 4 节: 应用示例).

3 实现方案

为了实现上述扩展, 在现有的程序开发与运行机制下, 首先需要定义一个集合类. 下面是一个类 C++ 语言的这样的集合类定义的核心部分.

```
class reex{
private:
    int m;        //当前元素个数
    string members[]; //存放元素的动态数组
    string set_name;    //集合名称
    sort( );        //排序
```

```
public:
```

```
    add_members(string a[], int n); //添加数组 a 到集合
    add_member(string s);           //添加一个元素
    delete_member(string s);        //删除一个元素
    reset(string a[], int n );      //用数组 a 重置集合
    setup_name(string name);        //设置集合名称
    string find(string s);           //在集合中检索元素 s
    deleteall( );                   //删除所有元素
}
```

上述定义中, 排序函数 `sort` 之所以定义为私有函数, 其中隐含着这样的思想: 数组的有序性(它是快速检索的基础)由类本身来保证, 即外部对类的任何公有成员函数的调用都不会改变数组的有序性, 所有的公有成员函数在必要时都会自动调用 `sort` 函数进行排序. 另外, 对于那些不支持动态数组但是支持指针的语言, 例如 VC++, 可用链表代替数组 `members`.

其次, 还需要定义一个正则表达式类, 对正则表达式进行定义、解析、运行. 下面给出与集合扩展相关的部分.

```
class regular_expression{
private:
    reex set_array[]; //动态的 reex 类型数组
    int na;            //数组元素(集合)的个数
    .....

public:
    add_set(string a[], int n, string set_name);
    add_to_set(string s, string set_name);
    delete_from_set(string s, string set_name);

public:
    add_set(reex set);
    delete_set(string set_name);
    int find(stream f, string regu_exp);
    .....
}
```

上述定义中, 第一组公有函数的功能从函数名及参数不难推断, 之所以与 `reex` 集合类中的公有函数功能相似, 主要是为了方便用户: 用户可以直接运用这组函数创建集合、添加或删除集合元素, 而不必了解集合类 `reex`. 另外, 在第二组公有函数中重载了 `add_set` 函数, 这样, 用户可以将已有的 `reex` 集合类对象加入到当前的 `regular_expression` 对象中, 也可以先

创建 reex 集合对象,再用 add_set 函数进行添加.总之,用户可以根据具体情况灵活运用这些函数.

当应用程序要使用带集合扩展的正则表达式时,首先要创建一个 regular_expression 对象,其次调用其中的相关函数创建一个命名集合,然后以正则表达式为参数,调用 find 函数进行检索.find 函数在对正则表达式的解析-运行过程中,如果发现其中含有集合类,则根据集合的名称找出其在集合数组中的定位,然后可调用该集合对象的 find 函数.

4 应用示例

设有存放着水果名称数据的数组 fruit, 下面的例子从字符流 f 中检索包括水果类关键字的字符串.代码如下:

```
.....  
regular_expression re;           //语句 1  
re.add_set(fruit, GetSize(fruit), "fruit"); //语句 2  
keys = keys1 + " \p[fruit]" + keys2; //语句 3  
result = re.find(f, keys);        //语句 4  
.....                          //语句 5
```

其中,语句 2 中的 add_set 函数将执行一系列动作:首先给集合数组 set_array 添加一个元素,然后,调用该元素对象的相关成员函数,将其命名为“fruit”,并将 fruit 数组中的所有元素添加到 members 成员数组中,最后进行排序以备快速检索.

语句 3 示意了集合在正则表达式中的表达方法.字符“\”表示紧跟其后的字符(除“\”外)是一个命令字符.不言而喻, p 表示集合,紧跟其后的中括号中的字符串是集合的名称.显然,语句 4 中的 find 函数很容易从 keys 参数中检出集合“fruit”,然后即可从集合数组 set_array 中检出名为“fruit”的元素的下标,然后 可调

用该元素对象的成员函数 find 进行匹配运算.

5 结语

本文通过集合扩展给出了一种方法,解决了在正则表达式中包括事先并不确定的某一类可替换数据的问题.应当指出的是,本文仅仅给出了解决问题的一种可行方法,这种方法的性能如何?是否有更好的可替代方法?以及如何提高整个正则表达式的计算效能,则都是需要进一步研究的课题.

参考文献

- 1 张树壮,罗浩,方滨兴.面向网络安全的正则表达式匹配技术.软件学报,2011,22(8):1838-1853.
- 2 王杰,石成辉.基于正则表达式的动态应用层协议识别方案.计算机工程与应用,2010,146(18):103-106.
- 3 胡军伟,秦奕青,张伟.正则表达式在 Web 信息抽取中的应用.北京信息科技大学学报(自然科学版),2011,126(6):86-89.
- 4 MSDN,字符类,http://msdn.microsoft.com/zh-cn/library/bw873z(v=vs.90).aspx.
- 5 MSDN,原子零宽度断言,http://msdn.microsoft.com/zh-cn/library/h5181w5w(v=vs.90).aspx.
- 6 MSDN,限定符,http://msdn.microsoft.com/zh-cn/library/3206d374(v=vs.90).aspx.
- 7 MSDN,分组构造,http://msdn.microsoft.com/zh-cn/library/bs2tw2tah(v=vs.90).aspx.
- 8 MSDN,更改日期格式,http://msdn.microsoft.com/zh-cn/library/c8e9427h(v=vs.90).aspx.
- 9 MSDN,从字符串中剥离无效字符,http://msdn.microsoft.com/zh-cn/library/844skk0h(v=vs.90).aspx.
- 2 Blattberg RA, Deighton J. Manage market by the customer equity test. Harvard business review, 1996, July-August: 136-144.
- 3 Barbara D. Using Self-Similarity to Cluster Large Data Sets. Data Mining and Knowledge Discovery, 2003,7.
- 4 蔡颖琨,谢昆青,马修军.屏蔽了输入参数敏感 DBSCAN 改进算法.北京大学学报,2004,3.
- 5 谭颖,胡瑞飞,殷国富.多密度阈值的 DBSCAN 改进算法.计算机应用,2008,3.
- 6 马秀红,宋建社,董晟飞.数据挖掘中决策树的探讨.计算机工程与应用,2004,1.
- 7 曲开社,成文丽,王俊红.ID3 算法的一种改进算法.计算机工程与应用,2003,25.

(上接第 154 页)