

基于触发路径的主动规则集终止性分析

熊 伟 吴 焯 张 震 吴秋云

(国防科学技术大学电子科学与工程学院 长沙 410073)

摘 要 为满足用户对网络服务的个性化、定制化和主动化需求,主动规则成为解决这些问题的关键技术,研究了在网络环境下基于规则的复杂应用中,大量规则集同时触发所带来的规则终止性问题,提出的分析方法确保主动规则能够有效运行,以提供更加灵活的主动服务.讨论了以静态分析方法为主的主动规则终止性分析相关工作,随后给出问题描述和相关形式化定义.分析了基于关联图的终止性分析方法的保守性,引入触发路径和有限触发环概念,提出了基于触发路径的两种终止情形分析方法,提高了规则集终止性分析的准确性,采用两阶段分析算法保证了分析效率.与相关分析方法的实验比较说明,文中方法能够更准确高效地检测主动规则集的终止性,并适应基于主动规则的其它应用.

关键词 主动规则;触发路径;有限触发环;终止性分析;关联图

中图法分类号 TP311 DOI号: 10.3724/SP.J.1016.2012.00065

Termination Analysis for Active Rule Set Based on Triggering Path

XIONG Wei WU Ye ZHANG Zhen WU Qiu-Yun

(College of Electronic Science and Engineering, National University of Defense Technology, Changsha 410073)

Abstract The demands for Web service are becoming more and more personalized, customized and active, active rule is one of the key techniques to meet this requirement. In those large scale and complicated Web applications with active rule, the termination problem of rule sets triggered simultaneously is important and attracted great interests of research. Therefore, analytical methods are provided for efficient execution of rule and flexibility of active service. Compile-Time analysis methods and problems are overviewed. Then, the problem descriptions and formal definitions are presented. For the conservation of previous techniques based on relationship graph, triggering path method is proposed to improve the accuracy of termination analysis for active rule set. The main contribution of this method is the definition of finite trigger ring. Two new cases of termination in the triggering path are used to improve the accuracy of traditional methods. Because of the higher complexity of triggering path algorithm, a two-stage analysis algorithm is proposed to ensure efficiency of the analysis. Compared with other methods, the experiments show that triggering path technique can detect more termination cases. Combined the two stage strategy, the time efficiency of the algorithm is close to relationship graph algorithm. The results also show that this method can be applied in other concurrent and dynamic network environments based on active rule.

Keywords active rule; triggering path; finite trigger ring; termination analysis; relationship graph

收稿日期:2011-01-18;最终修改稿收到日期:2011-06-28. 本课题得到国家自然科学基金(40801160,60902036,61070035)、国家“八六三”高技术研究发展计划重点项目基金(2007AA120400)以及高等学校博士学科点专项科研基金(20104307110017)资助.熊 伟,男,1976年生,博士,副教授,中国计算机学会(CCF)会员,研究方向为空间数据库与地理信息系统. E-mail: xiongwei@nudt.edu.cn. 吴 焯,男,1986年生,博士研究生、中国计算机学会(CCF)学生会会员,研究方向为空间数据库与并行处理技术.张 震,男,1986年生,硕士,研究方向为主动数据库和基于位置的服务技术.吴秋云,男,1971年生,副教授,研究方向为信息系统与高级数据库技术.

1 引言

随着以服务为形式的各种应用的发展,用户对网络服务的需求呈现个性化、定制化和主动化的趋势,主动规则成为解决这些问题的关键技术.例如,对等网络中,用户可以利用自定义的规则来进行网络学习和协作科研^[1].网络中用户感兴趣的最新学习资料、实验数据和结果都可以实时地向用户发布.主动规则用来实现节点或超级节点片段数据的更新,特殊之处在于 P2P 网络中参与规则计算和执行的节点数量是非常庞大的,对定义的规则集进行准确的终止性分析,能够提高 P2P 网络的执行效率;在传感器网络中,主动规则作为一种新的查询处理模式,可以应用于安全管理、环境监控,当出现危险情况时进行应急处理^[2].其与传统主动数据库应用的区别在于各个传感器节点数目众多且功能独立,规则终止性分析需要考虑其动态自组网络特点;商业数据库产品也纷纷推出规则管理器,以支持商品订购、位置服务等应用^[3],通过用户自主定制的主动规则,随时提供满足用户需求的商品或服务.其特点是引入了位置信息和空间查询.

从这些应用来看,主动规则的功能已经不再局限于传统的数据库完整性约束维护,而是逐渐用于满足更复杂和高级的商业应用.此外,这些支持规则技术的大规模复杂网络应用都具备类似特点,即常有大量的用户自定义规则集同时触发,网络结构也有可能发生动态变化,因此带来了规则并发行为的控制问题^[4].每当应用及其相关数据状态发生改变时,都会进行规则处理.由于规则动作的执行也会产生新的状态,从而会引起一组规则集的相互级联触发,这种级联触发可能会一直延续下去,无法在有限步内终止.终止性(Termination)问题是一个不可判定的问题^[5],规则处理行为的非结构化、不可预测性是主动规则设计的一大难题.为此,需要提供一些分析方法和工具,以使主动规则能够更加有效地运行.

主动规则常采用事件驱动的 ECA 规则模型^[6],即认为一个主动规则由事件(Event)、条件(Condition)、动作(Action)三部分组成,语义是:当规定的事件 E 发生且条件 C 成立时自动执行相应动作 A,并且在动作执行过程中接着逐个检查下一个被级联触发的规则,直到执行完为止.

为了使研究更具有普遍性,本文讨论的范围主要是 ECA 规则模型.首先综述以静态分析方法为

主的主动规则终止性分析相关工作;随后给出问题描述和相关形式化定义;分析基于关联图的终止性分析方法的保守性,通过定义触发路径,证明基于触发路径的规则终止性相关定理,并给出触发路径构建算法、优先级判定算法和规则集终止性分析算法.最后与相关分析方法进行比较.本文的方法能够更有效地检测主动规则集的终止性,并且顾及规则执行的优先级,不依赖于具体的主动数据库实现模型,复合事件、复杂条件等都可以很方便地集成于该方法,能够适应基于主动规则的其他应用.

2 相关工作

规则的终止性分析目前主要有两大类方法:静态分析方法和动态分析方法.静态分析方法关注于提出编译时保证规则集终止的充分条件,动态分析方法则通过规则执行仿真、动态分析方法等来获取运行时的终止性.

动态分析也称运行时分析方法,需要知道数据库的内部状态,分析结果也只能根据测试过的内部数据库状态获得^[5-7].由于该方法的高昂代价,在商业数据库中采取的是比较简单的实现方案,比如可以固定规则级联的深度或者指定规则执行的时间来控制规则终止,然而这种方法无疑会回滚掉许多本来可以终止的规则事务.

本文主要关注静态分析,相关研究涉及面广.Ray 等人^[8]提出的使用软件规范和验证的自动工具,如符号模型检查方法,将规则转化为一种模型检查器的规范,然后使用模型检查器通过穷尽搜索检查终止性是否保持;有学者将主动规则转化为形式化逻辑进行推理分析^[9].Baralis 等人^[10]采用代数方法对转化的规则进行更强的终止性分析,但是没有考虑触发条件的影响.另外 Bailey 等人使用抽象解释技术,基于抽象数据库状态来对主动规则的执行进行仿真,如果规则在抽象状态上终止,则真实的数据库状态也可以保证终止^[5].近几年,Petri 网理论也被引入到规则终止性分析方法中.Li 介绍了将 Petri 网划分为多个子网负责 ECA 规则模型的各个部分,分别对应于激活和触发状态,终止性分析是基于 Petri 网的可达性理论^[11];有学者对于给定任意规则集,构造了改进的谓词/变迁网,再根据该 Petri 网自动检测出循环规则集、不一致规则集和矛盾规则集^[9];Li 等人^[11]基于着色 Petri 网对主动规则行为建模,采用 Petri 网理论中的关联矩阵来检查规

则集中是否存在循环路径,但没有考虑规则的优先级,Baha-hamed 等人对该方法进行了扩展以支持规则优先级^[4].上述方法都是通过转化的策略,将主动规则转化到另一个技术比较成熟的领域来解决终止性分析问题.这些方法或者在转化语义上实现复杂,或者不够直观.

大部分静态分析方法研究以 Aiken 等人^[12]定义的触发图概念为基础进行扩展和改进. Baralis 等人^[7]通过引入激活图改善了触发图的条件分析能力.有学者提出精化触发图通过对触发图中触发公式的分析,可以得出触发图继续归约的算法^[9].精化触发图方法的改进方法如 Lee 等人^[13]提出的路径删除技术和解圈技术,路径删除技术解决了只删除一条边的分析方法的不足,综合考虑规则执行路径,同时考虑了可更新谓词/属性对规则终止性的影响,采用节点分割方法删除触发图中的规则路径,规则归约中可以删除整条路径而不是一个节点.解圈技术对于触发图中执行次数少于 k 次的圈,构造一个等价的复制节点的触发图,这样就将这个 k 圈“解开”并进行触发图归约.关于如何计算圈的 k 值,有学者提出一种基于约束的方法计算圈的最大迭代次数^[9].基于精化触发图,Couchot^[14]提出了一系列的改进方法,通过将原始的触发图进行分割,可以删除被无限次触发的规则节点,而且该节点的条件也可以是恒真的,从而检测出新的终止情况;也提出了具有复合事件的主动规则的静态终止性分析算法,扩展了触发图,触发图节点不仅仅包括规则,还包括事件;又提出复合路径的概念,采用处理规则的路径最大序 M 替换传统表示圈的方法.该方法对于合取事件和依次事件的主动规则尤其有效.针对规则执行优先级问题,Couchot 提出一种基于触发规则及路径的优先级的分析方法.左万利等人^[15]引入惰化图,提出了基于关联图的终止性分析算法,但该算法分析结果仍然存在保守性.Xiong 等人^[16]提出为有效活化路径建立条件公式的思想,在此基础上给出处理一个新的主动规则终止性判定方法.郝忠孝等人^[17]提出了判断规则集活化路径的同步无限次执行问题的方法,来分析终止性.熊伟等人^[9]采用标注事件图扩展 Couchot 的方法并应用于空间数据库主动规则终止性分析.

3 问题描述

定义 1. 规则集 $R = \{r_i \mid i = 1, 2, \dots, n\}$ 是所有

规则 r_i 的有限集合,规则 r_i 是形如 $\langle Evt, Cnd, Act \rangle$ 的三元组,其中 Evt 是事件, Cnd 是条件, Act 是动作.

定义 2. 对于任意的 $r_i \in R$, $pri(r_i)$ 是规则 r_i 的优先级,它是一个整数.对任意的 $r_i, r_j \in R, i \neq j$,若 $pri(r_i) > pri(r_j)$,则 r_i 优先 r_j 执行;反之,则 r_j 优先 r_i 执行.

尽管分析方法不同,规则执行模型中的耦合模式也不一样,但是主动规则的处理算法是相同的,规则处理算法 Rule_Process 的自然语言描述如下:

1. 无规则被触发则退出;
2. 令被触发规则的集合为 $R_T, \forall r_j \in R_T, j \neq i$,若 $pri(r_i) > pri(r_j)$,则选择 r_i ;
3. 从触发规则实例集中删除选择的规则实例;
4. 计算 r_i 的条件,若条件为真,则执行 r_i 的动作;
5. 转步 1.

基于规则执行模型,相关学者提出以下定义.

定义 3. 规则集 R 的触发图(triggering graph)为 TG ,它是由 $\langle R, TE \rangle$ 构成的有向图^[12].图中每个结点表示规则 $r_i \in R$,有向边 $\langle r_i, r_j \rangle \in TE$ 为触发边,表示规则 r_i 的执行产生某事件,该事件可能触发规则 r_j .在规则的触发图中,触发边用带箭头的实线表示.如果一个规则的执行会触发自身,则称之为自触发规则.

定义 4. 规则集 R 的活化图(activation graph)为 AG ,它是由 $\langle R, AE \rangle$ 构成的有向图.图中每个结点表示规则 $r_i \in R$,有向边 $\langle r_i, r_j \rangle \in AE$ 为活化边,表示规则 r_i 的执行可能会使规则 r_j 的条件为真^[7].在规则的活化图中,活化边用带箭头的虚线表示.如果一个规则的执行会使自身的条件为真,则称之为自活化规则.

定义 5. 规则集 R 的惰化图(deactivation graph)为 DG ,它是由 $\langle R, DE \rangle$ 构成的有向图.图中每个结点表示规则 $r_i \in R$,有向边 $\langle r_i, r_j \rangle \in DE$ 为惰化边,表示规则 r_i 的执行一定使规则 r_j 的条件为假^[15].在规则的惰化图中,惰化边用带箭头的双实线表示.如果一个规则的执行使自身的条件为假,则称之为自惰化规则.

定义 6. 令规则集 R 的触发图为 $TG = \langle R, TE \rangle$,活化图为 $AG = \langle R, AE \rangle$,惰化图为 $DG = \langle R, DE \rangle$,则 R 的关联图(relationship graph)为 $RG = \langle R, E \rangle$,其中 R 为所有规则结点的集合, $E = TE \cup AE \cup DE$ 是所有边的集合^[15].

定义 7. 规则 r 由规则集 R 直接触发可达是

指, $\exists r' \in R, r'$ 触发 r . 规则 r 由规则集 R 触发可达是指, r 由 R 直接触发可达, 或 $\exists r', r'$ 触发 r 且 r' 由 R 触发可达.

令 $r_i, r_j \in R$, 由规则 r_i 到规则 r_j 的触发可达集 $tr(r_i, r_j)$ 是规则集 R 的一个子集, 其中包含 r_j 但不包含 r_i , 它满足如下特性:

- (1) 使用规则集 $tr(r_i, r_j)$, 由 r_i 触发可达 r_j ;
- (2) $tr(r_i, r_j)$ 是最小的, 即删除 $tr(r_i, r_j)$ 中的任一规则, 性质(1)不再成立.

由规则 r_i 到规则 r_j 的触发可达集可能不止一个, 这些可达集的集合记为 $Str(r_i, r_j)$.

定义 8. 触发可达集的 $tr(r_i, r_j)$ 的制约优先级是指 $tr(r_i, r_j)$ 中所有规则的最小优先级. 触发可达集集合 $Str(r_i, r_j)$ 中的最大和最小制约优先级分别记为 $p_{i \rightarrow j}^{\max}$ 和 $p_{i \rightarrow j}^{\min}$.

定义 9. 令规则集 R 的触发图为 $TG = \langle R, TE \rangle$, 若 $\exists r_1, r_2, \dots, r_k \in R, \langle r_1, r_2 \rangle \in TE, \langle r_2, r_3 \rangle \in TE, \dots, \langle r_k, r_1 \rangle \in TE$, 则称 $\rho = \langle r_1, r_2, \dots, r_k \rangle$ 为 TG 的一个触发环. 触发环 ρ 上所有规则的集合记为 C_ρ .

定义 10. 令 ρ 为一个触发环, $r \notin C_\rho, r$ 由 C_ρ 触发可达, $\exists r_c \in C_\rho, tr(r_c, r) \neq \emptyset$ 且 $tr(r_c, r) \cap C_\rho = \emptyset$, 则称 r_c 为 r 与 ρ 的交点规则. 即 r 与 ρ 的交点规则 r_c 位于触发环 ρ 上, 且由 r_c 出发不经过 ρ 上的其它规则触发可达 r .

基于上述定义, 通过分析关联图的特征, 可以对规则集终止性进行判定. 如: 基于触发图和活化图、触发环和活化环是主动规则集呈现非终止行为的必要条件^[7]. 即对于任意 $r_i \in R$, 若不存在与之相连的触发边或活化边, 则 r_i 不会执行多次, 连同其相关边可在关联图中去掉, 这个步骤即规则集的基本归约算法 Basic_Reduce, 如果规则集能够归约为空, 那么表明其行为能够终止.

4 基于触发路径的终止性分析算法

4.1 关联图算法保守性分析

文献[15]提出了基于关联图的终止性分析算法, 但对禁止触发环的定义中条件不够充分, 没有明确惰化规则. 本文在此基础上进行改进, 并分析该算法的保守性.

定义 11. 令 ρ 为不可归约集 R' 上的一个触发环, $r_o \in C_\rho, \langle r_o, r_o \rangle \in DE, r_t \in C_\rho, \langle r_t, r_o \rangle \in TE, \langle r_t, r_o \rangle \notin AE$, 即 r_o 为环上的一个自惰化规则, r_t 为

环上触发但不活化 r_o 的规则. $\forall r_a \in R', \langle r_a, r_o \rangle \in AE, tr(r_o, r_a) \neq \emptyset$ 即 r_a 由 r_o 触发可达, 当满足以下条件之一成立时, 称 r_o 为被禁止的规则, 并称 ρ 为被禁止的触发环:

- (1) $r_a \in C_\rho, \exists r_d \in C_\rho, \langle r_d, r_o \rangle \in DE$, 使 ρ 呈 $(r_o, \dots, r_a, \dots, r_d, \dots, r_t)$ 形式;
- (2) $r_a \notin C_\rho$, 令 r_c 为 r_a 与 ρ 的交点规则, $\exists r_d \in C_\rho, \langle r_d, r_o \rangle \in DE$, 使触发环 ρ 的形式为 $(r_o, \dots, r_c, \dots, r_d, \dots, r_t)$, 且 $p_{c \rightarrow d}^{\max} < p_{c \rightarrow a}^{\min}$;
- (3) $r_a \notin C_\rho$, 令 r_c 为 r_a 与 ρ 的交点规则, $p_{c \rightarrow a}^{\max} < p_{c \rightarrow t}^{\min}$ 且 $p_{c \rightarrow a}^{\max} < pri(r_o)$.

定理 1. 对于不可归约集 R' , 如果由于拓扑结构和优先级级别的设置使得 R' 上的所有触发环都被禁止, 则仅由 R' 确定的规则集不会呈现非终止行为^[15].

相应的终止性分析算法如下.

算法 1. 基于关联图的终止性分析算法 Termination_RG.

输入: 关联图 $G = \langle R', E \rangle$

输出: 非被禁止的触发环及其可达的边

begin

for each $\rho_i \in G$ 标记为非被禁止;

for each $\rho_i \in G$ do

$R_o := \{r_o \mid \exists r_d \in C_{\rho_i}, \langle r_d, r_o \rangle \in DE \text{ or } r_o \text{ 不被 } C_{\rho_i} \text{ 中的规则活化}\};$

while (ρ_i 非被禁止 and $R_o \neq \emptyset$) do

{ for each $r_o \in R_o$ do

$\{R_a := \{r_a \mid \langle r_a, r_o \rangle \in AE, \text{ and } tr(r_o, r_a) \neq \emptyset\};$

if ($\forall r_a \in R_a, r_a \in C_{\rho_i}, \exists r_d \in C_{\rho_i}, \langle r_d, r_o \rangle \in DE$ and ρ_i 形式为 $(r_o, \dots, r_a, \dots, r_d, \dots, r_t)$)

then 标记 ρ_i 为被禁止;

if ($(\forall r_a \in R_a, r_a \notin C_{\rho_i}, \exists r_d \in C_{\rho_i}, \langle r_d, r_o \rangle \in DE$ and $p_{c \rightarrow d}^{\max} < p_{c \rightarrow a}^{\min}, r_c$ 为 r_a 与 ρ_i 的交点规则 and ρ_i 形式为 $(r_o, \dots, r_c, \dots, r_d, \dots, r_t)$)

then 标记 ρ_i 为被禁止;

if ($(\forall r_a \in R_a, r_a \notin C_{\rho_i}, p_{c \rightarrow a}^{\max} < p_{c \rightarrow t}^{\min}, r_c$ 为 r_a 与 ρ_i 的交点规则) and $p_{c \rightarrow a}^{\max} < pri(r_o)$ and ρ_i 形式为 $(r_o, \dots, r_c, \dots, r_t)$)

then 标记 ρ_i 为被禁止;

}

for all 没被禁止的 ρ_i

return (ρ_i 是没被禁止的触发环);

end

例 1. 考虑如图 1 所示的规则集 R' , 该规则集不可用基本归约算法归约, 是不可归约集.

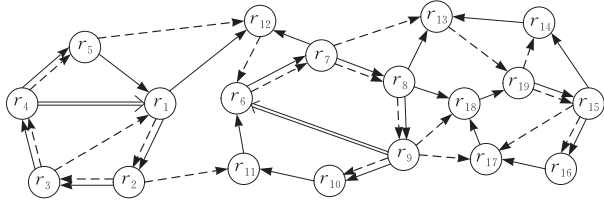


图 1 基于关联图的终止性分析算法

R' 包含 3 个触发环: $\rho_1 = (r_1, r_2, r_3, r_4, r_5)$, $\rho_2 = (r_6, r_7, r_8, r_9, r_{10}, r_{11})$, $\rho_3 = (r_{19}, r_{15}, r_{16}, r_{17}, r_{18})$. 令 $\forall r_i \in R'$, $\text{pri}(r_i) = i$, 且 r_i 是自惰化规则, 则对 $\rho_1: r_o = r_1, r_a = r_5, r_t = r_5, \exists r_d = r_4, \rho_1$ 呈 $(r_o, \dots, r_a, \dots, r_d, \dots, r_t)$ 形式, 满足定义 11 的条件(1), r_1 是被禁止的规则, ρ_1 是被禁止的触发环; 对 $\rho_2: r_o = r_6, r_a = r_{12}, r_t = r_{11}, r_c = r_7$ 是 r_a 与 ρ_2 的交点规则, ρ_2 呈 $(r_o, \dots, r_c, \dots, r_d, \dots, r_t)$ 形式, 且 $\text{Str}(r_7, r_9) = \{\{r_8, r_9\}\}$, $p_{c-d}^{\max} = p_{7-9}^{\max} = 8$, $\text{Str}(r_7, r_{12}) = \{\{r_{12}\}\}$, $p_{c-a}^{\min} = p_{7-12}^{\min} = 12$, $p_{c-d}^{\max} < p_{c-a}^{\min}$, 满足条件(2), r_6 是被禁止的规则, ρ_2 是被禁止的触发环; 对 $\rho_3: r_o = r_{19}, r_a = r_{13}, r_t = r_{18}, r_c = r_{15}$ 是 r_a 与 ρ_3 的交点规则, $\text{Str}(r_{15}, r_{13}) = \{\{r_{14}, r_{13}\}\}$, $p_{c-a}^{\max} = p_{15-13}^{\max} = 13$, $\text{Str}(r_{15}, r_{18}) = \{\{r_{16}, r_{17}, r_{18}\}\}$, $p_{c-t}^{\min} = p_{15-18}^{\min} = 16$, $p_{c-a}^{\max} < p_{c-t}^{\min}$, $p_{c-a}^{\max} < \text{pri}(r_{19})$, 满足条件(3), r_{19} 是被禁止的规则, ρ_3 是被禁止的触发环. 综上, 根据算法 1 分析可知 R' 是可终止的规则集.

根据重新定义的有限触发环, 基于关联图的终止性分析算法分析结果比基本归约算法更精确, 但仍存在一定的保守性. 即由该算法分析可终止的规则集一定不存在非终止行为, 然而, 存在非有限触发环的规则集也有可能终止. 如例 2 所示.

例 2. 令规则集 $R = \{r_1, r_2, r_3, r_4, r_5, r_6\}$, $\text{pri}(r_i) = i$, 其关联图如图 2 所示.

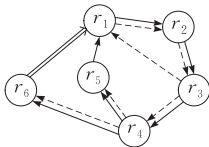


图 2 基于关联图修正算法的保守性示意

该规则集是不可归约集, 且不满足文献[15]中禁止触发环的任一条件, 故由基于关联图的终止性分析修正算法不能判定 R 是非终止的. 但是, 我们按以下步骤分析: r_3 的执行活化 r_1 并活化触发 r_4 ; r_4 的执行同时活化触发 r_5 和 r_6 ; 由于 $\text{pri}(r_6) > \text{pri}(r_5)$, 先执行 r_6 , r_6 的执行惰化 r_1 ; 再执行 r_5 , r_5 的执行触发 r_1 , 此时 r_1 条件为假不会执行. 因此, 规则 r_1 不会多次执行, 该规则集 R 是可以终止的.

因此, 在基于关联图的修正算法基础上, 为了提高规则集终止性分析的精确性, 引入触发路径, 并提出基于触发路径的主动规则集终止性分析算法.

4.2 触发路径相关定义

为了表达规则集中两个规则之间的触发关系, 本文提出了触发路径等概念, 并提出触发路径构建算法和触发路径的优先级判定算法.

定义 12. 令规则集 $R = \{r_1, r_2, \dots, r_N\}$, 若 $\langle r_i, r_{i+1} \rangle \in TE, \langle r_{i+1}, r_{i+2} \rangle \in TE, \dots, \langle r_{j-1}, r_j \rangle \in TE, i \neq j$, 则存在一条 r_i 到 r_j 的触发路径 $t\text{path}(r_i, r_j) = (r_i, r_{i+1}, r_{i+2}, \dots, r_{j-1}, r_j)$. 规则 r_i 到 r_j 的触发路径可能不止一条, 称这些路径的集合为 r_i 到 r_j 的触发路径集, 记为 $\text{Stpath}(r_i, r_j)$.

若触发路径 $t\text{path}(r_i, r_j)$ 的所有规则顶点都不重复出现, 则称该触发路径为简单触发路径, 否则称复合触发路径. 显然, 复合触发路径构成触发环.

定义 13. 考虑规则集 $R = \{r_1, r_2, \dots, r_N\}$, 若: $t\text{path}(r_m, r_n) = (r_m, r_{m+1}, \dots, r_{n-1}, r_n), m, n < N, t\text{path}(r_i, r_j) = (r_i, \dots, r_m, r_{m+1}, \dots, r_{n-1}, r_n, \dots, r_j), i, j < N$, 则称触发路径 $t\text{path}(r_m, r_n)$ 包含于触发路径 $t\text{path}(r_i, r_j)$, 记为 $t\text{path}(r_m, r_n) \subseteq t\text{path}(r_i, r_j)$. 若 $t\text{path}(r_m, r_n) \subseteq t\text{path}(r_i, r_j), t\text{path}(r_i, r_j) \subseteq t\text{path}(r_m, r_n)$, 则称触发路径 $t\text{path}(r_m, r_n)$ 和触发路径 $t\text{path}(r_i, r_j)$ 完全重合. 记为 $t\text{path}(r_m, r_n) = t\text{path}(r_i, r_j)$.

定义 14. 考虑规则集 $R = \{r_1, r_2, \dots, r_N\}$, 若: $t\text{path}(r_i, r_j) = (r_i, r_{i+1}, \dots, r_s, r_m, \dots, r_j)$, $t\text{path}(r_i, r_k) = (r_i, r_{i+1}, \dots, r_s, r_n, \dots, r_k)$, 即这两条路径自 r_s 之前的路径相同, r_s 之后的路径不同, 记为 $t\text{path}(r_i, r_j) \wedge t\text{path}(r_i, r_k) = (r_i, \dots, r_s)$, 则称 r_s 为触发路径 $t\text{path}(r_i, r_j)$ 和 $t\text{path}(r_i, r_k)$ 的分离点, 若 $r_s = r_i$, 则称 $t\text{path}(r_i, r_j)$ 和 $t\text{path}(r_i, r_k)$ 分离; 若 $r_s \neq r_i$, 则称 $t\text{path}(r_i, r_j)$ 和 $t\text{path}(r_i, r_k)$ 部分分离.

令规则集 R 的触发图为 TG , 则由 v_o 到 v_t 的触发路径 $t\text{path}(r_o, r_t)$ 可由以下算法得出.

算法 2. 触发路径构建算法 Build_TP.

输入: 触发图 TG , 出发规则 r_o , 目标规则 r_t

输出: 触发路径集 L

begin

$l_o := r_o; \text{visited}[r_o] := \text{TRUE};$

if $r_o = r_t$ then add l_o to L ;

$w := \text{FIRSTADJ}(TG, r_o);$ // 找邻接点

while $w \neq 0$ do

{if not $\text{visited}[w]$ and $w \neq r_t$

then $\text{Build_TP}(TG, w, r_t, L);$

```

for each  $l \in L$  do  $l = \text{link}(l_o, l)$ 
 $w := \text{NEXTADJ}(TG, r_o, w)$ ; //找下一邻接点
}
end

```

定理 2. 算法 2 是正确的,可终止的,时间复杂度为 $O(n^2)$. 其中 n 为触发图节点个数.

证明. 正确性. 该算法从规则 r_o 出发进行深度优先遍历,当访问到 r_i 时找到一条触发路径并存储,存储的路径满足定义 12,而且由于是遍历,所以所有路径都会被找到,故算法是正确的.

可终止性. 该算法的主要步骤是由一个规则出发进行深度优先遍历,由于规则集有限,算法是可终止的.

时间复杂度. 遍历过程的实质是对每个顶点查找其邻接点,耗费的时间取决于所采用的存储结构. 仍采用邻接矩阵作为触发图的存储结构,找邻接点所需时间为 $O(n^2)$,其中 n 为触发图节点个数.

综上,算法 2 是正确的,可终止的,其时间复杂度为 $O(n^2)$. 证毕.

定义 15. 考虑规则集 $R = \{r_1, r_2, \dots, r_N\}$, 存在 r_o 至 r_i 和 r_j 的触发路径 $t\text{path}(r_o, r_i)$ 和 $t\text{path}(r_o, r_j)$, $i \neq j$, 当满足下列条件之一时,称 $t\text{path}(r_o, r_i)$ 的优先级高于 $t\text{path}(r_o, r_j)$, 记为 $p(t\text{path}(r_o, r_i)) > p(t\text{path}(r_o, r_j))$:

- (1) $t\text{path}(r_o, r_i) \subseteq t\text{path}(r_o, r_j)$, 即 $t\text{path}(r_o, r_i)$ 包含于 $t\text{path}(r_o, r_j)$;
- (2) 若 $t\text{path}(r_o, r_i)$ 和 $t\text{path}(r_o, r_j)$ 分离, $p_{o-j}^{\max} < p_{o-i}^{\min}$;
- (3) 若触发路径 $t\text{path}(r_o, r_i)$ 和 $t\text{path}(r_o, r_j)$ 部分分离,它们的分离点为 r_s , 且 $p(t\text{path}(r_s, r_i)) > p(t\text{path}(r_s, r_j))$.

定理 3. 令规则 r_o 至 r_i 和 r_j 的触发路径集为 $S_1 = \text{Stpath}(r_o, r_i)$ 和 $S_2 = \text{Stpath}(r_o, r_j)$, 若 $\forall t\text{path}(r_o, r_i) \in S_1, t\text{path}(r_o, r_j) \in S_2$, 有 $p(t\text{path}(r_o, r_i)) > p(t\text{path}(r_o, r_j))$, 则 r_i 将优先于 r_j 被 r_o 触发执行,记为 $\text{ptr}(r_o, r_i) > \text{ptr}(r_o, r_j)$.

证明. 首先证明若 $p(t\text{path}(r_o, r_i)) > p(t\text{path}(r_o, r_j))$, 则在只考虑这两条触发路径的情况下, r_i 优先于 r_j 被 r_o 触发执行. 按定义 4 分 3 种情况考虑:

- (1) $t\text{path}(r_o, r_i) \subseteq t\text{path}(r_o, r_j)$, 则 r_j 的触发依赖于 r_i 的执行,故 r_i 优先于 r_j 被 r_o 触发执行;
- (2) $p_{o-j}^{\max} < p_{o-i}^{\min}$, 显然 r_i 优先于 r_j 被 r_o 触发执行;

(3) 由于分离点为 r_s, r_i 和 r_j 的触发都依赖于 r_s 的执行,又由于 $p(t\text{path}(r_s, r_i)) > p(t\text{path}(r_s, r_j))$, r_i 优先于 r_j 被 r_s 触发执行,故 r_i 优先于 r_j 被 r_o 触发执行.

然后用反证法. 假设 r_j 优先于 r_i 被 r_o 触发,则存在两条路径 $t\text{path}(r_o, r_i) \in S_1$ 和 $t\text{path}(r_o, r_j) \in S_2$, $p(t\text{path}(r_o, r_j)) > p(t\text{path}(r_o, r_i))$, 这与条件矛盾,故假设不成立.

综上,若 $\forall t\text{path}(r_o, r_i) \in S_1, t\text{path}(r_o, r_j) \in S_2$, 有 $p(t\text{path}(r_o, r_i)) > p(t\text{path}(r_o, r_j))$, 则 r_i 将优先于 r_j 被 r_o 触发. 证毕.

需要说明的是,对于包含触发环的复合路径,假设复合路径为 $t\text{path}(r_1, \dots, r_i, r_j, r_1 \dots r_n)$, 则由于 $t\text{path}(r_1, \dots, r_i, r_j, r_1) \subseteq t\text{path}(r_1, \dots, r_i, r_j, r_1 \dots r_n)$, 所以 $\text{ptr}(r_1, \dots, r_i, r_j, r_1) > \text{ptr}(r_1, \dots, r_i, r_j, r_1, \dots, r_n)$. 因此在终止性分析时只需要考虑其中触发环部分即可,对于含多个触发环的复合路径,要一一判断.

算法 3. 触发路径的优先级判定算法 Determination_Priority.

输入: $G = \langle R, E \rangle, r_o, r_i, r_j$

输出: TRUE 表示规则 r_i 优先 r_j 被 r_o 触发; FALSE 表示不确定

begin

$S_1 := 0; S_2 := 0;$

$\text{Build_TP}(TG, r_o, r_i, S_1);$

$\text{Build_TP}(TG, r_o, r_j, S_2);$

$\text{result} := \text{TRUE};$

{for each $p(t\text{path}(r_o, r_i)) \in S_1$

for each $p(t\text{path}(r_o, r_j)) \in S_2$

if $(t\text{path}(r_o, r_i) \subseteq t\text{path}(r_o, r_j))$ then continue;

else if $(t\text{path}(r_o, r_i) \wedge t\text{path}(r_o, r_j) = (r_o))$ and $p_{o-j}^{\max} < p_{o-i}^{\min}$

then continue;

else if $((t\text{path}(r_o, r_i) \wedge t\text{path}(r_o, r_j) = (r_o, \dots, r_s))$ and $\text{Determination_Priority}(R, r_s, r_i, r_j) = \text{TRUE})$ then continue;

else { $\text{result} := \text{FALSE};$ break;}

}

return (result);

end

根据定理 3 可知该算法是正确的. 算法主要步骤是分别从两个触发路径集中取触发路径进行对比,由于触发路径集有限,所以触发路径的个数也是有限的,故该算法可以终止. 设 n 为触发环中规则的

个数,构造触发路径集的时间复杂性为 $O(n^2)$,比较触发路径优先级的时间复杂性为 $O(n^2)$,故该算法的时间复杂性为 $O(n^2)$.

4.3 基于触发路径主动规则集终止性分析

定义 16. 考虑不可归约集 R' ,令 ρ 为 R' 上的一个触发环, $r_o \in C_\rho, r_t \in C_\rho, \langle r_t, r_o \rangle \in TE, \langle r_t, r_o \rangle \notin AE$,即 r_t 为环上触发但不活化 r_o 的规则; $\forall r_a \in R', \langle r_a, r_o \rangle \in AE, \langle r_a, r_o \rangle \notin TE, tr(r_o, r_a) \neq \emptyset$ 即 r_a 由 r_o 触发可达; $\forall r_d \in R', \langle r_d, r_o \rangle \in DE, tr(r_o, r_d) \neq \emptyset$ 即 r_d 由 r_o 触发可达.当满足下列条件之一时,称 r_o 为有限规则,并称 ρ 为有限触发环:

- (1) $ptr(r_o, r_a) > ptr(r_o, r_d) > ptr(r_o, r_t)$;
- (2) $ptr(r_o, r_d) > ptr(r_o, r_t) > ptr(r_o, r_a)$,且 $p_{o-a}^{\max} < pri(r_o)$.

定理 4. 令 ρ 为不可归约集 R' 上唯一的触发环,如果 ρ 是有限触发环,则 R' 具有可终止性.

证明. 分情况证明:

(1) 由于 $ptr(r_o, r_a) > ptr(r_o, r_d) > ptr(r_o, r_t)$,故 r_a 优先于 r_d 被 r_o 触发执行, r_d 优先于 r_t 被 r_o 触发执行,所以 r_o 先被活化,再被惰化,再被触发, r_o 不会多次执行,故则 R' 具有可终止性.

(2) 由于 $ptr(r_o, r_d) > ptr(r_o, r_t) > ptr(r_o, r_a)$,故 r_d 优先于 r_t 被 r_o 触发执行, r_t 优先于 r_a 被 r_o 触发执行, r_o 先被惰化,再被触发,再被活化;又由于 $p_{o-a}^{\max} < pri(r_o)$, r_o 会在被活化之前选择执行;而此时 r_o 条件为假,不会多次执行,故 R' 具有可终止性.

综上,如果 ρ 是有限触发环, R' 具有可终止性.

证毕.

定理 5. 对于不可归约集 R' ,如果由于拓扑结构和优先级别的设置使得 R' 上的所有触发环是有限的,则仅由 R' 确定的规则集不会呈现非终止行为.

证明. 反证法.假定 R' 呈非终止行为,对于 R' 上的任一触发环 ρ_i ,令 r_i 为 ρ_i 上代表制约优先级的规则.由于 ρ_i 是有限的,其上规则的无限执行只能依赖于另外的呈非终止行为的触发环,且另外的触发环必须具有相对较低的制约优先级.令 ρ_j 就是这样一个触发环, r_j 是 ρ_j 上代表其制约优先级的规则,那么有 $pri(r_j) < pri(r_i)$.同理, ρ_j 上规则的无限执行依赖于另外的呈非终止行为且具有相对较低制约优先级别的触发环,令 ρ_k 为这样一个触发环, r_k 为 ρ_k 上代表其制约优先级的规则,则有 $pri(r_k) < pri(r_j)$.依次下去,由于 R' 中的规则个数有限,根据上述制约优先级之间的单向依赖关系,必在有限步骤之后找到某一触发环 ρ_l ,该触发环没有可以依赖的触发

环,那么 ρ_l 必须是呈非终止行为才能使规则集 R' 非终止,这与所有触发环是有限的相矛盾. 证毕.

算法 4. 基于触发路径的终止性分析算法

Termination_TP.

输入: $G = \langle R, E \rangle$

输出: 触发环及其可达的边; NULL 表示规则集 R 一定终止

begin

$G' := Basic_Reduce(G)$;

if ($G' = \text{NULL}$) then return (NULL);

else

标记 G' 上的触发环 ρ 为无限触发环;

for each $\rho \in G'$ do

$R_o := \{r_o \mid r_o \in C_\rho, \exists r_t \in C_\rho, \langle r_t, r_o \rangle \in TE, \langle r_t, r_o \rangle \notin AE\}$;

while(ρ 是无限触发环) and $R_o \neq \emptyset$ do

for each $r_o \in R_o$ do

$R_a := \{r_a \mid tpath(r_o, r_a) \neq \emptyset, \langle r_a, r_o \rangle \notin TE, \langle r_a, r_o \rangle \in AE\}$;

$R_d := \{r_d \mid tpath(r_o, r_d) \neq \emptyset, \langle r_d, r_o \rangle \in DE\}$;

if $\forall r_a \in R_a, r_d \in R_d$

($Determination_Priority(G', r_o, r_a, r_d)$ and

$Determination_Priority(G', r_o, r_d, r_t)$) or

($Determination_Priority(G', r_o, r_d, r_t)$ and

$Determination_Priority(G', r_o, r_t, r_a)$ and

$p_{o-a}^{\max} < pri(r_o)$)

then 标记 ρ 是有限触发环;

for 所有没有被标记为有限的触发环 ρ do

return ρ 及其可达的边;

return(NULL);

end

由定理 4 可知该算法是正确的.由于规则集中的规则个数有限,若 n 为触发环中规则的个数,该算法是可终止的.该算法分两个部分:第一部分调用基本归约算法,其时间复杂度为 $O(n^2)$;第二部分对每个满足条件的规则调用触发路径优先级判定算法,其时间复杂度为 $O(n^3)$.故算法的总体时间复杂度为 $O(n^3)$. 证毕.

由于该算法时间复杂度较高,因此可以采用两阶段算法加以改进,第一阶段采用基本归约算法和关联图分析算法,无法得到终止性结论的规则集才会进入第二阶段,从而降低了规则集的规模,减少了分析时间.

算法 5. 两阶段终止性分析算法 Termination_2P.

输入: $G = \langle R, E \rangle$

输出: 触发环及其可达的边; NULL 表示规则集 R 一定终止

```

begin
   $G' := \text{Basic\_Reduce}(G)$ ;
  If ( $G' = \text{NULL}$ ) then return (NULL);
  else
     $G'' := \text{Termination\_RG}(G, E)$ ;
    If ( $G'' = \text{NULL}$ ) then return (NULL);
    else
      return  $\text{Termination\_TP}(G, E)$ ;
  end
end

```

5 与相关算法的比较

下面将基于触发路径的终止性分析算法与相关算法进行比较。

例 3. 对于例 2 中的规则集 $R = \{r_1, r_2, r_3, r_4, r_5, r_6\}$, $\text{pri}(r_i) = i$, 其关联图如图 2 所示。

应用文献[5]的方法, 通过抽象数据库状态来模拟规则的处理, 需要列出规则处理前所有的初始触发规则. 在例 2 中, 数据库事务可以在规则处理前触发每条规则任意次数目的出现, 如 $\rho = (r_1, r_2, r_3, r_4, r_5, r_1)$ 中 r_1 的两次出现和 r_2 的一次出现, 无法准确列出初始触发规则, 该方法也无法保证规则集的终止。

应用文献[7]的方法, 触发环 $\rho = (r_1, r_2, r_3, r_4, r_5, r_1)$ 中任何一条边无法移除, 因此无法得到终止的结论。

应用文献[13]的方法, 由 Lee 给出的算法, 触发环 $\rho = (r_1, r_2, r_3, r_4, r_5, r_1)$ 的激活公式只能包含不被规则动作更新的属性, 或者是可以从触发图中删除的规则所更新的属性. 如果环中任一规则包含被规则 r_6 所更新的属性, 则解圈方法无法保证终止性。

应用文献[17]的方法, 图 2 中存在一个触发环 $\rho = (r_1, r_2, r_3, r_4, r_5, r_1)$ 和一个活化环 $\rho' = (r_1, r_2, r_3, r_1)$, 由于 ρ' 中每个规则都处于触发环 ρ 中, 或者由 ρ 可达, 并且每次自惰化后均可再次被活化, 因此判定此规则集具有不可终止性。

由图 2 可知, 该规则集是不可归约集, 且其触发环不满足文献[15]中关于被禁止的触发环的定义, 故由基于关联图的终止性分析算法不能判定 R 可终止。

在时间复杂度方面, 基于触发路径的终止性分析算法的时间复杂度为 $O(n^3)$, 基于关联图的终止性分析算法的时间复杂度为 $O(n^2)$, 因此基于触发

路径的终止性分析算法比基于关联图的终止性分析算法在时间复杂度高. 然而, 在大规模的网络环境中, 触发路径方法的贡献在于辅助应用开发人员设计出更加合理有效的规则集, 虽然规则分析精度的提高带来分析效率的降低, 但不会影响到应用本身性能, 而是提高了应用设计的灵活性. 对于使用规则的用户来说, 规则集已经成为了应用系统的一部分, 复杂度较高的分析在设计阶段已经完成了, 即使是在传感器网络这样规则集会出现动态变化的环境, 采用两阶段算法也可以提高分析效率。

在分析结果的精确性方面, 基于触发路径的终止性分析算法比基于关联图的终止性分析算法的精确性更高. 文献[15]中的定义禁止触发环的条件(1)和(2)是本文有限触发环定义条件(1)的子集, 其条件(3)是本文定义条件(2)的子集, 因此被禁止的环一定是有限环, 即由基于关联图的算法判定可终止的规则集, 用基于触发路径的算法分析也一定是可终止的. 而对于用基于关联图的算法分析不一定终止的规则集, 用基于触发路径的算法有可能判定可终止. 对于例 2, 应用本文提出的基于触发路径的终止性分析方法进行分析, 规则 r_1 满足本文有限触发环定义, 是有限规则, 故触发环 $\rho = (r_1, r_2, r_3, r_4, r_5, r_1)$ 是有限环, 由基于触发路径的分析算法可知规则集 R 是可终止的规则集. 因此, 基于触发路径的终止性分析算法比上述其它算法的分析结果更精确, 进一步提高了规则集终止性分析的精确性。

6 实验分析

以下通过实验分析 Basic_Reduce、Termination_RG、Termination_TP、Termination_2P 4 个算法的准确性和性能. 性能测试中, Termination_RG、Termination_TP 算法均先执行 Basic_Reduce 算法过滤规则集。

实验环境设置为: Intel Pentium 双核 CPU 主频 2.6 GHz, 内存 2048 MB, 算法采用 C 语言实现, 操作系统为 Windows 2003 Server. 实验数据采用随机生成规则节点数, 规则节点之间随机生成触发边、活化边或惰化边(根据实际应用, 惰化边与触发边、活化边不同时出现), 各种边的方向随机, 共生成 1000 组. 测试生成的规则集 1000 组中能够得到终止性结论的百分比以及得到结果所需要的时间如图 3 所示。

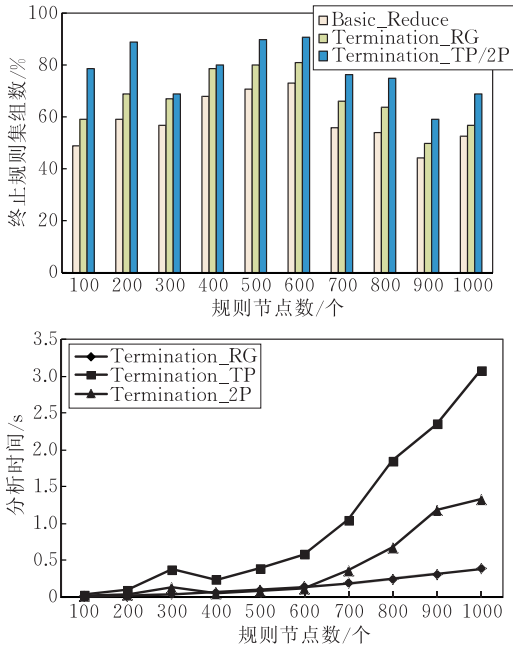


图3 触发路径分析算法实验结果

根据实验结果, Basic_Reduce 算法得到终止性结论的百分比最低, 依次是 Termination_RG 和 Termination_TP、Termination_2P 算法 (Termination_2P 主要提高 Termination_TP 分析效率, 不改变其结论), 表明触发路径分析方法提高了规则集终止性分析的准确性. 从执行时间来看, 如果不采用两阶段分析方法, Termination_TP 分析时间超出 Termination_RG 很多, 采用两阶段算法后, 在 700 个规则节点规模内, Termination_2P 分析时间几乎接近 Termination_RG 算法.

考虑到 P2P、传感器网络中应用具有多用户并发、规则集规模大、网络动态变化等特点, 进一步基于 Loadrunner 8.0 软件仿真测试规则分析方法在多用户查询并发时负载情况. 实验通过 1000 次随机增加和删除规则节点, 分别考察规则集基数为 500 时, 分析方法随用户访问数变化的负载情况以及用户访问数为 60 时, 分析方法随规则集基数变化时的负载情况. 实验结果如图 4 所示.

根据实验结果, 由于规则分析时间相对于仿真设置的多用户查询访问时间消耗很少, 因此, 3 种方法总体负载情况相差不大. 规则集基数为 500 时, 随用户访问数增加, 系统平均响应时间也增加, Termination_TP 平均响应时间最长, Termination_RG 和 Termination_2P 算法平均响应时间几乎相同. 表明多用户访问时, 即使规则集发生动态变化, 规则分析方法性能差异也不大. 对于用户访问数一定的情

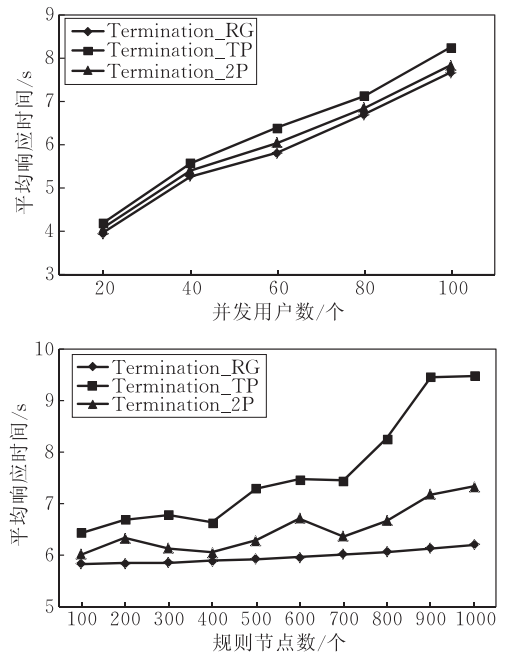


图4 规则分析算法多用户负载实验结果

况, 随着规则集数目增加, Termination_RG 算法性能最为稳定, 平均响应时间呈线性增长, 但增长幅度很小, 表明在规则动态分析中, 如果多用户并发访问中的网络延迟、任务调度等时间消耗远超过规则分析时间时, 规则数目增加与分析时间基本无关. 而 Termination_TP 和 Termination_2P 算法性能随规则集数目增加而呈下降趋势, 并且中间出现动荡, 可能是因为随机增加和删除规则节点产生的性能差异, 但 Termination_2P 算法总体性能好于 Termination_TP, 而接近于 Termination_RG.

7 结 语

本文分析了触发环中规则不会多次执行的两种情形:

- (1) 先被活化, 再被惰化, 最后被触发;
- (2) 先被惰化, 再被触发, 再被活化.

基于此原则, 引入了触发路径和有限触发环的概念, 根据触发路径优先级判断规则触发环是否是有限触发环, 提出了基于触发路径的终止性分析算法, 提高了分析结果的精确性. 本文算法已经采用 C 语言实现, 并将其应用于开源数据库 Ingres 的规则管理器. 本文方法与基于关联图的规则终止性分析算法相比时间复杂度略高, 但采用第一阶段进行基本归约和关联图分析, 第二阶段进行触发路径分析后, 能够保证分析效率不会降低太多, 并且提高了分

析的准确性. 实验结果表明, 基于触发路径的主动规则集终止性分析算法结合两阶段分析策略后, 能够很好地适应规则集动态变化和多用户并发的负载情况. 下一步的工作是将复合事件和复杂条件集成到该分析方法中, 进一步加强本文算法的适应能力.

致 谢 感谢墨西哥 Cinvestav-IPN Xiaoou Li 教授提供了相关参考资料!

参 考 文 献

- [1] Papamarkos G, Poulouvasilis A, Wood P T. Performance modelling of event-condition-action rules in P2P networks. *Journal of Computer and System Sciences*, 2011, 77(4): 621-636
- [2] Zoumboulakis M, Roussos G, Poulouvasilis A. Active rules for sensor databases//*Proceedings of the DMSN'04*. Toronto, Canada, 2004: 98-103
- [3] Beauregard B. Oracle Corp. Oracle Database 11g Complex Event Processing with Rules Manager. USA: Oracle Corporation, 2009
- [4] Baba-Hamed L, Belbachir H. The priority of rules and the termination analysis using Petri nets. *The International Arab Journal of Information Technology*, 2007, 4(2): 177-183
- [5] Bailey J, Dong G. On the decidability of the termination problem of active database system. *Theoretical Computer Science*, 2004, 311(1-3): 389-437
- [6] Jin Y, Urban S D, Dietrich S W. A concurrent rule scheduling algorithm for active rules. *Data & Knowledge Engineering*, 2007, 60(3): 530-546
- [7] Baralis E, Ceri S, Paraboschi S. Compile-time and run-time analysis of active behaviors. *IEEE Transactions on Knowledge and Data Engineering*, 1998, 10(3): 353-370
- [8] Ray I, Ray I. Detecting termination of active database rules using symbolic model checking//*Proceedings of Advances in Databases and Information Systems*. Vilnius, Lithuania, 2001: 266-279
- [9] Xiong Wei, Liao Wei, Chen Hong-Sheng, Jing Ning. Concurrent behavior control for active rules of spatial database. *Journal of Computer Research and Development*, 2006, 43(8): 1464-1470(in Chinese)
(熊伟, 廖巍, 陈宏盛, 景宁. 空间数据库主动规则并发行为控制研究. *计算机研究与发展*, 2006, 43(8): 1464-1470)
- [10] Baralis E, Widom J. An algebraic approach to static analysis of active database rules. *ACM Transactions on Database System*, 2000, 25(3): 269-332
- [11] Li X, Medina J M, Chapa S V. Applying Petri nets in active database systems. *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, 2007, 37(4): 482-493
- [12] Aiken A, Widom J, Hellerstein J M. Behavior of database production rules: Termination, confluence, and observable determinism//*Proceedings of the SIGMOD'92*. New York, NY, USA, 1992: 59-68
- [13] Lee S Y, Ling T W. Unrolling cycle to decide trigger termination//*Proceedings of the VLDB'99*. San Francisco, CA, USA, 1999: 483-493
- [14] Couchot A. Improving the refined triggering graph method for active rules termination analysis//*Proceedings of the BN-COD 2002*. London, UK, 2002: 114-133
- [15] Zuo Wan-Li, Liu Ju-Hong, Liu Shu-Fen. Relationship graph and termination analysis for active rule set. *Journal of Software*, 2001, 12(2): 276-282(in Chinese)
(左万利, 刘居红, 刘淑芬. 关联图与主动规则集的终止性分析. *软件学报*, 2001, 12(2): 276-282)
- [16] Xiong Z M, Hao Z X. Refined rules termination analysis through activation path//*Proceedings of the 6th International Conference on Advances in Web-Age Information Management*. Hangzhou, China, 2005: 392-403
- [17] Hao Zhong-Xiao, Xiong Zhong-Min. An efficient algorithm about computing an irreducible rule set in active database. *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. *计算机研究与发展*, 2006, 43(2): 281-287)



XIONG Wei, born in 1976, Ph.D., assistant professor. His research interests include geographical information system and spatial database.

WU Ye, born in 1986, Ph.D. candidate. His research interests include spatial database and parallel processing.

ZHANG Zhen, born in 1986, M. S. candidate. His research interests include active database and location based service.

WU Qiu-Yun, born in 1971, associate professor. His research interests include information system and advanced database.

Background

Termination analysis is one of the important problems in the behavior control of active rule sets, and efficient termination analysis can be used to enhance the flexibility of active service. The termination analysis research belongs to the field of active database. With the development of Web service application, recent studies of termination analysis focus on concurrent and dynamic network environment. In these circumstances, thousands of users may trigger the active rule sets, so the most important challenge is how to improve the analysis accuracy, and at the same time keep the efficiency. The researchers paid more attention to the study of run-time and compile-time analysis. However, the existing run-time methods are costly for practical application and little work of compile-time methods consider the dynamic environment where the rule sets are updated frequently.

The conservation of typical previous techniques based on relationship graph is analyzed, and we found two new cases of termination to improve traditional methods. The main contribution of this paper is in two aspects. Firstly, the forbidden trigger ring definition in relationship graph is improved. Triggering path and finite trigger ring are incorporated to improve the accuracy of termination analysis. Secondly, with the two-stage policy, analysis efficiency is ensured. The results show the method can be used to support the active rule

termination analysis in the complicated network, it can provide more accuracy and gain acceptable performance.

This research is supported by the National Natural Science Foundation of China (40801160, 60902036, 61070035), National High Technology Research and Development Program (863 Program) of China (2007AA120400) and Research Fund for the Doctoral Program of Higher Education of China (20104307110017). These researches aim at providing active spatial information service to the customers. The background of these projects is that Web services are now becoming the main technique for large scale application of spatial information. There is a great demand on business rules in traffic schedule, location based service and mobile computing. Therefore, research group of these projects implement active rule in the spatial database core, construct the service chain, and use the active rule set to support active Web service of GIS development and application.

Theories and methods in this paper have been implemented and integrated into Ingres, an open source database management system. With these analysis functions, the database can be used and shared for the application of active spatial information service. In the future work, the complex event and complicated condition will be incorporated into the triggering path methods.