

文章编号:1007-5321(2017)02-0049-08

DOI:10.13190/j.jbupt.2017.02.008

# 事件约束的时间不确定事件流查询处理

李芳芳, 刘红蕾, 于戈

(东北大学 计算机科学与工程学院, 沈阳 110169)

**摘要:** 提出两种事件约束的时间不确定事件流查询处理方法. 基于可能世界的查询处理方法, 根据事件约束的不确定事件查询语义, 通过对可能世界的遍历获取查询结果. 基于事件约束优先的查询处理方法中, 根据事件约束计算其所涉及事件发生时间的联合分布律, 进而获得复合查询所涉及事件的边缘分布律, 最后利用时间不确定事件查询处理技术进行后续的查询处理. 实验证明了 2 种方法对时间不确定约束事件查询有效. 基于事件约束优先查询处理方法的可能世界集规模更小, 比基于可能世界的查询处理方法更高效.

**关键词:** 事件约束; 时间不确定; 复杂事件处理; 事件流

**中图分类号:** TP311

**文献标志码:** A

## Query Processing over Constraint Event Stream with Uncertain Timestamps

LI Fang-fang, LIU Hong-lei, YU Ge

(School of Computer Science and Engineering, Northeastern University, Shenyang 110169, China)

**Abstract:** Two approaches are proposed to process the constraint event stream with uncertain timestamps. The query processing approach based on possible worlds can obtain the results by scanning the possible worlds according to the semantic of the constraint events. The query processing approach based on constraint event priority calculates the joint probability density of event timestamps, furtherly, calculates the marginal probability density of events in the composited query, and finally processes the query according to the query processing technology for the uncertain timestamp events. Experiments verify that both the query processing approaches for constraint event streams with uncertain timestamps are effective, and due to the smaller possible worlds, the query processing approach based on constraint event priority is more efficient than the query processing approach based on possible worlds.

**Key words:** constraint event; uncertain timestamp; complex event processing; event stream

事件是 RFID (radio frequency identification devices) 系统的数据采集和管理形式, 是 CPS (cyber-physical system) 及物联网中物理数据到逻辑语义的重要元素. 随着 RFID、CPS、物联网技术研究的不断深入, 复杂事件处理技术研究也获得了广泛关注<sup>[1-3]</sup>. 目前, 很多应用要求处理带有约束事件的复杂事件流. 约束事件是指事件不是随机产生而是按照预定义的业务逻辑或规则产生, 事件之间存在着

约束关系<sup>[4]</sup>.

很多系统<sup>[5-7]</sup>针对发生时间确定的事件流进行处理, 通常假设事件流中的事件发生时间是精确的, 并且来自不同事件源的事件能够汇集成一个全序或者偏序流. 但是, 数据的漏读和错读、系统采样时间粒度不匹配、分布式系统事件采集时间戳时钟同步等原因, 都可能导致查询处理中事件发生时间未知或不精确. 系统需考虑每个事件发生时间的所有可

收稿日期: 2016-09-12

基金项目: 中央高校基本科研业务专项资金项目 (N161604005); 国家自然科学基金项目 (61472071)

作者简介: 李芳芳 (1977—), 女, 博士, 讲师, E-mail: lifangfang@mail.neu.edu.cn.

能情况,增加了查询复杂度. Zhang 等<sup>[8]</sup>建立了基于时间不确定模型、模式查询语义及处理框架. 有研究将时间戳不确定与空间不确定结合起来在事件流上进行序列模式的匹配<sup>[9]</sup>. 文献[10]提出了基于概率时间数据库模型用来清洗不确定时间事件.

本文研究的核心问题是在时间戳不确定的事件流中查询满足一定约束关系的事件处理,提出了基于可能世界的查询处理方法和基于事件约束优先的查询处理方法,利用实验验证这两个算法效率.

## 1 问题描述

**定义1** 不确定原子事件. 简称为不确定事件, 本文的不确定事件特指发生时间不确定的原子事件. 不确定事件可用元组(EventType, ID, UI: [lower, upper], ( $p: UI \rightarrow [0, 1]$ )?) 表示, 其中 EventType 为事件对应的类型; ID 为事件唯一的识别信息; UI: [lower, upper] 为事件提供者指定的事件发生的可能时间区间, lower 为其左边界, upper 为其右边界;  $p: UI \rightarrow [0, 1]$ ? 为发生时间在区间 UI 上的可选的概率分布函数, 默认的是均匀分布, 此概率分布函数主要由事件源所决定.

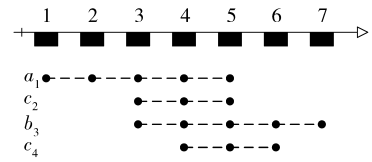
通过 UI 和  $p$  可知, 每个事件的发生时间有若干可能性, 即在连续时间点, 有如下关系式成立,  $\{(t_j, p(t_j)) \mid j = 1, \dots, |UI|, 0 \leq p(t_j) \leq 1, \sum p(t_j) = 1\}$ , 其中,  $p(t_j)$  是某一事件  $e$  发生在时刻  $t_j$  的概率.

**定义2** 概率原子事件. 简称为概率事件, 是由不确定事件按照其所有可能发生的时刻及其概率转化而成的事件. 概率事件可以用元组(ID, EventType,  $T, p$ ) 表示, 其中  $p$  是不确定事件在  $T$  时刻发生的概率.

**定义3** 不确定事件流. 由一系列不确定事件形成的组合, 即  $UncertainS = (e_1, e_2, \dots, e_n)$ , 其中, 下标为事件 ID 号.

**定义4** 概率事件流. 由一系列概率事件按时间顺序形成的组合, 即  $ProbabilityS = (e_1^1, e_2^2, \dots, e_n^n)$ , 其中, 下标为事件 ID 号, 上标为事件发生的可能时刻. 本文概率事件流是由不确定事件流转化而来. 图1展示了不确定事件流转化成概率事件流. 具体来说, 图1(a)是含有4个不确定事件的不确定事件流, 而图1(b)是由图1(a)转化而来的, 是含有16个概率事件的概率事件流.

**定义5** 匹配事件集 MES(matching event set).



(a) 含有4个不确定事件的不确定事件流

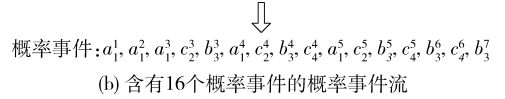


图1 获得概率事件流的过程

在事件流中, 与指定复合事件模式  $Pa = (E_1, E_2, \dots, E_n)$  匹配的事件序列集. 例如,  $MES(seq(E_1, and(E_2, E_3), E_4), S_1)$  指的是在事件流  $S_1$  中与嵌套事件模式(嵌套查询),  $seq(E_1, and(E_2, E_3), E_4)$  相匹配的事件序列集.

**定义6** 特征序列. 特征序列是针对一个概率事件序列定义的, 其定义为一个事件序列中具有的唯一的事件 ID 号序列. 例如, 概率事件序列  $(r_1^1, s_2^2, d_3^3, w_5^4, o_6^5)$  的特征序列为  $(r_1, s_2, d_3, w_5, o_6)$ ; 同理, 概率事件序列  $(r_1^4, s_2^6, d_3^7, w_5^8, o_6^{11})$  的特征序列同样为  $(r_1, s_2, d_3, w_5, o_6)$ .

**定义7** 事件约束(EC, event constraints). 描述前、后事件间的依赖关系的规则, 这种规则用于限制在某一事件发生后发生其它事件的可能性. 这里包括两种 EC, 分别是发生约束(occurrence constraints)和顺序约束(order constraints).

这里使用的事件约束集为 ECs.

$ECs_1: prior(E_i, E_j, he) := \forall he, \neg (E_j[he] < E_i[he])$ , 规定在任意事件序列中,  $E_j$  一定在  $E_i$  之后发生总是有效的. 此约束属于顺序约束.

$ECs_2: exclusive(E_i, E_j, he) := \forall he, E_i[he] \rightarrow \neg E_j[he]$ , 规定在任意事件序列中, 如果  $E_i$  发生,  $E_j$  没有发生总是有效的. 此约束属于发生约束.

$ECs_3: require(E_i, E_j, he) := \forall he, E_i[he] \rightarrow E_j[he]$ , 规定在任意事件序列中, 如果  $E_i$  发生,  $E_j$  一定发生总是有效的. 此约束属于发生约束.

EC 的不确定事件查询  $Q_C$  是指在不确定事件流  $UncertainS$  中, 查询并返回与事件模式  $Pa = (E_1, E_2, \dots, E_n)$  相匹配并满足  $ECs = (ECs_1, ECs_2, \dots, ECs_m)$ , 而且其发生概率于  $\xi$  的事件序列及其概率值. 即  $Q_C(Pa \wedge ECs, UncertainS, \xi) = \{(d_1, d_2, \dots, d_n), p(d_1, d_2, \dots, d_n) \mid (d_1^1, d_2^2, \dots, d_n^n) \in MES(Pa \wedge ECs, ProbabilityS), \sum p(d_1^1, d_2^2, \dots, d_n^n) =$

$p(d_1, d_2, \dots, d_n) > \xi\}$ . 其中,  $(d_1^1, d_2^2, \dots, d_n^n)$  为概率事件序列,  $(d_1, d_2, \dots, d_n)$  为  $(d_1^1, d_2^2, \dots, d_n^n)$  等概率事件序列通过特征序列分组后对应的不确定事件序列(特征序列),  $\sum p(d_1^1, d_2^2, \dots, d_n^n)$  为将对应相同特征序列  $(d_1, d_2, \dots, d_n)$  概率事件序列的概率和。

## 2 基于可能世界优先的约束事件查询处理方法

基于可能世界优先的查询处理方法(possibleF\_QP), 根据 EC 的不确定事件查询语义, 通过对可能世界的遍历, 获取查询结果. 详细过程如下。

首先, 根据目标不确定事件流 UncertainS 获得其概率事件流 ProbabilityS, 然后通过概率事件流 ProbabilityS 获得其可能世界集  $PWD(S)$ , 对每一个可能世界链  $\text{pwd}_i$  进行 EC 检测和模式查询处理, 获取满足 EC 且与查询模式 Pa 相匹配的概率事件序列  $\text{seq}_i$ . 最后计算每个满足条件的概率事件序列  $\text{seq}_i$  发生的概率  $p_i$ , 将概率大于 0 的概率事件序列选择出来, 形成概率事件序列候选结果集(CandResult). 并扫描候选结果集中的所有概率事件序列  $\text{seq}_i$  并按照其特征序列对候选结果集进行分组. 将每组的概率事件序列的概率相加, 若最终得到的概率值大于等于查询阈值  $\xi$ , 则该分组的特征序列为查询结果, 并记录下该序列的概率值; 若最终得到的概率值小于查询阈值  $\xi$ , 则舍弃该分组。

例如, 针对图 2 中的不确定事件流 UncertainS =  $(d_1, a_2, b_3, c_4)$ , 进行复合模式  $\text{Pa} = \text{seq}(A, B, C)$  的查询, 假定相关的 ECs 是  $\text{prior}(D, A, S)$ . 因为  $\text{prior}(D, A, S)$  所涉及的事件集为  $\{A, D\}$ , 复合模式  $\text{Pa} = \text{seq}(A, B, C)$  所涉及的事件集为  $\{A, B, C\}$ , 由此可求得两者的交集为  $\{A\}$ , 满足进行 EC 不确定事件查询的条件. 然后, 采用 possibleF\_QP 对其进行 EC 的不确定事件查询处理。

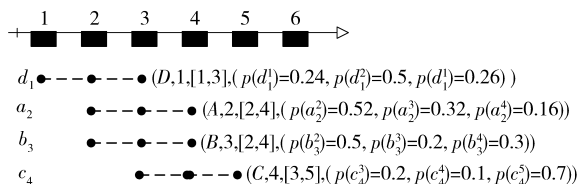


图 2 事件流

首先根据每个不确定事件的时间区间, 进行流扩展, 获得与不确定事件流相对应的概率事件流。

即由含有 4 个不确定事件的不确定事件流转化成含有 12 个概率事件的概率事件流. 转化过程见图 3。

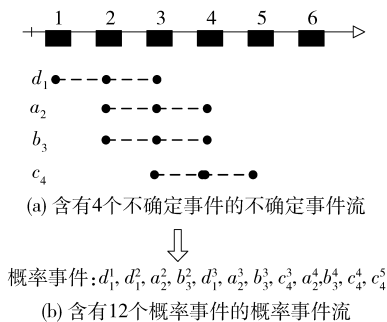


图 3 获得概率事件流的过程

接下来, 通过图 3(b) 的概率事件流获得其对应的  $PWD(S) = \{\text{pwd}_0, \text{pwd}_1, \text{pwd}_2, \dots, \text{pwd}_{79}, \text{pwd}_{80}\}$ . 对于  $\forall \text{pwd}_i \in PWD(S)$ , 进行模式  $\text{Pa} = \text{seq}(A, B, C)$  的匹配处理, 同时, 进行  $\text{prior}(D, A, S)$  的检测处理. 获取满足 EC 且与查询模式 Pa 相匹配的概率事件序列  $\text{seq}_i$ . 最后计算每个满足条件的概率事件序列  $\text{seq}_i$  的发生概率  $p_i$ , 将概率大于 0 的概率事件序列选择出来, 形成概率事件序列候选结果集. 例如, 可能世界  $\text{pwd}_4 = (d_1^1, a_2^2, b_3^3, c_4^4)$  与模式  $\text{Pa} = \text{seq}(A, B, C)$  匹配, 且满足  $\text{prior}(D, A, S)$ , 因此被放入候选结果集中; 可能世界  $\text{pwd}_1 = (d_1^1, a_2^2, b_3^2, c_4^4)$  虽然满足  $\text{prior}(D, A, S)$ , 但是与事件模式  $\text{Pa} = \text{seq}(A, B, C)$  不匹配, 因此不会被放入候选结果集中; 可能世界  $\text{pwd}_{62} = (d_1^3, a_2^2, b_3^4, c_4^5)$  虽然与事件模式  $\text{Pa} = \text{seq}(A, B, C)$  匹配成功, 但是不满足  $\text{prior}(D, A, S)$ , 因此不会被放入候选结果集中; 依次对可能世界进行模式  $\text{Pa} = \text{seq}(A, B, C)$  查询和  $\text{prior}(D, A, S)$  检测处理, 最终获得候选结果集. 由  $PWD(S)$  获得候选结果集的具体过程见图 4。

获得候选结果集之后, 计算其中每个候选结果的概率. 根据与此查询相关的 EC, 计算出相关事件发生时间的联合分布. 此例的相关事件是  $d_1$  和  $a_2$ , 计算 2 个事件发生时间的联合分布律. 为了计算事件  $d_1$  和  $a_2$  发生时间的联合分布律, 首先根据  $\text{prior}(D, A, S)$  和事件  $d_1$  和  $a_2$  发生时间的概率质量函数计算其条件概率. 获得条件概率  $P(a_2, T | d_1, T)$  的过程如下。

$$p(a_2 T = 2 | d_1 T = 1) = 0.52$$

$$p(a_2 T = 3 | d_1 T = 1) = 0.32$$

$$p(a_2 T = 4 | d_1 T = 1) = 0.16$$

$$p(a_2 T = 3 | d_1 T = 2) = \frac{0.32}{0.32 + 0.16} = \frac{2}{3}$$

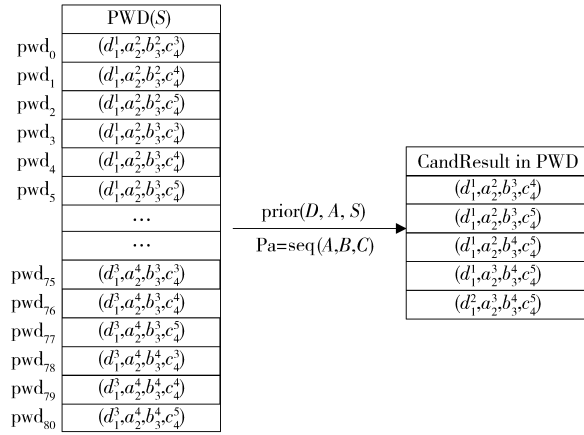


图4 获得候选结果集的过程

$$p(a_2 T=4 | d_1 T=2) = \frac{0.16}{0.32 + 0.16} = \frac{1}{3}$$

$$p(a_2 T=4 | d_1 T=3) = 1$$

$$P(a_2, T=t_i, d_1, T=t_j) = P(a_2, T=t_i | d_1, T=t_j) P(d_1, T=t_j) \quad (1)$$

然后根据联合概率计算的式(1)和事件  $d_1$  和  $a_2$  发生时间的条件概率结果计算其联合概率分布. 获得  $a_2, T$  和  $d_1, T$  的联合分布过程如下.

$$p(a_2, T=2, d_1, T=1) = p(a_2, T=2, d_1, T=1) (p(d_T=1) = 0.52 \times 0.24 = 0.1248$$

$$p(a_2, T=3, d_1, T=1) = p(a_2, T=3, d_1, T=1) (p(d_T=1) = 0.32 \times 0.24 = 0.0788$$

$$p(a_2, T=4, d_1, T=1) = p(a_2, T=4, d_1, T=1) (p(d_T=1) = 0.16 \times 0.24 = 0.0384$$

$$p(a_2, T=3, d_1, T=2) = p(a_2, T=3, d_1, T=2) (p(d_T=2) = \frac{2}{3} \times 0.5 = \frac{1}{3}$$

$$p(a_2, T=4, d_1, T=2) = p(a_2, T=3, d_1, T=2) (p(d_T=2) = \frac{1}{3} \times 0.5 = \frac{1}{6}$$

$$p(a_2, T=4, d_1, T=2) = p(a_2, T=4, d_1, T=3) (p(d_T=3) = 1 \times 0.26 = 0.26$$

由上述计算结果形成事件  $d_1$  和  $a_2$  发生时间的联合分布律见表1. 得到事件  $d_1$  和  $a_2$  发生时间的联合分布律之后, 计算候选结果集中每个候选结果的

概率. 由于事件  $d_1, a_2$  与事件  $b_3, c_4$  之间不存在 EC, 因此如果将事件  $d_1, a_2$  看作一个事件时, 其与事件  $b_3, c_4$  三者之间就是彼此独立的, 则可用式(2)计算其概率.

$$p(d_1, a_2, b_3, c_4) = \prod p(a_2, d_1) p(b_3) p(c_4) \quad (2)$$

表1  $a_2, T$  和  $d_1, T$  的联合分布律

$a_2$	$d_1$		
	1	2	3
2	0.124 8	0	0
3	0.076 8	1/3	0
4	0.038 4	1/6	0.26

最后根据式(2)计算每个候选结果的概率. 例如候选结果  $(d_1^1, a_2^2, b_3^3, c_4^4)$  的概率为  $p(d_1^1, a_2^2, b_3^3, c_4^4) = 0.1248 \times 0.2 \times 0.1 = 0.002496$ , 同理, 候选结果  $(d_1^1, a_2^2, b_3^3, c_4^5)$  的概率为  $p(d_1^1, a_2^2, b_3^3, c_4^5) = 0.1248 \times 0.2 \times 0.7 = 0.017472$ . 最终获得所有候选结果的概率. 具体计算过程详见图5.

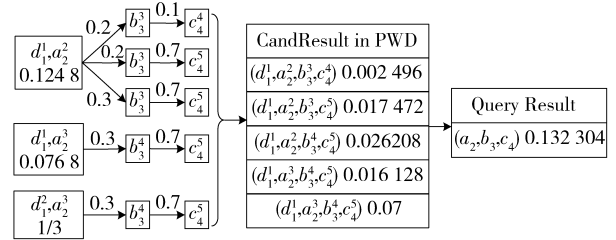


图5 获得查询结果的过程

接下来, 计算模式查询结果的概率值. 计算方法分两步, 首先将候选结果按照其特征序列进行分组, 然后按照式(3)计算每个分组的特征序列的概率值.

$$p(d_1, a_2, b_3, c_4) = \sum p(d_1^1, a_2^2, b_3^3, c_4^4) \quad (3)$$

其中,  $p(d_1^1, a_2^2, b_3^3, c_4^4)$  是一个分组中一个候选结果的概率值. 按照特征序列可以将本例中候选结果集分成一个组, 其特征序列(特征序列中只包含查询模式所涉及的事件)为  $(a_2, b_3, c_4)$ , 其概率为  $p(a_2, b_3, c_4) = p(d_1^1, a_2^2, b_3^3, c_4^4) + p(d_1^1, a_2^2, b_3^3, c_4^5) + p(d_1^1, a_2^2, b_3^4, c_4^5) + p(d_1^1, a_2^3, b_3^4, c_4^5) + p(d_1^1, a_2^3, b_3^4, c_4^5) = 0.132304$ .

最后, 将每个分组的概率值和用户给定的查询阈值进行比较. 该查询阈值由用户根据实际应用的精度要求以及实用经验给出. 如果其概率值大于等于查询阈值, 返回这个分组的特征序列及其概率值.



如果其概率值小于查询阈值,淘汰该分组. 至此,图 2 中的事件流用 possibleF\_QP 完毕.

possibleF\_QP 中,如果 EC 所涉及的事件和复合查询所涉及的事件的交集不为空,需要枚举此查询所涉及事件与 EC 所涉及事件发生时间的所有可能情况,假设 EC 所涉及的事件数为  $M$ ,复合查询所涉及的事件数为  $N$ ,事件的时间戳区间的平均大小为  $U$ ,那么 PWD( $S$ ) 的大小是指数级的,即  $O(U^{N+M})$ . 如果在其上直接进行模式查询处理和 EC 检测,则计算的开销较大,为提高处理效率,需要对 PWD( $S$ ) 进行适当的约减,因此提出基于事件约束优先的查询处理方法.

3 基于事件约束优先的查询处理方法

事件约束优先的查询处理方法 (constraintsF\_QP),即优先考虑复合查询所涉及的 EC. 该方法主要分三步完成. 第 1 步,根据 EC,计算其所涉及事件发生时间的联合分布律;第 2 步,基于事件发生时间的联合分布律,获得复合查询所涉及事件的边缘分布律,因为复合查询所涉及事件集和 EC 所涉及事件集的交集的事件其发生时间的概率分布会有所改变,需要使用其边缘分布律;第 3 步,按照时间不确定事件查询处理技术进行后续的查询处理. 下面以一个例子对 constraintsF\_QP 进行详细的说明.

为了进行对比说明,同样以图 2 中的事件流为例,用 constraintsF\_QP 进行处理. 首先进行第一步,根据 prior( $D, A, S$ ),计算事件  $d_1$  和  $a_2$  发生时间的联合分布律,计算结果见表 1.

$$P\{a_2, T=t_i\} = \sum_j P(a_2, T=t_i, d_1, T=t_j) \quad (4)$$

然后,进行第二步,根据表 1 中事件  $d_1$  和  $a_2$  发

生时间的联合分布律和式 (4),得到事件  $a_2$  发生时间的边缘分布律,其边缘分布律见图 6.

$d_1$				$\Rightarrow$				
$a_2$	1	2	3		$a_2$	2	3	4
2	0.124 8	0	0		$p$	0.124 8	0.076 8+1/3	0.298 4+1/6
3	0.076 8	1/3	0					
4	0.038 4	1/6	0.26					

图 6 事件  $a_2$  发生时间的边缘分布律

复合查询所涉及事件集和 EC 所涉及事件集的交集的事件,本例中即  $a_2$ ,其发生时间的概率质量函数会改变,变为其边缘分布律,然后将使用其边缘分布律作为其概率分布,事件  $a_2$  发生时间的概率质量的改变详见图 7.

最后,进行第三步,按照两种时间不确定事件查询处理技术进行后续的查询处理. 一种是基于点 (point-based) 的评估框架;一种是基于事件 (event-based) 的评估框架. 首先使用第一种基于点的评估框架进行查询处理.

1) 基于点的评估框架,因为经典的复杂事件处理系统处理针对的是发生在确定的时间点的事件,因此在基于点的评估框架中,先将目标的时间不确定事件转化成带有概率的点事件,然后再用经典系统 SASE<sup>[5]</sup> 进行查询匹配处理,最后查询匹配候选结果按照特征序列进行分组,计算各组特征序列的发生概率,形成最终的查询匹配结果.

对于本例,首先对图 8(a) 中的目标事件流进行流扩展,将含有 3 个不确定事件的事件流转化成含有 9 个概率事件的事件流,其转化过程见图 8,转化而成的概率事件流详见图 8(b). 然后针对转化而成的概率事件流,获得 PWD( $S$ ),见图 8(b). 图 9

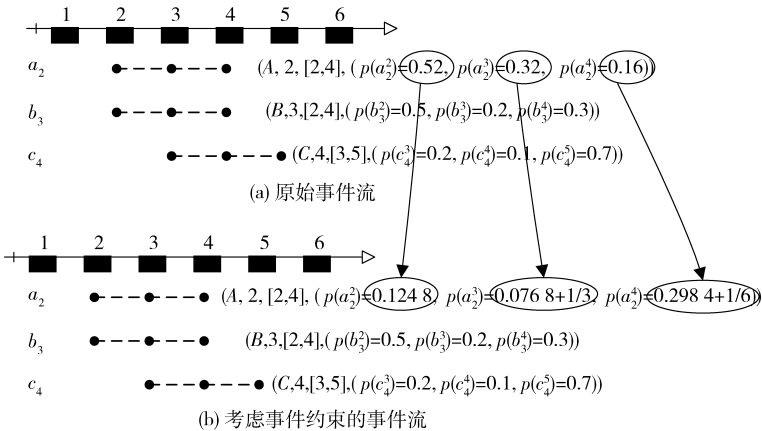


图 7 考虑事件约束的事件流



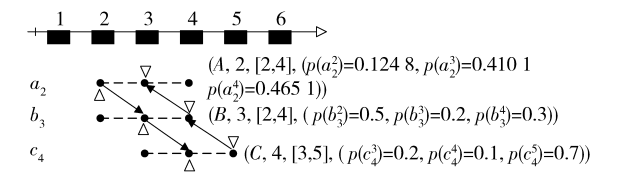


图 12 获得事件发生区间的有效的上下界

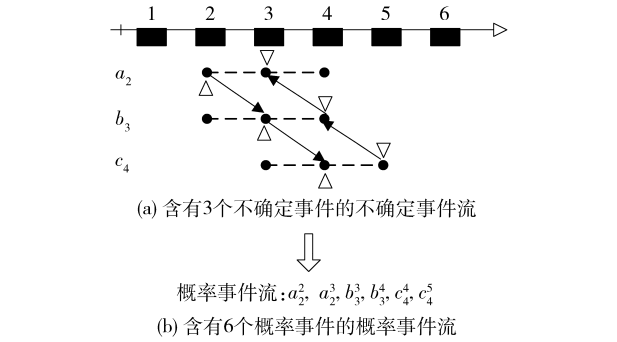


图 13 获得概率事件流的过程

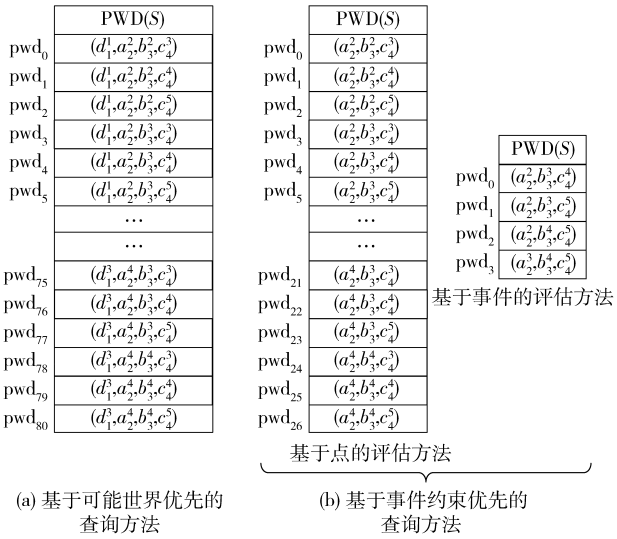


图 14 可能世界集的比较

了处理效率。

## 4 实验结果与分析

实验评估查询长度、事件约束数目、事件数目、时间区间对所提出的基于 possibleF\_QP 和基于 constraintsF\_QP 的处理时间的影响。实验使用 C++ 语言实现,运行环境为 Intel(R) Core(TM) i7 3.07 GHz CPU、8.00 G 内存和运行 64 位 Windows7 操作系统。仿照文献[8]模拟测试的做法,实现了一个事件流产生器,用事件流产生的事件对算法进行验证。

查询长度的影响。在事件约束数为 1,时间区间

为 3,事件数为 200 时,评价查询长度(查询中包括的事件数量)对算法处理时间的影响,如图 15 所示,查询长度的增长对 possibleF\_QP 的影响较大,因为 possibleF\_QP 在获取 PWD(S) 阶段,查询长度的增长使 PWD(S) 迅速增大,事件约束检测和复合查询的处理时间会显著增长。而 constraintsF\_QP 的 PWD(S) 比较小,而且在查询处理的第 3 步,可以使用剪枝的方法进一步缩减可能世界的数目,因此处理时间的变化较 possibleF\_QP 变化不大。

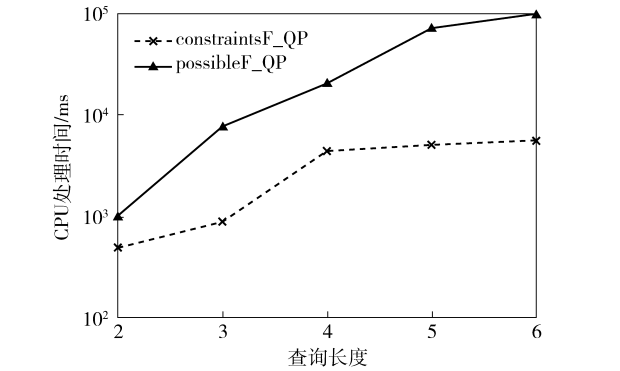


图 15 查询长度对处理时间的影响

事件数的影响。在事件约束为 1,时间区间为 3,查询长度为 3 时,评价事件数算法处理时间的影响。如图 16 所示,事件数的增长,对 possibleF\_QP 影响较大,而对 constraintsF\_QP 影响较小,原因与查询长度影响实验相同。

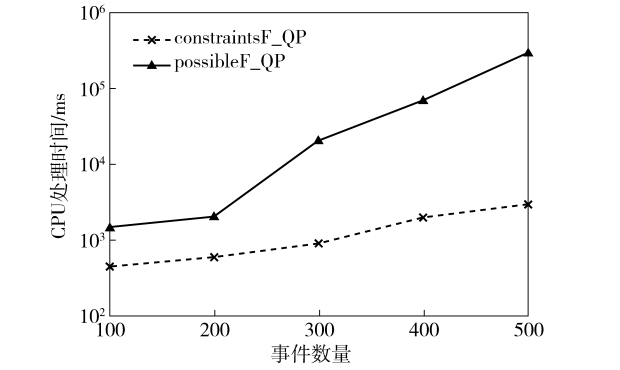


图 16 事件数对处理时间的影响

事件约束数的影响。在时间区间为 3,查询长度为 5,事件数为 200 时,评价 EC 的数目对算法处理时间的影响。如图 17 所示,随着 EC 数的增长,两种方法的处理时间都在增长,因为 EC 数目的增长可以导致概率计算代价增大。EC 数的增长对 possibleF\_QP 影响更大,因为 possibleF\_QP 在获取 PWD(S) 阶段和候选集阶段,由于 EC 数的增长,EC 检测

时间会显著增长. 而 constraintsF\_QP, 只需进行查询处理, 处理时间较 possibleF\_QP 小.

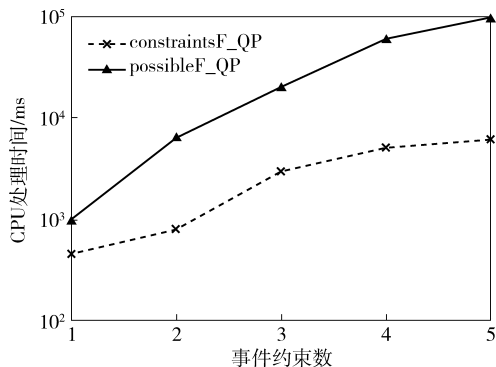


图 17 事件约束数对处理时间的影响

时间区间的影响. 在 EC 为 3 时, 查询长度为 3, 事件数为 300 时, 评价时间区间对算法处理时间的影响. 如图 18 所示, 时间区间的增长, 对 possibleF\_QP 影响较大, constraintsF\_QP 增长的幅度小于 possibleF\_QP. 原因与 EC 数影响实验相同.

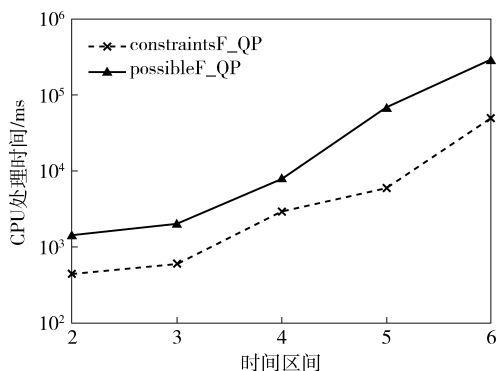


图 18 时间区间对处理时间的影响

## 5 结束语

本文研究了时间不确定模型下, 具有约束的事件流查询问题. 提出了基于可能世界优先的查询处理方法和基于事件约束优先的查询处理方法, 并且实验证明两种方法对于处理事件约束查询都是有效的, 事件约束优先的查询处理方法比基于可能世界

优先的查询处理方法效率更高.

## 参考文献:

- [1] Zhang H, Diao Y, Immerman N. On complexity and optimization of expensive queries in complex event processing [C] // SIGMOD2014. New York: ACM, 2014: 217-228.
- [2] Ray M, Lei C, Rundensteiner E A. Scalable pattern sharing on event streams [C] // SIGMOD 2016. New York: ACM, 2016: 495-510.
- [3] Poppe O, Lei C, Rundensteiner E A, et al. Context-Aware event stream analytics [C] // EDBT 2016. Konstanz: University Library, 2016: 413-424.
- [4] Ding L, Chen S, Rundensteiner E A, et al. Runtime semantic query optimization for event stream processing [C] // ICDE2008. Washington, DC: IEEE Computer Society, 2008: 676-685.
- [5] Gyllstrom D, Wu E, Chae H J, et al. SASE: complex event processing over streams [C] // CIDR2007. New York: ACM, 2007: 407-411.
- [6] Welbourne E, Khossainova N, Letchner J, et al. Cascadia: A system for specifying, detecting, and managing RFID events [C] // Proceedings of the 6th International Conference on Mobile Systems. New York: ACM, 2008: 281-294.
- [7] Wang D, Rundensteiner E A, Iii R T E. Active complex event processing over event streams [J]. Proceedings of the VLDB Endowment, 2011, 4(4): 634-645.
- [8] Zhang H, Diao Y, Immerman N. Recognizing patterns in streams with imprecise timestamps [J]. Information Systems, 2013, 38(3): 244-255.
- [9] Yongluan Zhou, Chunyang Ma, Qingsong Guo, et al. Sequence Pattern Matching over Event Data with Temporal Uncertainty [C] // EDBT2014. Konstanz, Germany: University Library, 2014: 205-216.
- [10] Maximilian Dylla, Iris Miliaraki, Martin Theobald. A temporal-probabilistic database model for information extraction [J]. Proceedings of the Vldb Endowment, 2013, 6(14): 1810-1821.