维普资讯 http://www.cqvip.com

一种新的模式合一算法*)

王树西　赵星秋　刘瑞林　黄健青
(对外经济贸易大学信息学院　北京 100029)

摘　要　传统的模式合一，使用递归调用的方法，算法的时间复杂度是指数级的，因此，往往容易耗费大量的系统资源，从而造成系统的崩溃。为了解决这个问题，本文提出一种新的模式合一算法，其时间复杂度为线性的。实验结果表明，本算法可以有效地解决原来算法中存在的递归调用问题。

关键词　模式，模式合一

A New Algorithm of Pattern Unification

WANG Shu-Xi　ZHAO Xing-Qiu　LIU Rui-Lin　HUANG Jian-Qing
(Information Academy of the University of Interational Business and Economics, Beijing 100029)

Abstract　Traditional pattrn unification algorithm adopts the recursive method, which time complexity is exponential. Traditional pattern unification algorithm consumes so much system resourch that the system is easy to breakdown. To solve the problem, this paper proposes a new pattern unification algorithm, which time complexity is linear. Experiment result indicates that the new algorithm can successfully solve the recursive problem which exists in customary algorithm.

Keywords　Pattern, Pattern Unification

模式，又称“模板”，是指一个字符串，它由终结符和非终结符构成。模式可以形式化为二元组$\langle \Sigma, V\rangle$，其中，Σ是终结符集合，V是非终结符集合，$\Sigma \cap V=\phi$[4]。

模式合一，又称“合一”，是指对两个模式的耦合，即通过一组确定的规则，用一个模式的子串，代换另外一个模式的变量，使得两个模式完全相同[3]。

传统的模式合一算法，使用递归调用的方法，算法的时间复杂度是指数级的，因此，往往容易耗费大量的系统资源，从而造成系统的崩溃[2]。

本文提出一种新的模式合一算法——“基于图匹配的双模式合一算法”，它的时间复杂度是线性的，从而可有效解决原来算法中的递归调用问题[1]。

本文首先提出一系列相关的定义；然后提出了对单个模式建立索引的算法；在此基础上，提出了模式合一的算法——“基于图匹配的双模式合一算法”，并对该算法进行了分析。实验结果表明，本算法可以有效解决原来算法中的递归调用问题。最后指出了下一步的工作方向。

1　相关定义

①联立合一

一个模式n元组$\langle P_1, P_2, \cdots, P_n\rangle$是另一个模式$n$元组$\langle Q_1, Q_2, \cdots, Q_n\rangle$的联立细化，如果存在一个代换$n$元组$\langle x_1/y_1, x_2/y_2, \cdots, x_n/y_n\rangle$($n\geqslant 1, x_i\in V, y_i\in(\Sigma\cup V)^*$, $1\leqslant i\leqslant n$)，使得把$Q_1, Q_2, \cdots, Q_n$中的所有x_i的出现，都同时替换成y_i，结果恰好得到$P_1, P_2, \cdots, P_n$($1\leqslant i\leqslant n$)。两个模式n元组$\langle P_1, P_2, \cdots, P_n\rangle$和$\langle Q_1, Q_2, \cdots, Q_n\rangle$的公共联立细化，称为它们的联立合一。

②字的偏移量集合

又称为“当前字的偏移量集合”，是指一个字在事实库、规则库中所有偏移量所构成的集合，记作 CurWordOffSet。

例如，如果字(常量)“司”在事实库中的偏移量是“F1.5”、“F2.2”，并且“司”在规则库中的偏移量是“R1.3”、“R2.1”，那么，“司”的偏移量集合是[F1.5, F2.2, R1.3, R2.1]。再例如，如果字(变量)“$_1_$”在规则库中的偏移量是“R1.2”，那么，“$_1_$”的偏移量集合是[R1.2]。

③起始字的偏移量集合

事实库中每条事实的起始字的偏移量，规则库中每条规则的起始字的偏移量，所构成的集合，称为“起始字的偏移量集合”，记作 IniWordOffSet。

例如，如果事实库中有3条事实，2条规则，那么，起始字的偏移量集合是[F1.0, F2.0, F3.0, R1.0, R2.0]。

④终结字的偏移量集合

事实库中每条事实的终结字的偏移量，规则库中每条规则的终结字的偏移量，所构成的集合，称为“终结字的偏移量集合”，记作 EndWordOffSet。

例如，如果事实库中有3条事实，2条规则，并且第1条事实的终结字的偏移量是F1.6，第2条事实的终结字的偏移量是F2.8，第3条事实的终结字的偏移量是F3.9，第1条规则的终结字的偏移量是R1.7，第2条规则的终结字的偏移量是R2.5。那么，在本例中，终结字的偏移量集合是[F1.6, F2.8, F3.9, R1.7, R2.5]。

⑤先前字的偏移量集合

*)本文有关研究得到“中俄经贸合作网”项目资助。**王树西**　博士，讲师，主要研究领域为计算语言学等。**赵星秋**　副教授，硕士研究生导师，主要研究领域为人工智能、数据库等。**刘瑞林**　副教授，硕士研究生导师，主要研究领域为电子商务、数据挖掘等。**黄健青**　副教授，硕士研究生导师，主要研究领域为电子商务等。

维普资讯 http://www.cqvip.com

是相对于“当前字的偏移量集合”而言的。当前字向前回溯，前面一个字的所有偏移量构成的集合，称为“先前字的偏移量集合”。记作 PreWordOffSet。

PreWordOffSet 初始化为 IniWordOffSet。

⑥变量字的偏移量集合

规则库中所有变量字的偏移量所构成的集合，称为“变量字的偏移量集合”，记作 VarOffSet。

例如，如果规则库中有两个变量：“$_1_$”、“$_2_$”，并且“$_1_$”的偏移量为 R1.3，R1.6，“$_2_$”的偏移量为 R2.0，R2.9，那么，在本例中，变量的偏移量集合是[R1.3，R1.6，R2.0，R2.9]。

⑦变量字的最小偏移量集合

“变量字的偏移量集合”中，每个变量字的最小偏移量所构成的集合，称为“变量字的最小偏移量集合”，记作 VarWordMinOffSet。

例如，如果变量的偏移量集合是[R1.3，R1.6，R2.0，R2.9]，那么，变量的最小偏移量集合为[R1.3，R2.0]。

⑧最小偏移量集合

每个字的最小偏移量所构成的集合，称为“最小偏移量集合”，记 wt WordMinOffSet。

例如，如果偏移量集合是[F1.3 F1.5 R1.3，R1.6，R2.0，R2.9]，那么，最小偏移量集合为[F1.3 R1.3，R2.0]。

2 对单个模式建立索引

在实际应用中，需要对单个模式建立索引，称作“常量、变量的一体化索引”[5]。例如，对模式“$_1_$是$_2_$的父亲”建立的索引如下：

$_1_$	P1.0
是	P1.1
$_2_$	P1.2
的	P1.3
父	P1.4
亲	P1.5

本文提出两种对单个模式建立索引的算法，一种是“建立索引链表的算法”，另外一种是改进的算法——“建立索引数组的算法”。

2.1 建立索引链表的算法

```
void fn_vCreatePatternIndexLink(string sPattern, struct stLinkWord
* &pLinkWordHead)
{
    顺序取出模式 sPattern 的每一个字(变量)sWord
    {
        计算出 sWord 的偏移量 sOffset;
        如果 sWord 已经在字链表 pLinkWordHead 中存在，那么，
        将 sOffset 插入到 sWord 的偏移量链表中。
        否则，建立一个新的结点，将 sWord 及其偏移量 sOffset，插
        入到字链表 pLinkWordHead 的尾部
    }
}
```

算法的输入，是模式 sPattern；算法的输出，是字链表的头指针(struct stLinkWord * &pLinkWordHead)；算法的时间复杂度，是 $O(n)$。

算法的主要部分，是将每个字(变量)sWord，及其对应的偏移量 sOffset，插入到的字链表中。

2.2 改进的算法——建立索引数组的算法

```
void fn_vCreatePatternIndexArray(string sPattern, struct stWord
aWordIndex[MAX_WORD_NUM], int& iWordNum)
{
    顺序取出模式 sPattern 的每一个字(变量)sWord
    {
        计算出 sWord 的偏移量 sOffset;
        如果 sWord 已经在数组 aWordIndex 中存在，那么，将 sOff-
        set 插入到 sWord 的偏移量链表中
        否则，将 sWord 及其偏移量 sOffset，按照 sWord 递增的次
        序，插入到数组 aWordIndex 中，iWordNum++。
    }
}
```

算法的输入，是模式 sPattern；算法的输出，是递增排列的字(变量)数组(struct stWord aWordIndex[MAX_WORD_NUM])，以及数组中字(变量)的个数(iWordNum)。

3 基于图匹配的双模式合一算法

3.1 算法的输入

①多元多次模式 sTarget(目标模式)。

②多元多次模式 sText(待合一的文本模式)。

3.2 算法的输入

①判断 sTarget 与 sText 能否合一。

②如果能够合一，得到 sTarget 与 sText 的合一结果。

③sTarget 中不同变量的个数 iVarNum。

④sTarget 中的每个变量，及其分别对应的代换量。

3.3 算法思想

首先，对模式 sText 建立索引；然后，目标模式 sTarget 对这个索引进行检索，从而判断 sTarget 与 sText 能否合一，如果能够合一，那么得到目标模式 sTarget 中变量的个数，并分别得到目标模式 sTarget 中每个变量对应的代换量。

3.4 具体算法

①初始化工作。

对待合一的模式 sText 建立索引。

分别得到“起始字的偏移量集合”IniWordOffSet、“终结字的偏移量集合”EndWordOffSet、“变量字的偏移量集合”VarWordOffSet。

构造“先前字的偏移量集合”PreWordOffSet＝IniWordOffSet(初始化)。

(说明：这里所提到的集合，其类型均为 set(struct stOffset*)。)

将目标模式中变量的个数，记作 iVarNum。iVarNum=0(初始化)。

将目标模式中的每个变量，及其分别对应的代换量。分别表示为两个数组，即：变量数组 aVar[]，变量代换数组 aVarRep[]。

构造布尔型变量 bPreWordIsVar，它表示当前字 sCurWord 前面的字(前接字)sPreWord 是否为变量。bPreWordIsVar=false(初始化)。

构造新的模式 sPattern=sTarget。

②如果 sPattern 为空字符串，那么返回；否则，取出目标模式 sPattern 中的第一个字 sCurWord。sPattern 去除第一个字 sCurWord。

③如果当前字 sCurWord 是常量，那么，转④。否则，如果当前字 sCurWord 是变量，那么，转⑨。

④当前字 sCurWord 是常量。得到集合 CurWordOffSet。

·根据集合 VarWordOffSet，得到集合 VarWordMinOffSet。将集合 VarWordMinOffSet，归并到集合 CurWordOffSet 中。

·如果当前字 sCurWord 前面的字 sPreWord 为变量，即 bPreWordIsVar 的值为 true，那么，根据集合 CrossWordOffSet，得到变量 sPreWord 对应的代换量 sRep。

⑤计算集合 PreWordOffSet、CurWordOffSet 的交集：Cr-

维普资讯 http://www.cqvip.com

ossWordOffSet。

⑥如果集合 CrossWordOffSet 是空集，那么，多模式合一失败，返回 false。

⑦根据集合 CrossWordOffSet，对集合 VarWordOffSet 中的偏移量进行删减，从而得到新的集合 VarWordOffSet。

⑧如果当前字 sCurWord 不是目标模式 sPattern 的最后一个字，那么根据集合 CrossWordOffSet、集合 VarWordMinOffSet，得到新的集合 PreWordOffSet。

转⑩。

⑨当前字 sCurWord 是变量。

·根据集合 PreWordOffSet、集合 EndWordOffSet，得到新的集合 CurWordOffSet、PreWordOffSet、CrossWordOffSet。

·将当前字 sCurWord，加入变量数组 aVar[]，iVarNum ++。

对布尔变量 bPreWordIsVar 进行赋值，令 bPreWordIsVar = true。

⑩考察当前字 sCurWord 在目标模式 sPattern 中的位置。

·如果 sCurWord 是目标模式 sPattern 的最后一个字，那么，计算集合 CrossWordOffSet 与集合 EndWordOffSet 的交集，并记作集合 UnifyResultWordOffSet。如果集合 UnifyResultWordOffSet 为空集，那么多模式合一失败，返回 false。

·否则，如果 sCurWord 不是目标模式 sPattern 的最后一个字，那么，转②。

3.5 算法分析

本算法的特点，是先对一个模式建立索引("常量、变量一体化索引")，构成一个图，然后，目标模式在这个图中进行检索。具体就是：目标模式从图中寻找一个入口，然后检索下去，直到图的末梢。本算法的计算过程，是字级别的计算过程，粒度很细。

3.6 算法测试

测试用例 1：

sTarget = "司马懿是 $_1_ $ 的父亲"

sText = "司马懿是司马昭的父亲"

测试结果：

合一结果 sUnifyResult = 司马懿是司马昭的父亲

变量个数：iVarNum = 2

变量："$_1_ $"，对应的字符串："司马昭"

测试用例 2：

sTarget = "_1_ $ 是 $_2_ $ 的 $_3_ $"

sText = "司马懿是司马炎的祖父"

测试结果：

合一结果 sUnifyResult = 司马懿是司马炎的祖父

变量个数：iVarNum = 3

变量："$_1_ $"，对应的字符串："司马懿"；

变量："$_2_ $"，对应的字符串："司马炎"

变量："$_3_ $"，对应的字符串："祖父"

测试用例 3：

sTarget = "$_1_ $ 是 $_2_ $ 的祖父"

sText = "毛泽东是毛新宇的 $_3_ $"

测试结果：

合一结果 sUnifyResult = 毛泽东是毛新宇的祖父

变量个数：iVarNum = 2

变量："$_1_ $"，对应的字符串："毛泽东"

变量："$_2_ $"，对应的字符串："毛新宇"

测试用例 4：

sTarget = "$_1_ $ 是 $_2_ $ 的 $_3_ $"

sText = "$_4_ $ 是 $_5_ $ 的祖父"

测试结果：

合一结果 sUnifyResult = $_4_ $ 是 $_5_ $ 的祖父

变量个数：iVarNum = 3

变量："$_1_ $"，对应的字符串："$_4_ $"

变量："$_2_ $"，对应的字符串："$_5_ $"

变量："$_3_ $"，对应的字符串："祖父"

结论和下一步的工作 本文首先介绍了模式合一的相关定义、研究现状；然后指出了现有模式合一算法的不足之处；在此基础上，提出了一种新的模式合一算法——"基于图匹配的双模式合一算法"，并对该算法进行了分析；实验结果表明，本算法可以有效解决原来算法中的递归调用问题。

下一步的工作，将把本算法应用于问答系统中，以提高问答系统的性能。

参 考 文 献

1 Lin D, Pantel P. Discovery of Inference Rules for Question Answering. Natural Language Engineering, 2001, 7(4): 343～360

2 王树西，白硕，姜吉发. 模式合一的"斩首"算法及其应用. 计算机工程，2004，21：22～24

3 王树西，白硕，等. 模式合一的"斩首"算法. 见：中国人工智能学会第 10 届全国学术年会论文集(上)，2003，528～532

4 白硕. 大规模内容计算. 语言计算与基于内容的文本处理. 清华大学出版社，2006，33(4)：174～176

5 王树西，白硕. 事实库、规则库的一体化全文索引算法. 计算机科学，2005

(上接第 4 页)

7 Atanassov K T. Intuitionistic Fuzzy Sets [M]. NY: Physica-Verlag, 1999

8 Dubois D, Ostasiewicz W, Prade H. Fuzzy sets: History and basic notions [M]. In: Dubois D, Prade H, eds. Fundamentals of Fuzzy sets Dordrecht. Kluwer Academic Publishers, 2000. 80～93

9 Iczany G. A method of inference in approximate reasoning based on interval valued fuzzy sets [J]. Fuzzy Sets and Systems, 1987, 21 (1): 1～17

10 Gau Wen-Lung, Buehrer D J. Vague sets [J]. IEEE Transactions on systems, Man, and Cyernetics, 1993, 23 (2): 610～614

11 Bustince H, Burillo P. Vague sets are intuitionistic fuzzy sets [J]. Fuzzy Sets and Systems, 1996, 79(3): 403～405

12 Deschrijver G, Kerre E E. On the relationship between some extensions of fuzzy set theory [J]. Fuzzy sets and Systems, 2003, 133 (2): 227～235

13 雷英杰，王宝树，孙金萍. 模糊知识处理与模糊集理论的若干拓展[J]. 空军工程大学学报(自然科学版)，2004，5(3)：40～44

14 雷英杰，王宝树. 拓展模糊集之间的若干等价变换. 系统工程与电子技术，2004，26(10)：1414～1418

15 张江，林华，贺仲雄. 统一集论与人工智能[J]. 中国工程科学，2002，4(3)：40～47