# 多模式合一的"图检索"算法*)

王树西[1,2]　白　硕[1]
(中国科学院计算技术研究所软件研究室　北京 100080)[1]
(中国科学院研究生院　北京 100000)[2]

**摘　要**　多模式合一，又称为联立合一，是一个有着重要研究价值的课题。在问答系统的研究中，多模式合一作为一种新的研究途径，具有较高的应用价值，也因此受到较高的关注和研究。本文首先介绍了多模式合一的相关定义，然后给出了多模式合一的一个具体实例，并对多模式合一的计算过程进行了分析。在此基础上，重点给出了多模式合一的算法——"图检索"算法。实验结果进一步表明，本算法可以有效地解决多模式合一问题。最后，介绍了本算法在中文问答系统中的具体应用。

**关键词**　模式，多模式合一，"图检索"算法，问答系统

## The "Graph Retrieving" Algorighem in Multi-pattern Unification

WANG Shu-Xi[1,2]　BAI Shuo[1]
(Institute of Computing Technology, The Chinese Academy of Sciences, Beijing 100080)[1]
(Graduate Student School, The Chinese Academy of Sciences, Beijing 100080)[2]

**Abstract**　Multi-pattern unification, also called unification, is a high valuable subject. It can be applied in the domain of Question Answering System(QAS). This paper firstly proposes the definition of multi-pattern unification, and proves the quality of it. Then, this paper gives the algorithm of "graph retrieving". The experiment results indicate that this algorithm is effective and can solve the problem of multi-pattern unification. In the end, this paper introduces the application of the algorithm in Question Answering System(QAS).

**Keywords**　Pattern, Multi-pattern unification, Algorithm of "graph retrieving", Question answering system

在基于自由文本的问答系统[1]中，一个较大的挑战在于：用户查询和侯选答案文本经常用不同的词汇进行表述。也就是说，用户查询和侯选答案之间，即使语义上等价，但是从字面上来说，不见得是匹配的。例如，令用户查询"X writes Y"，侯选答案为"X is the author of Y"，从语义上来说，这两个句子是等价的，但是从字面上来说，这两个句子是无法匹配的[3,5]。

这就产生了一个问题：对于用户的查询，如果仅仅采用关键字(词)匹配的方法求解，那么，即使文本中存在答案，也很有可能无法匹配到。

为了解决这个问题，我们引进"多模式合一"的方法。所谓"多模式合一"，又称为"复杂模式合一"、"联立合一"，具体来说，所谓"多模式合一"，就是目标模式与事实库、规则库同时进行合一，从而得到与目标模式相匹配的所有事实，以及与目标模式相匹配的所有规则的右部。"多模式合一"的基础，是"常量、变量一体化全文索引"，也就是混合索引。

## 1　相关定义

### 1.1　联立合一

一个模式 $n$ 元组$\langle P1,P2,\cdots,Pn\rangle$是另一个模式 $n$ 元组$\langle Q1,Q2,\cdots,Qn\rangle$的联立细化，如果存在一个代换 $n$ 元组$\langle x1/y1,x2/y2,\cdots,xn/yn\rangle$($n\geqslant 1$, $xi\in V$, $yi\in(\Sigma\cup V)^*$, $1\leqslant i\leqslant n$)，使得把 $Q1,Q2,\cdots,Qn$ 中的所有 $xi$ 的出现，都同时替换成 $yi$，结果恰好得到 $P1,P2,\cdots,Pn(1\leqslant i\leqslant n)$。

两个模式 $n$ 元组$\langle P1, P2, \cdots, Pn\rangle$和$\langle Q1, Q2, \cdots, Qn\rangle$的公共联立细化，称为它们的联立合一。

### 1.2　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，记做 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]。

### 1.3　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和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"，记做 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]。

### 1.4　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和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"，记做 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

*)本文有关研究得到 973 项目资助(课题名称：大规模文本内容计算；课题编号：2004CB318109)。王树西　博士生，主要研究领域为计算语言学等；白　硕　博士，研究员，博士生导师，主要研究领域为人工智能、计算语言学等。

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]。

### 1.5 先前字的偏移量集合

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”，记做 PreWordOffSet。

PreWordOffSet 初始化为 IniWordOffSet。

### 1.6 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”，记做 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]。

### 1.7 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”，记做 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]。

### 1.8 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”，记做 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 多模式合一的实例

为了对多模式合一有一个直观的印象，下面给出一个实例，以直观地说明多模式合一的具体计算过程。

目标模式：

张三是李四的$_7_$

事实库：

==========

F1 张三和李四是夫妻

F2 李四是男的

==========

规则库(原始输入形态)：

==========

R1 X 和 Y 是夫妻 → Y 和 X 是夫妻

R2 X 和 Y 是夫妻，X 是男的 → X 是 Y 的丈夫

R3 X 是 Y 的丈夫 → Y 是 X 的妻子

==========

规则库原文(存储形态，使用内部变量)：

==========

R1 $_1_$和$_2_$是夫妻 → $_2_$和$_1_$是夫妻

R2 $_3_$和$_4_$是夫妻，$_3_$是男的 → $_3_$是$_4_$的丈夫

R3 $_5_$是$_6_$的丈夫 → $_6_$是$_5_$的妻子

==========

带变量的全文索引：

张：F1.0

三：F1.1

和：F1.2 R1.1

李：F1.3 R2.0

四：F1.4 F2.1

是：F1.5 R3.1 R2.1 R1.3 F2.2

夫：F1.6 R2.5 R1.4

妻：F1.7 R3.4 R1.5

男：F2.3

的：F2.4 R3.3 R2.3

$_2_$：R1.0

$_1_$：R1.2

$_3_$：R2.0

$_4_$：R2.2

丈：R2.4

$_6_$：R3.0

$_5_$：R3.2

子：R3.5

多模式合一的具体计算过程：

| 目标模式中字符 | 候选常量字符 | 偏移量 | 候选变量字符 | 对应偏移量 |
|---|---|---|---|---|
| 张 | 张 | F1.0 | $_2_$/$_3_$/$_6_$ | R1.0/R2.0/R3.0 |
| 三 | 三 | F1.1 | $_2_$/$_3_$/$_6_$ | R1.0/R2.0/R3.0 |
| 是 | 是 | R2.1/R3.1 | | |
| 李 | 李 | R2.2/R3.2 | | |
| 四 | 四 | R2.2/R3.2 | | |
| 的 | 的 | R2.3/R3.3 | | |
| $_7_$ | | R2.4/R2.5/R3.4/R3.5 | | |
| $_7_$ | | R2.5/R3.5 | | |

==========

经过上述计算，可以得知，在事实库的所有事实、规则库的所有规则中，只有规则 R2 的右部“$_3_$是$_4_$的丈夫”、规则 R3 的右部“$_3_$是$_4_$的丈夫”，与目标模式“张三是李四的$_7_$”相匹配。

也就是说，经过多模式合一的运算，目标模式“张三是李四的$_7_$”，与事实库、规则库联立合一成功，锁定规则 R2、规则 R3。

## 3 多模式合一过程分析

从上例可以看出，多模式合一的过程可以看作是目标模式跟[一堆规则的右部＋一堆事实]进行比对，然后将与其合一的规则右部、事实挑选出来的过程。

在具体实现中，多模式合一是目标模式在一个图(“常量、变量一体化索引”)中进行检索的过程。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。多模式合一的计算过程是字级别的计算过程，粒度很细。

目标模式在混合索引中进行“检索/合一”，最后得到的规则右部、事实可能不止一个。这若干个规则右部、事实之间的关系是“或”的关系。也就是说，所有这些规则右部、事实，都

能够与目标模式合一成功。

在计算过程中，首先要得到当前字的偏移量集合 CurWordOffSet，然后与前一个字的偏移量集合 PreWordOffSet 进行求交集 CrossWordOffSet 的运算。其中，集合 PreWordOffSet 初始化为 IniWordOffSet。

如果交集 CrossWordOffSet 为空集，那么多模式合一失败。错误返回。否则，如果 CrossWordoffset 不是空集，并且那么对集合 PreWordOffSet 进行更新，得到新的集合 PreWordOffSet。

如果当前字是目标模式的最后一个字，并且 CrossWordOffSet 不是空集，CrossWordOffSet 中的某些元素在 EndWordOffSet 中出现，那么本次多模式合一运算结束，目标模式与事实库、规则库联立合一成功。否则，联立合一失败。

## 4 多模式合一的“图检索”算法

在上述工作的基础上，提出多模式合一的“图检索”算法。

### 4.1 算法的输入

①目标模式 sPattern。

②事实库（string sFactBase）、规则库（string sRuleBase）的名称。

### 4.2 算法的输出

①与目标模式合一的所有事实、每条事实在事实库中的位置，以及所有这些事实的总数目。

②与目标模式合一的所有规则右部、每条规则在规则库中的位置、每条规则与目标模式的合一结果，以及所有这些规则的总数目。

### 4.3 具体算法

①初始化；

将“起始字的偏移量集合”记做 IniWordOffSet。

将“终结字的偏移量集合”记做 EndWordOffSet。

将“变量字的偏移量集合”记做 VarWordOffSet。

将“先前字的偏移量集合”记做 PreWordOffSet。

将“当前字的偏移量集合”记做 CurWordOffSet。

将“变量安的最小偏移量集合”记做 Varwodminoffset。

将“字的最小偏移量集合”记做 WordMinOffSet。

将“常量、变量一体化全文索引”记做递增排列的字索引数组：struct。

stWord aWordIndex[MAX_WORD_NUM]。将其中字的个数记做 iWordNum。

根据“事实库”、“规则库”得到 IniWordOffSet、EndWordOffSet、VarWordOffSet、aWordIndex []、iWordNum。

赋值，令 PreWordOffSet = IniWordOffSet。

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，得到 CurWordOffSet；

④得到 VarWordMinOffSet，然后归并到 CurWordOffSet 之中。

合并方法如下：对于集合 VarWordMinoffSet 中的每一个偏移量 pVarWordMinoffset，如果集合 CurWordoffSet 中存在偏移量 pCurWordOffset，这两个偏移量的“sType”值、“iLineNum”值相同，并且 pVarWordMinOffset 的“iOffset”值大于 pCurWordOffset 的“iOffset”值，那么将偏移量 pVarWordMinOffset 插入到集合 pCurWordOffset 中。

对于集合 VarWordMinOffSet 中的每一个偏移量 pVarWordMinOffset，如果集合 CurWordOffSet 中不存在偏移量 pCurWordOffset，使得这两个偏移量的“sType”值、“iLineNum”值全都相同，那么将偏移量 pVarWordMinOffset 插入到集合 pCurWordOffset 中；

⑤计算 PreWordOffSet 与 CurWordOffSet 的交集：CrossWordOffSet；

⑥如果 CrossWordOffSet 是空集，那么多模式合一失败，返回 false；

⑦根据交集 CrossWordOffSet，对 VarWordOffSet 中的偏移量进行删减，从而得到新的 VarWordOffSet。

方法如下：对于 CrossWordOffSet 中的每一个偏移量 pCrossWordoffset，以及集合 VarWordOffset 中的每一个偏移量 PvarWordOffset，如果这两个偏移量的“sType”值均为“RULE_TYPE”，并且“iLineNum”值相同，并用 pVarWordOffset 的“iOffset”值大于等于 pCrossWordOffset 的“iOffset”值，那么集合 VarWordOffSet 保留偏移量 pVarWordOffset。否则，集合 VarWordOffSet 删除偏移量 pVarWordOffset。

⑧如果当前字 sCurWord 不是目标模式 sPattern 的最后一个字，那么根据交集 CrossWordOffSet，得到新的 preWordOffSet；

方法如下：preWordOffSet 置空。对于 CrossWordOffSet 中的每一个偏移量 pCrossWordOffset，将 pCrossWordOffset 的“iOffset”值加 1，然后插入到集合 PreWordOffSet 中。

如果偏移量 pCrossWordOffset 在 VarWordMinOffSet 中出现，那么将偏移量 pCrossWordOffset 插入到集合 PreWordOffSet 中。

⑨如果当前字 sCurWord 是变量，那么根据 PreWordOffSet 和 EndWordOffSet，得到新的 PreWordOffSet、CrossWordOffSet；

方法如下：根据 PreWordOffSet，得到“字的最小偏移量集合”WordMinOffSet。然后，将 WordminOffSet 与 EndWordOffSet 之间的所有偏移量构成一个集合，然后赋值给 PreWordOffSet。令 CrossWordOffSet=PreWordOffSet。

⑩如果当前字 sCurWord 是目标模式 sPattern 的最后一个字，那么计算 CrossWordOffSet 与 EndWordOffSet 的交集，并记做 UnifyEndWordOffSet；

如果 UnifyEndWordOffSet 为空集，那么本次多模式合一失败，返回 false。

否则，得到本次多模式合一的结果，返回 true。

⑪转②。

### 4.4 算法测试

#### 4.4.1 测试一：目标模式与事实库的匹配

事实库 FactBase：

F1：司马懿是司马昭的父亲

F2：司马昭是司马炎的父亲

测试用例 1：

sPattern=“$_1_$是司马昭的父亲”

测试结果：

锁定事实 F1：司马懿是司马昭的父亲。

变量$_1_$代换：“司马懿。”返回 true 值。

测试用例 2：

（下转第 223 页）

统需要解决的问题，是实现信息共享和网络协同的前提。基于这样的目的，本文从模型实现的功能和计算效率出发，对草图信息进行分层管理。在此次基础上，从信息表示和信息传输两个维度，建立了草图信息表示模型。文章还从静态和动态两方面，分别对草图信息的建模过程、草图信息到 XML DTD 的转换规则以及信息表示的实现策略、信息传输技术进行了详细叙述。通过实验证明，草图信息表示模型能够满足草图信息的一致表示、统一管理和信息传输要求。由于模型基于域管理的思想，可以通过域派生机制实现功能扩充，从而完成更复杂的草图信息管理。

今后工作中，将对草图系统在网络环境下共享与协同方面进行研究，同时完善和扩充草图信息表示模型，最终使草图系统的计算模型和表示模型形成完整的统一，以进一步扩展手绘草图的功能和应用领域。

## 参 考 文 献


1 周若鸿，孙正兴，张莉莎，等. 草图理解研究进展[J]. 计算机科学，2004，31(4)：140～146

2 Slate Corporation, Scottsdale A Z. JOT - A Specification for an Ink Storage and Interchange Format[J/OL]. http://hwr.nici.kun.nl/unipen/jot.html

3 栗阳，关志伟，戴国忠. 笔式用户界面开发工具研究[J]. 软件学报，2003，14(03)：392～400

4 Li Y, Guan ZW, Chen YD, Dai Gz. Penbuilder: platform for the development of PUI(pen-based user interface). In: Tan T, Shi Y, Gao W, eds. Proceedings of the 3rd International Conference on Multimodal User Interfaces(ICMI 2000)[J]. Springer-Verlag, 2000. 534～541

5 Chee Yi-Min, Magaña J, et al. Ink Markup Language -W3C Working Draft 28 September 2004[S]. http://www.w3.org/TR/InkML

6 徐晓刚. 草图理解系统及其关键技术研究：[硕士学位论文]. 南京：南京大学计算机科学与技术系，2003

7 W3C. Extensible Markup Language(XML) 1.0(Third Edition) W3C Recommendation 04 February 2004[S]. http://www.w3.org/TR/2004/REC-xml-20040204


(上接第 173 页)

sPattern="司马懿是$_1_$的父亲"

测试结果：

锁定事实 F1：司马懿是司马昭的父亲。

变量$_1_$代换："司马昭"。返回 true 值。

测试用例 3：

sPattetn="司马昭是司马炎的$_1_$"

测试结果：

锁定事实 F2：司马昭是司马炎的父亲

变量$_1_$代换："父亲"。返回 true 值。

测试结果分析：

测试结果表明，本算法可以有效地解决目标模式与事实库的匹配问题。并且对于目标模式中的变量，可以得到对应的代换它的常量字符串。

4.4.2 测试二：目标模式与规则库的匹配

规则库 RuleBase：

R1：$_1_$是$_2_$的父亲，$_2_$是$_3_$的父亲 → $_1_$是$_3_$的祖父

R2：$_1_$是$_2_$的母亲，$_2_$是$_3_$的父亲 → $_1_$是$_3_$的祖母

测试用例 1：

sPattetn="$_1_$是司马炎的祖父"

测试结果：

锁定规则 R1："$_1_$是$_2_$的父亲，$_2_$是$_3_$的父亲→$_1_$是$_3_$的祖父"

变量$_1_$代换："$_1_$"，返回 true 值。

测试用例 2：

spattern="司马懿是$_1_$的祖父"

测试结果：

锁定规则 R1："$_1_$是$_2_$的父亲，$_2_$是$_3_$的父亲→$_1_$是$_3_$的祖父"

变量$_1_$代换："$_1_$"。返回 true 值。

测试用例 3：

spattern="司马懿是司马炎的$_1_$"

测试结果：

锁定规则 R1："$_1_$是$_2_$的父亲，$_2_$是$_3_$的父亲→$_1_$是$_3_$的祖父"。

锁定规则 R2："$_1_$是$_2_$的母亲，$_2_$是$_3_$的父亲 → $_1_$是$_3_$的祖母"。

变量$_1_$代换："祖父"、"祖母"。返回 true 值。

测试结果分析：

实验结果表明，本算法可以有效地解决目标模式与规则库的匹配问题，可以有效地锁定匹配成功的规则。

## 5 "图检索"算法在问答系统中的应用

上述算法可以递归进行。在问答系统中的具体应用如下：

①如果目标模式匹配事实库、规则库均失败，那么本次模式推理失败，返回 false；

②如果匹配事实库成功，那么本次模式推理成功，返回 true；

③如果匹配事实库失败，但是匹配规则库成功，那么通过代换规则库中的变量，得到新的目标模式，继续进行模式推理。

将用户查询进行处理，作为目标模式。通过上述算法，可以得到用户查询的答案。

总之，本算法可以应用于问答系统，是提高问答系统准确率的一条有效途径。所以，对于问答系统来说，多模式合一有很高的应用价值。

**结论和下一步的工作** 本文首先介绍了多模式合一的相关定义、研究现状，然后给出了多模式合一的模型和机制。在此基础上，重点给出了多模式合一的算法——"图检索"算法。实验结果表明，本算法可以有效地解决多模式合一问题。并且，本算法可以直接应用到问答系统中，较大地提高问答系统的准确率(precision)。下一步工作，将致力于减小算法的时间复杂性，进一步提高算法的效率。

## 参 考 文 献


1 Turing A M. Computing Machinery and Intelligence. Mind, 1950, 59(236): 433～460

2 Searle J R. Minds, brains, and programs. Behavioral and Brain Sciences, 1980, 3: 417～424

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

4 Simmons R F. Natural Language Question-Answering Systems. Communications of the ACM, 1969, 13(1): 15～30

5 白硕. 语言计算与基于内容的文本处理. 见：大规模内容计算. 北京：清华大学出版社，2003. 13～25