

基于条件冲突分析的主动规则集汇流性判定方法

熊中敏 朱吉光 李长绩 袁红春
(上海海洋大学信息学院 上海 201306)

摘要 主动规则使数据库具备自反应能力,在数据库、知识库和无线传感器网络等领域得到了广泛应用。汇流性分析是保持数据一致性的重要手段,但有效判定汇流性仍是一个难题。现有的方法基于规则可交换性进行判定,没有考虑规则之间是否存在条件冲突而导致它们不可能同时出现在同一个执行序列中;同时基于触发关系的集合分析没有考虑在规则触发序列中的先后时序性。为此,提出了规则的触发序列概念,分析了单个触发序列上的条件冲突和一对无序规则所在的两条触发序列之间的条件冲突,从而提出了更有效的汇流性先决条件及其判定定理,同时给出了新的判定算法的完整描述及其可终止性、正确性证明和复杂度分析。

关键词 主动数据库,规则分析,汇流性,条件冲突,数据一致性

中图法分类号 TP311.131 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.5.031

Confluence Decision Method for Active Rule Set Based on Condition Conflict Analysis

XIONG Zhong-min ZHU Ji-guang LI Chang-ji YUAN Hong-chun
(School of Information, Shanghai Ocean University, Shanghai 201306, China)

Abstract With active rules, many databases have automatic reaction function. Many areas including databases, knowledge database and wireless sensor network have employed active rules. Confluence analysis can maintain data consistency of database, but it is a very hard task. Based on rule commutativity, the existing methods do not consider whether there exists a condition conflict between rules and then these rules cannot be executed at the same executive sequence. And the existing analysis is based on the rules set and does not consider the time order between triggering rules. To address these problems, the concepts of triggering sequence was provided and the condition conflicts in a triggering sequence were analyzed. Finally, a new algorithm of confluence detection was provided and its correctness and termination were proved.

Keywords Active database, Rule analysis, Confluence, Condition conflict, Data consistency

1 引言

主动规则在传统商业数据库中得到了成熟的应用,提高了数据库的自反应能力,比如应用于数据库更新一致性约束;其近年来还应用到许多新领域中:例如 XML 文档^[1,2]、RDF^[3]、语义网络^[4]和传感器数据库^[5]等。当系统中触发事件产生,就检测条件是否满足,一旦满足就执行规则动作。如果有两个并发触发的规则且没有规定先后执行的优先级别,就可以随机地选取其中一个先触发执行,这样就会导致两个不同的执行序列。如果一个规则集具有汇流性,则规则集的执行导致的最后状态只能有一个,与调度执行序列无关。

主动规则集的汇流性维护了数据的一致性,从而保证了数据质量。文献[6-12]基于规则的可交换性进行判别,具有一定的保守性:只有当任意一对无序规则满足可交换性,才能准确判定规则集具有汇流性;否则,它们无法判定本文第3节例1所示的规则集的汇流性。文献[6]提出具有优先权限的

规则可交换性概念;文献[9]提出将主动规则转换为演绎规则并结合规则被触发的优先级进行判定;文献[10]提出针对 RDF 规则的可交换性识别方法;文献[11]利用模糊佩里网模型推导出规则集上触发优先次序的一个全序并进行判定;文献[12]提出了无优先级且含排他性选择规则的规则集的汇流性判定方法。上述方法存在以下不足:

1)文献[6-11]没有考虑规则之间是否存在执行条件冲突,从而导致它们不可能同时出现在同一个执行序列中。文献[12]部分地考虑了这一问题,但相较于软件程序设计中选择分支的出现,显然排他性选择规则无法包含以下情形:在一个触发序列(即一个执行选择分支)中出现执行条件冲突的规则。现有方法无法支持在上述情形下判定汇流性。

2)文献[6-12]在分析中都是基于触发关系形成的规则集合进行分析,没有考虑规则先后触发形成的触发序列的时序性,导致分析效率较低,详细分析可见下文论述。

到稿日期:2015-04-23 返修日期:2015-08-30 本文受上海市科委科技支撑项目(14391901400),国家自然科学基金(61272098),国家 973 重大基础研究基金(2012CB316200),2015 年度上海海洋大学大学生创新计划项目资助。

熊中敏(1971-),男,博士,副教授,CCF 会员,主要研究方向为数据库、数据挖掘,E-mail: zmxiong@shou.edu.cn;朱吉光(1993-),男,硕士生,主要研究方向为数据库;李长绩(1994-),男,主要研究方向为数据挖掘;袁红春(1971-),男,博士,教授,主要研究方向为知识库与专家系统。

2 相关知识

2.1 主动规则

按图 1 所示流程处理规则^[13]。

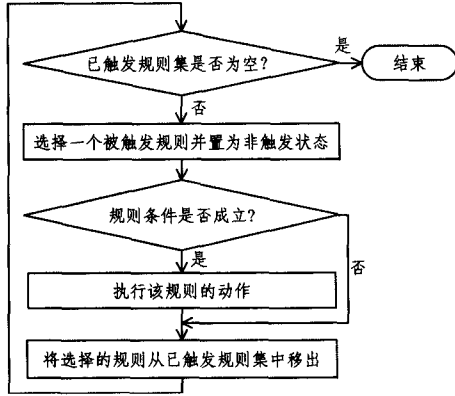


图 1 规则执行处理过程

主动规则遵循如下定义形式: On Event if Condition then Action。选取关系数据库模式, 可作以下解释^[13]: 事件(Event)集合是一组被监视的数据操纵操作; 条件(Condition)是一个在当前数据库状态和规则的过渡值上表示的谓词语句; 动作(Action)是一组数据操纵操作。在以上流程中, 规则动作的执行可能产生新的事件, 会进一步触发其它的规则而添加到已触发规则集中。若集合非空, 则以上过程被循环执行; 一旦为空, 规则处理过程终止。

在已触发规则集中, 可能多个规则被触发, 形成一个冲突规则集, 若一对规则未分配优先级, 则可随机选择一个规则先执行, 此时称为非确定性规则处理过程^[13], 即可出现不同的执行序列, 选取不同的执行序列执行将产生前述的汇流性问题。

2.2 汇流性判定定理

非确定性处理过程中, 一个规则集可随机调度两个不同执行序列中的一个作为实际被执行的序列, 若从这两个序列中分别抽取任意一个规则形成一对规则, 它们的执行次序具有可交换性, 则实际运行任选其中一个序列执行不会影响数据库具有唯一的最后状态, 也就是规则集具有汇流性。如果在执行图的某一状态 S 时, 先执行规则 r_i 再执行规则 r_j 和先执行规则 r_j 再执行规则 r_i 产生相同的执行图状态 S' (如第 3 节图 2 所示), 则称这两个规则是可交换的; 否则称规则 r_i 和 r_j 是不可交换的。

文献[6]基于可交换规则的定义, 提出了规则集具备汇流性的先决条件。为了方便描述, 引入以下记号^[6-8]: 1) 令 $R = \{r_1, r_2, \dots, r_n\}$ 代表一个被分析的规则集; 2) 用 $T = \{t_1, t_2, \dots, t_m\}$ 代表数据库中的表, $C = \{t_i.c_j, t_k.c_l, \dots\}$ 代表表的列; 3) 用 O 代表修改数据操作: $O = \{\langle I, t \rangle | t \in T\} \cup \{\langle D, t \rangle | t \in T\} \cup \{\langle U, t, c \rangle | t.c \in C\}$, 其中 $\langle I, t \rangle$ 代表插入数据, $\langle D, t \rangle$ 代表删除数据, $\langle U, t, c \rangle$ 代表对表的列进行数据更新。

进一步地, 引入如下的集合定义和标识符号^[6-8]: 1) $Triggered-By(r)$ 表示引起规则 r 被触发的包含在 O 中的修改数据操作; 2) $Performs(r)$ 表示规则 r 的执行动作; 3) $Triggers(r)$ 表示被规则 r 触发的所有规则 r' (可能包含 r), $Triggers(r) = \{r' \in R | Performs(r) \cap Triggered-By(r') \neq \emptyset\}$; 4) $Uses(r)$ 为在规则 r 执行时可能用到的数据表中的数据列;

5) $Can-Untrigger(O')$ 表示执行 O' ($O' \in O$) 的操作后不能被当前数据库状态满足执行条件的被触发的规则集, $Can-Untrigger(O') = \{r \in R | \langle D, t \rangle \in O', \langle I, t \rangle$ 或 $\langle U, t, c \rangle \in Triggered-By(r), t \in T, c \in C\}$ 。

引理 1^[6-8] r_i 和 r_j 表示两个不同规则, 若如下任一条件都不成立, 则它们是可交换规则, 否则它们可能不具备交换性: 1) $r_j \in Triggers(r_i)$, 即 r_i 的执行可导致触发 r_j ; 2) $r_j \in Can-Untrigger(Performs(r_i))$, 即 r_i 的执行导致 r_j 因条件不满足不被触发; 3) r_i 的执行动作导致 r_j 执行时用到的值发生变化; 4) $\langle I, t \rangle$ 在 $Performs(r_i)$ 中, $\langle D, t \rangle$ 或 $\langle U, t, c \rangle$ 在 $Performs(r_j)$ 中 ($t \in T$ 或 $t.c \in C$), 即 r_i 的插入数据动作能改变 r_j 更新或删除数据后的数据库状态; 5) $\langle U, t, c \rangle$ 同时在 $Performs(r_i)$ 和 $Performs(r_j)$ 中, 即 r_i 的更新数据能改变 r_j 数据更新的结果; 6) 将 1)–5) 中 r_i 和 r_j 互换。

引理 2^[12] 符合以下条件的规则互为无序规则:

1) 触发图中所有无入边的规则, 即这些规则不被任何其他规则的动作所触发。

2) 触发图中入边来自同一个规则的规则, 即这些规则为同一个规则的动作所触发。

3) 若规则 r_1 和 r_2 是无序的, r_i 和 r_j 分别被 r_1 和 r_2 触发, 分别来自于 $\{r_1, r_i\}$ 和 $\{r_2, r_j\}$ 的两个规则。

如文献[12]一样, 为了分析的简便, 下面定义中的无序规则对是指按引理 2 中前两个条件识别, 然后按触发关系推导, 得到全部无序规则对。

定义 1^[12] (无优先级汇流先决条件) 规则集 R 中规则未定义优先级, 其中任意规则对 r_i 和 r_j 是无序的, 按如下方式建立 $R_1 \subseteq R$ 和 $R_2 \subseteq R$: $R_1 \leftarrow \{r_i\}$, $R_2 \leftarrow \{r_j\}$ 。

重复执行以下步骤 1)、2), 直到 R_1 和 R_2 保持不变为止:

1) $R_1 \leftarrow R_1 \cup \{r \in Triggers(r_1), r_1 \in R_1\}$;

2) $R_2 \leftarrow R_2 \cup \{r \in Triggers(r_2), r_2 \in R_2\}$ 。

对每一对 $r_1 \in R_1$ 和 $r_2 \in R_2$, r_1 和 r_2 可交换。

引理 3^[12] (无优先级的汇流定理) 如果规则集 R 未定义优先级且具备汇流先决条件, 并且 R 的任何执行图只存在有限长度的路径, 则 R 的所有执行图最终状态都相同, 即 R 具备汇流性。

由引理 3 可知, 保证汇流性要求规则集执行是可终止的, 如何判定执行的可终止性是一个难点^[14-17], 由于这不是本文的研究重点, 因此本文不作阐述并设定以下所指规则集都是可终止的。

3 研究实例

文献[6-11]对有优先级分配的规则集进行分析; 文献[12]扩展至无优先级规则集分析。以下实例中的规则集为无优先级规则集, 用来说明本文的研究动机。

例 1 说明在一个规则触发序列中出现执行条件冲突的规则对规则集汇流性的影响。规则条件或动作访问事件参数的方式^[13]为: old 指一个数据项在事件更新其值前所具有的数据值; insert 指最近插入的值; delete 指最近删除的值; update 指为一个更新(update)事件所影响的数据项的属性。

例 1 在一个数据库模式中包含如下两个表:

Account (num, name, balance, rate, repute) Customer (num, name, address, city)

关系 Account 包含了一个银行的帐户信息;关系 Customer 包含了银行的客户的信息;斜体的属性为关系的关键字。

1)当账户余额发生改变,若余额小于 20000 元且信用值不小于 0,新增余额大于 500 元,则将其用户等级增加 1。规则 r_1 :

```
On update(Account, balance)
If(updated, balance<20000) & (repute>=0) & ((updated,
balance-old, balance)> 500)
```

```
Then update Account set rate=rate+1
```

```
Where num=updated, num;
```

2)当账户余额发生改变,若余额大于 20000 元且新增余额大于 1000 元,将其用户等级增加 2。规则 r_2 :

```
On update(Account, balance)
If(updated, balance>20000) & ((updated, balance-old,
balance)>1000)
```

```
Then update Account set rate=rate+2
```

```
Where num=updated, num;
```

3)当用户等级发生改变,若信用值小于 0、用户等级大于 0、账户余额小于 10000 元且标识变量 $reducerate=1$ (防止多次循环触发执行设置的临时变量),则将其用户等级减 1。规则 r_3 :

```
On update(Account, rate)
If(repute<0) & (updated, rate>0) & (updated, balance
<10000) & reducerate=1
```

```
Then update Account set rate=rate-1, reducerate =0
```

```
Where num=updated, num;
```

4)当账户余额发生改变,若余额大于 10000 元、信用值小于 0、用户等级小于 0 且新增余额大于 1000 元,发出提示信息。规则 r_4 :

```
On update(Account, balance)
If(updated, balance>10000) & (repute<0) & (rate<0)
& ((updated, balance-old, balance)>1000)
```

```
Then setmessage("请改善你的信誉或增加你的余额,以
便升级你的用户等级")
```

```
Where num=updated, num;
```

与例 1 相对应的触发图如图 2 所示。

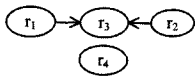


图 2 与例 1 相对应的触发图

在图 2 中,如文献[12]所述,规则 r_1 、 r_2 和 r_4 任意两个形成一对无序规则,其中 r_1 和 r_2 、 r_1 和 r_4 因为存在条件冲突互为排他性选择规则。显然文献[6-11]认为规则 r_1 和 r_2 是不可交换规则(引理 1 的第 5 条),该规则集被判定不具备汇流性;由于 r_2 和 r_4 不是排他性选择规则,文献[12]要求 r_4 和 $\{r_2, r_3\}$ 中任一规则互为可交换规则,显然根据引理 1 的第 3)条 r_4 和 r_3 是不可交换规则($rate$ 分别出现在两个规则的动作和条件中),因此文献[12]也制定该规则集不具备汇流性。实际上 r_2 和 r_3 存在条件冲突(条件中含 $repute$ 的谓词), r_3 虽然可以被 r_2 触发但不能在同一执行序列 $r_2 \rightarrow r_3$ 中被执行。所以只要求 r_2 和 r_4 是可交换的,显然条件成立,因此该规则集实际具备汇流性。

4 基于触发序列的条件冲突分析

在规则处理过程中,如图 1 所示,规则的执行首先需要触发事件发生,然后才能有资格进行条件评估,所以实际上规则集的执行由先后触发次序形成了一个触发序列。由汇流判定引理 3 可知规则处理过程的可终止性是保证汇流性的前提,故本文所指触发序列都是有限次执行的序列。

定义 2 在触发图上由一个起始规则 r 为起点的一条有限次执行路径称为以 r 为起点的触发序列。

显然文献[6-12]提出的汇流先决条件只是以集合形式表示一个起始规则所触发的规则集,不能鲜明地表示出规则执行时的先后触发次序。

定义 3^[12] 谓词 q 为规则集 R 的规则条件中出现的谓词且不被 R 中任意规则的动作所更新,则称 q 为规则集 R 的不可更新谓词。

所谓条件公式就是从两个或两个以上规则的条件中选择相应的谓词,这些谓词形成一个具有 false 或 true 真值的逻辑表达式。文献[18,19]提出了为一个执行序列建立条件公式的方法,并且提出了选择谓词的过程,我们可以引用来建立两个规则之间的条件公式。不可更新谓词的状态在执行过程中不发生变化,避免了形成的条件公式的真值因执行过程中谓词状态发生变化而导致的不确定性。

定义 4 对于规则集 R 中任意一对规则 r_i 和 r_j ,如果基于规则集 R 的不可更新谓词分别建立条件公式 $formula(r_i)$ 和 $formula(r_j)$ 且 $formula(r_i) \cap formula(r_j) = false$,则 r_i 和 r_j 为执行条件冲突规则。

由例 1 的相关分析可知,执行条件冲突规则比文献[12]定义的排他性选择规则更具普遍性。

定理 1 若 r_i 和 r_j 表示规则集 R 中一对执行条件冲突规则,则 R 的任意一条执行序列如果包含 r_i 则必不包含 r_j ,反之亦然。

证明:通过反证法证明。假设存在 R 的一个执行序列 Sq 同时包含 r_i 和 r_j ,即此时 r_i 和 r_j 的条件都可以满足并被调度执行,亦即基于 R 的不可更新谓词建立的条件公式有 $formula(r_i) \cap formula(r_j) = true$,这与执行条件冲突规则满足 $formula(r_i) \cap formula(r_j) = false$ 矛盾,即假设不成立,故结论成立。证毕。

由例 1 的相关分析可知:文献[6-12]没有考虑触发序列上规则的条件冲突导致无法精确判定规则集中无序规则对不具备可交换性时的汇流性。为此,本文提出定理 2。

定理 2 若 r_i 和 r_j 表示规则集 R 中一对执行条件冲突规则且 r_i 触发 r_j (因为设定触发图不存在无穷路径,故 r_j 不可触发 r_i),此时如果触发图上某一路径既包含 $r_i \rightarrow r_j$ 又包含 $r_j \rightarrow r_k$ (即 r_j 触发 r_k),则该路径对应的执行序列必不包含 r_i 和 r_k 。

证明:通过反证法证明。假设存在该路径的一个执行序列 Sq 同时包含 r_i 、 r_j 和 r_k 。显然按触发次序,该路径包含子路径 $r_i \rightarrow r_j \rightarrow r_k$ 。已知 r_i 和 r_j 是执行条件冲突规则,按定理 1 可知 Sq 若包含 r_i 则必不包含 r_j ;若 Sq 不包含 r_i ,在此执行序列中 r_j 不被 r_i 触发,即 r_j 也不被包含。故 Sq 必不包含 r_j 。则 Sq 中由于不包含 r_j , r_k 不能被触发,即 r_k 也不能被 Sq 包含。假设不成立,故结论成立。证毕。

按定理 2,可寻找到一个触发序列中第一个与排序在其前面的规则发生执行条件冲突的规则,则该规则及排序在其后的规则都不包含在该触发序列中。由此,本文提出一种裁剪触发序列的操作,使其达到精简,从而更准确、有效地判定规则集的汇流性。

定义 5 设 δ 是一条以 r_i 为起始规则的触发序列,则定义操作 $Curtail(\delta, r_i)$ 如下:以 r_i 为起点,在 δ 上寻找到第一个规则 r_k ,使 r_k 与子序列 $r_i \rightarrow \dots \rightarrow r_{k-1}$ 上某一规则互为执行条件冲突规则,则将规则 r_k 及排序在其后的规则裁剪掉,即 δ 裁剪为子序列 $r_i \rightarrow \dots \rightarrow r_{k-1}$,并称经过此操作后的触发序列为精简触发序列。

5 基于条件冲突规则的汇流性判定

在定理 2 的基础上,提出如下含条件冲突规则的汇流性先决条件。

定义 6(含条件冲突规则的汇流先决条件) 规则集 R 中规则未定义优先级,其中任意规则对 r_i 和 r_j 是无序的,设在 R 对应的触发图 G 中分别以 r_i 和 r_j 为起始规则的任意两条触发序列(即图中触发路径)为 δ_1 和 δ_2 ,在图 G 中执行 $Curtail(\delta_1, r_i)$ 和 $Curtail(\delta_2, r_j)$,记经过裁剪操作后的精简触发图为 G' 。 r_i 和 r_j 满足以下条件之一:

(1)它们是执行条件冲突规则;

(2)否则,在精简图 G' 上,按如下方式建立 $R_1 \subseteq R$ 和 $R_2 \subseteq R$: $R_1 \leftarrow \{r_i\}$, $R_2 \leftarrow \{r_j\}$ 。

重复执行以下步骤 1)、2),直到 R_1 和 R_2 保持不变为止:

1) $R_1 \leftarrow R_1 \cup \{r \in Triggers(r_1), r_1 \in R_1\}$;

2) $R_2 \leftarrow R_2 \cup \{r \in Triggers(r_2), r_2 \in R_2\}$ 。

对每一对 $r_1 \in R_1$ 和 $r_2 \in R_2$, r_1 和 r_2 是可交换的。

定理 3(含条件冲突规则的汇流定理) 如果规则集 R 未定义优先级且具备含条件冲突规则的汇流先决条件,并且 R 的任何执行图只存在有限长度的路径,则 R 的所有执行图最终状态都相同,即 R 具备汇流性。

证明:通过反证法证明。假设 R 中的规则不具有汇流性,则必存在不等价的两条执行序列 S_{q_1} 和 S_{q_2} ,也即无法通过规则的可交换性使将 S_{q_1} 和 S_{q_2} 中的规则达到一致。不妨假定根据规则的可交换性 S_{q_1} 向 S_{q_2} 的形式调整,必然存在有无法交换次序的两个规则,导致 S_{q_1} 和 S_{q_2} 出现执行序列的不同。从规则处理过程看,正是由于从一对无序规则形成的冲突规则集选取规则执行的次序不同导致了不同的执行序列,因此这两个规则必包含在两个无序规则为起点的两个不同的触发序列上。1)如果其中一个规则或二者皆为所在触发序列上的条件冲突规则,根据定义 5 的精简操作,其中一个规则或二者均可被裁剪掉,显然 S_{q_1} 和 S_{q_2} 形式变成一致,这与 S_{q_1} 和 S_{q_2} 不等价相矛盾。2)如果这两个规则不为条件冲突规则,则由上述先决条件可知它们是可交换的,这与它们在 S_{q_1} 和 S_{q_2} 中无法交换相矛盾。故 R 具备汇流性。证毕。

以下算法判定含条件冲突规则的汇流性。 R 表示任意优先级的可终止执行规则集,并给定与之相关的触发图 TG ; $Out(r_i)$ 表示在 TG 中所有属于以规则 r_i 为始点的有向边的终点; r_i . T 表示在 TG 中以 r_i 为终点的有向边个数; L 表示一个存储规则的表结构。

函数 1 Trigger_Set()

输入:规则 r

输出:由 r 为初始规则的触发序列对应的规则集合 Q

Begin

$L := \emptyset; Q := \emptyset; Q := Q \cup \{r\}; L := \text{append}(L, r)$

while not Empty(L) do

$\{r_i := \text{pop}(L);$

FOR 每个规则 $r_j \in Out(r_i)$ do

$\{Q := Q \cup \{r_j\}; L := \text{append}(L, r_j)\}$

Return(Q);

END

函数 2 Curtail(δ, r)

输入:触发路径 δ 和 δ 上起始规则 r

输出:裁剪后的路径

Begin

formula(δ)为基于规则集不可更新谓词建立的 δ 上规则的联合条件公式,且有 formula(δ) := $\emptyset$;

formula(δ) := formula(r); $r_i := r$;

FOR 每个规则 $r_j \in \delta$ 且为 r_i 的下一条触发规则 do

{If (formula(δ) $\cap$ formula(r) = true) then

{formula(δ) := formula(δ) $\cap$ formula(r); $r_i := r_j$;

else

{从图 G 中删除 δ 上 r_i 到 r_j 的触发入边,即将 r_j 及在 δ 上排序在其后的规则从 δ 上裁剪掉;

return;}

}

END

算法 1 Refined Confluence-test based on Condition conflict detection

输入:规则集 R 和它的触发图 G

输出:如果判断是汇流的,返回 true;否则,返回 false

Begin

$R_1 := \emptyset; R_2 := \emptyset; Q_1 := \emptyset; Q_2 := \emptyset; R_1 := \emptyset; R_2 := \emptyset;$

(1)FOR each rule $r \in R$ do /* 第一类无序:无人边 */

{IF r . $T=0$ then $Q_1 := Q_1 \cup \{r\}$;

IF $Q_1 \neq \emptyset$ then

{FOR each rule $r_1, r_2 \in Q_1$ do

{FOR 任意分别以 r_1, r_2 为起点的触发路径 δ_1 和 δ_2 do

$\{\delta_1 := Curtail(\delta_1, r_1); \delta_2 := Curtail(\delta_2, r_2)\}$

记精简后的图为 G' ;

IF NOT(r_1, r_2 为执行条件冲突规则) then

{在 G' 上执行: $R_1 := Trigger_Set(r_1); R_2 := Trigger_Set(r_2)$;

FOR $\forall r_i \in R_1, r_j \in R_2$ do

{IF NOT(r_i, r_j 根据引理 1 为可交换规则) then
return(false);}}}

(2)FOR each rule $r \in R$ do

{IF Numberof($Out(r)$) > 1 then $Q_2 := Q_2 \cup \{Out(r)\}$; /* 第二类无序:多出边 */

IF $Q_2 \neq \emptyset$ then

{FOR each rule $r_1, r_2 \in Q_2$ do

{FOR 任意分别以 r_1, r_2 为起点的触发路径 δ_1 和 δ_2 do

$\{\delta_1 := Curtail(\delta_1, r_1); \delta_2 := Curtail(\delta_2, r_2)\}$

记精简后的图为 G' ;

IF NOT(r_1, r_2 为执行条件冲突规则) then

{在 G' 上执行: $R_1 := Trigger_Set(r_1); R_2 := Trigger_Set(r_2)$;

For $\forall r_1 \in R_1, r_2 \in R_2$ do

{IF NOT(r_1, r_2 根据引理 1 为可交换规则) then
return(false); } }

$Q_2 := \emptyset$

(3) Return(true); /* 满足定理 3, 可汇流 */

END

定理 4 算法 1 是可终止的、正确的, 复杂度为 $O(m^4 + m^2 \times n^2)$, m 为规则集 R 的大小, n 表示评价算法中符号数量。

证明: 1) 正确性: 定理 3 为算法 1 的正确性的理论基础。
2) 可终止性: 因为 R 中规则数和由函数 Trigger-Set 计算出的 R_1 和 R_2 都是有限的规则; 规则集 Q_1 和 Q_2 小于 R 可知它们所含规则有限; 需要建立条件公式判定 r_1, r_2 是否为条件冲突规则, 因为 r_1, r_2 的条件中含有限个谓词, 所以条件公式个数也有限, 故有限次循环执行算法 1 中的 FOR 语句, 即算法 1 可终止。3) 分析: 显然算法复杂度由步骤 (1) 或 (2) 决定。步骤 (1) 和 (2) 的复杂度相同, 故只分析步骤 (1)。函数 Trigger_Set 在最坏时可包含 R 中全部规则, 同理, R_1, R_2, Q_1 和 Q_2 中不多于 m 个规则; 语句“FOR each rule $r_1, r_2 \in Q_1$ do”可执行 C_m^2 次; 用作精简序列的 FOR 语句可执行 $(C_m^1 \times C_m^1)$ 次; 执行“IF NOT(r_1, r_2 是执行条件冲突规则)”语句时, 可通过文献[20]提出的算法计算条件公式是否成立, 符号分析表在该算法对每对规则分析时最多扫描 $O(n^2)$ 次; 语句“FOR each rule $r_1 \in R_1, r_2 \in R_2$ do”执行 $(C_m^1 \times C_m^1)$ 次。步骤 (1) 最内层的 For 语句最多执行 $C_m^2 \times (C_m^1 \times C_m^1 + O(n^2) + C_m^1 \times C_m^1)$ 次。故算法的时间复杂度为 $O(m^4 + m^2 \times n^2)$ 。证毕。

结束语 当无优先级规则集含有条件冲突规则时, 现有汇流性判定方法不能精确判定其汇流性。为了进行有效判定, 条件冲突规则和含条件冲突规则的汇流先决条件的概念先后被本文提出, 由此提出的汇流性判定定理和新的识别算法更加有效。同时还给出了该算法的正确性、可终止性证明及其时间复杂度分析。

参考文献

- [1] Bonifati A, Ceri S, Paraboschi S. Active rules for XML: A new paradigm for e-services[J]. VLDB Journal, 2001, 10(1): 39-47
- [2] Mansour E, Dube K, Wu Bing. AIM: An XML-Based ECA Rule Language for Supporting a Framework for Managing Complex Information[C] // Proceedings of International Symposium. RuleML 2007. 2007; 232-241
- [3] Bae H, Yu Y W. Auto-Triggering of RFID-based Logistic Process in Inter-workflow Using Complex Event[M] // Dynamics in Logistics. Springer, 2011; 329-339
- [4] Behrends E, Fritzen O, May W, et al. Embedding Event Algebras and Process for ECA Rules for the Semantic Web[J]. Fundamenta Informaticae, 2008, 82(3): 237-263
- [5] Öztarık H, Akkaya K, Yazici A. Providing Automated Actions in Wireless Multimedia Sensor Networks via Active Rules[M] // Computer and Information Sciences II. Springer, 2012; 185-190
- [6] Aiken A, Hellerstein J, Widom J. Static Analysis Techniques for Predicting the Behavior of Database Production Rules[J]. ACM Transactions on Database Systems, 1995, 20(1): 3-41
- [7] Shu Jie. Analyzing Rules of Active Database based on Propagation Algorithm[D]. Harbin: Harbin University of Science and Technology, 2006 (in Chinese)
舒杰. 基于传播算法的主动数据库规则特性分析[D]. 哈尔滨: 哈尔滨理工大学, 2006
- [8] Yang Ze-xue. Termination Analysis of Active Rules[D]. Harbin: Harbin University of Science and Technology, 2006 (in Chinese)
杨泽雪. 主动规则的终止性分析[D]. 哈尔滨: 哈尔滨理工大学, 2006
- [9] Comai S, Tanca L. Termination and Confluence by Rule Prioritization[J]. IEEE Transactions on Knowledge and Data Engineering, 2003, 15(2): 257-270
- [10] Papamarkos G, Poulouvasilis A, Wood P T. Event-Condition-Action Rules on RDF Metadata in P2P Environments[J]. Computer Networks, Elsevier, 2006, 50(10): 1513-1532
- [11] Bostan-Korpeoglu B, Yazici A. A fuzzy Petri net model for intelligent databases[J]. Data & Knowledge Engineering, 2007, 62(2): 219-247
- [12] Xiong Zhong-min, Zhao Meng-lu, Zhou Xue-nan, et al. A Confluence Decision Method for an Active Rule Set with Exclusive Rules [J]. Journal of Computer Research and Development, 2013, 50(Suppl.): 109-116 (in Chinese)
熊中敏, 赵梦露, 周雪楠, 等. 含排他性选择规则的主动规则集的汇流性判定方法[J]. 计算机研究与发展, 2013, 50(增刊): 109-116
- [13] Baralis E, Ceri S, et al. Compile-Time and Runtime Analysis of Active Behaviors[J]. IEEE Transactions on Knowledge and Data Engineering, 1998, 10(3): 353-370
- [14] Xiong Zhong-min, Zhao Meng-lu, Huang Dong-mei, et al. Active rules termination analysis based on activation path and enhanced formula [J]. International Journal of Intelligent Information and Database Systems, 2013, 7(1): 53-78
- [15] Xiong Wei, Wu Ye, Zhang Zhen, et al. Termination Analysis for Active Rule Set Based on Triggering Path [J]. Chinese Journal of Computers, 2012, 35(1): 65-75 (in Chinese)
熊伟, 吴烨, 张震, 等. 基于触发路径的主动规则集终止性分析[J]. 计算机学报, 2012, 35(1): 65-75
- [16] Xiong Zhong-min, Hao Zhong-xiao. An Approach to Termination Decision for a Rule Set Based on Activation Path and Conditional Formula [J]. Journal of Computer Research and Development, 2006, 43(5): 901-907 (in Chinese)
熊中敏, 郝忠孝. 基于活化路径和条件公式的主动规则集可终止性判定方法[J]. 计算机研究与发展, 2006, 43(5): 901-907
- [17] Hao Zhong-xiao, Xiong Zhong-min. An Efficient Algorithm for Computing an Irreducible Rule Set in Active Database [J]. Journal of Computer Research and Development, 2006, 43(2): 281-287 (in Chinese)
郝忠孝, 熊中敏. 计算主动数据库中不可归约规则集的有效算法[J]. 计算机研究与发展, 2006, 43(2): 281-287
- [18] Lee S Y, Ling T W. Refined Termination Decision in Active Databases[C] // Proceedings of Int'l Conf. On Database and Expert Systems Applications (DEXA). Toulouse, France, 1997
- [19] Lee S Y, Ling T W. A path Removing Technique for Detecting Trigger Termination[C] // Proceedings of Int'l Conf. on Extended Database Technology (EDBT). Valencia, Spain, 1998
- [20] Urban S D, Tschudi M K, Dietrich S W, et al. Active Rule Termination Analysis: An Implementation and Evaluation of the Refined Triggering Graph Method[J]. Journal of Intelligent Information Systems, 1999, 12(1): 27-60