

文章编号: 1000-5862(2017)06-0645-06

# 基于子结构逻辑的不确定性语义时态查询技术研究

滕少华 涂宏俊 刘冬宁

(广东工业大学计算机学院, 广东 广州 510006)

**摘要:** 为解决不确定性语义时态查询及其计算资源配置的优化问题, 使用子结构逻辑对时态演算机制进行了完善并构建了查询中间件原型. 基于构造性逻辑语义, 构建了中间件演算流程, 将时态属性映射为类型语义, 使针对时间属性的运算从句法演算中剥离, 在保证表达能力的前提下, 仅进行语义计算, 加速并优化了查询演算流程. 实验结果表明: 该方法与中间件原型可行、高效, 具有共性, 为时态信息处理提供理论与技术支撑.

**关键词:** 时态数据库; 时态查询; 不确定性语义; 子结构逻辑

**中图分类号:** TP 301      **文献标志码:** A      **DOI:** 10.16357/j.cnki.issn1000-5862.2017.06.18

## 0 引言

时间是自然界无所不在的客观属性, 随着信息技术的发展, 时态信息处理已成为许多新应用领域的关键技术. 如遥感卫星和地理系统(GIS)、物联网与云计算、生物医药技术、大数据处理和路网物流系统等<sup>[1-5]</sup>. 作为时态信息处理的基石, 时态查询及演算在数据查询、数据挖掘、模式识别和语义网、XML建模等方面深入结合、交叉运用<sup>[6-11]</sup>.

近年来随着云计算、大数据、物联网与移动互联网及相关技术与应用的兴起与普及, 时态信息处理也随之多元化, 作为时态信息处理的基石, 时态演算日趋复杂. 时态演算若想表达更全面的时态知识, 往往需对不确定性语义、模糊语义进行处理, 使得在计算上要付出更大的代价. 但如降低计算复杂性, 采取近似计算, 则造成精度下降, 难以对知识理想表达<sup>[1]</sup>. 表达能力与计算复杂性之间的矛盾, 导致在根据时态数据库进行时态查询与数据共享、数据索引时难于取舍, 不易融合. 不确定性语义时态查询与演算及其计算资源配置的优化问题急需解决.

粗细、不确定性语义计算一直是时态查询的研

究热点. 针对不确定性语义时态查询与演算, 其需要探讨: (i) 能否设计系列具共性的计算模型适用于语义计算; (ii) 该类模型不能过简, 必须具备一定知识表达能力, 能针对各种时间结构进行演算; (iii) 该类模型不能过繁, 计算复杂性与演算流程需合理, 否则耗散大量计算资源与时间.

演算系统的基础在于形式化理论, 而逻辑则是形式化理论的重要基础. 基于时态演算的研究主要采用了传统 Hilbert 风格的逻辑系统, 该系统注重演算对象, 如算子引入和消去, 却忽视了演算的结构, 在规则增删及结构变更方面缺乏灵活性, 表达能力与计算复杂性难以调节, 能效难以优化, 因此已不能满足不确定性语义时态查询及演算的要求.

为此, 本文使用子结构逻辑对时态演算机制进行了完善并构建了查询中间件原型. 该方法与中间件演算流程主要是基于构造性逻辑意义的, 将时态属性映射为类型语义, 使针对时间属性的运算从句法演算中剥离, 在保证表达能力的前提下, 仅进行语义计算, 加速并优化了查询演算流程. 实验结果表明: 该方法与中间件原型可行、高效, 具有共性, 为时态信息处理提供理论与技术支撑.

收稿日期: 2017-05-13

基金项目: 国家自然科学基金(61402118, 61673123), 广东省科技计划(2015B090901016, 2016B010108007), 广东省教育厅项目(粤教高函[2014]97号, 粤教高函[2015]133号)和广州市科技计划(201604020145, 201604046017)资助项目.

作者简介: 滕少华(1962-), 男, 江西南昌人, 教授, 博士, 主要从事网络安全、数据挖掘和协同计算的研究. E-mail: shteng@gdut.edu.cn

## 1 相关工作

### 1.1 国内外现状

当前,针对不确定性语义时态查询及演算的研究主要集中在时间模型与时间粒度的分别研究上. Lu Wei 等<sup>[12]</sup>提出了 DTA 算法( Dominating Tree Algorithm) 对时态数据库的近似实体进行了抽取和分析; 蒋运承等<sup>[13]</sup>使用了描述逻辑对时序 ER 模型进行了建模; 刘冬宁等<sup>[14]</sup>使用了动态逻辑对时态数据库的时间轴进行了建模; A. Luca 等<sup>[15]</sup>基于 Max、Min、Null 和 Point 的 4 种语义粒度对与 Now 语义相关的时态查询进行了研究; E. P. Pons 等<sup>[16]</sup>使用不确定的有效时间与粒度, 结合关系模型对时态数据库进行了建模, 以处理模糊和不确定的时态知识; D. Doder 等<sup>[17]</sup>使用了概率逻辑对时空数据库进行建模, 并对不确定语义进行了分析与处理; S. Slobodan 等<sup>[18]</sup>使用了高阶 Petri 网对模糊时空知识进行了表达与推理; 左亚尧等<sup>[19]</sup>提出基于时态粒度的层次映射转换方法, 克服了弹性时态粒度的影响; R. Emonet 等<sup>[20]</sup>提出了针对多变时间序列的多元模态词与非参数贝叶斯网文本分析模型.

### 1.2 工作基础

在已有文献的基础上, 为对不确定性语义度时态查询与演算及其计算资源配置的优化问题进行处理. 刘冬宁等<sup>[21]</sup>提出基于向量时间集与区间粒度, 针对时态数据库建立了极小子结构逻辑演算系统 TDLmin. 该子结构逻辑系统由原子类型公式  $p, q, r$  通过二元连词 “ $\cdot$ ” 构成, Gentzen 系统为

公理:  $p \Rightarrow p (Id)$ ;

结构规则:

$$\frac{\Delta \Rightarrow A \quad \Gamma \Delta \Rightarrow B}{\Gamma \Delta \Rightarrow B} (Cut),$$

$$\frac{\Gamma \Rightarrow C}{\Gamma \Delta \Rightarrow C} (Weakening),$$

$$\frac{\Gamma \Delta \Rightarrow A}{\Gamma \Delta \Rightarrow A} (Contraction).$$

操作规则:

$$\frac{\Gamma \Delta \Rightarrow B \Rightarrow C}{\Gamma \Delta \Rightarrow C} (\cdot L), \quad \frac{\Gamma \Rightarrow A \quad \Delta \Rightarrow B}{\Gamma \Delta \Rightarrow A \cdot B} (\cdot R).$$

代数语义模型定义如下: 子结构逻辑系统的代数模型为  $G = (M, \cdot, \leq)$ ,  $G$  为交换半群,  $\leq$  为  $M$  上的偏序关系, 并且对所有的  $a, b, c \in M$  满足: (i)  $a \leq a \cdot a$ ; (ii) if  $a \leq c$ , then  $a \cdot b \leq c$ .

在文献[21]中, 已证明上述句法逻辑相对于该代数模型是可靠且完全的, 其中句法逻辑等价于传

统 Armstrong 函数依赖, 句法演算时间复杂度为  $O(n^2)$ , 而语义演算时间复杂度为  $O(n)$ .

由此, 可将代数模型同构映射于时间向量集半群模型.

定义 1 (时间向量集)  $W$  为整数时间轴  $T$  上的一个向量集, 满足:

$$\begin{cases} W = [t_1, \dots, t_i, \dots, t_n] (1 \leq i \leq n), \\ t_{i+1} = t_i + 1, \end{cases}$$

令  $Vec(W)$  为  $W$  上的所有向量子集, 形如:

$$Vec(W) = \{ [a_1, a_2, \dots, a_n] \mid t_1 \leq a_1 \leq a_n \leq t_n \}.$$

定义 2 (时间向量集半群) 代数结构  $S = (Vec(W), \cap, \subseteq)$  是一个交换半群, 其中  $\subseteq$  是  $Vec(W)$  上的偏序关系,  $\cap$  是  $Vec(W)$  上可交换的二元运算.

在此基础上, 本文进一步使用子结构逻辑对时态演算机制进行了完善并构建了查询中间件原型, 对接 13 种时态区间演算的 Allen 方法, 在保证表达能力的前提下, 仅进行语义计算, 加速并优化了查询演算流程.

## 2 系统整体架构

本研究致力于不确定性语义时态查询的共性化及其在表达能力与计算复杂性间的能效优化, 完善不确定性语义时态演算机制, 提高演算精度与效率, 优化演算流程, 以子结构逻辑为工具, 建立不确定性语义时态演算机制, 整合不确定性语义时态查询及其流程, 解决计算资源配置问题, 以此形成系统化的不确定性语义时态查询中间件方法与技术, 为时态信息处理提供理论与技术支撑.

在满足了时态查询需求的同时, 时间复杂度仅为  $O(n)$ , 基于子结构逻辑的不确定性语义时态查询中间件技术的整体框架如图 1 所示, 包含以下模块: (i) 数据库连接驱动模块, 通过 JDBC 方式与非时态数据库管理系统建立连接. (ii) 时态数据库配置模块, 通过属性-类型-时间集映射的方式, 配置时态数据库数据表格属性之间的约束推导关系参数, 完成从时间向量集类型(type)到子结构逻辑的句法映射, 使得传统非时态数据库管理系统中完成配置后的数据表格支持时态查询操作. (iii) 时态 SQL 语句解析模块, 提取具有时态特征的 SQL 语句中时态查询所特有的约束函数. (iv) 不确定性语义的时态关系运算模块, 通过子结构逻辑对时态 SQL 进行运算, 得出时态查询的查询结果.

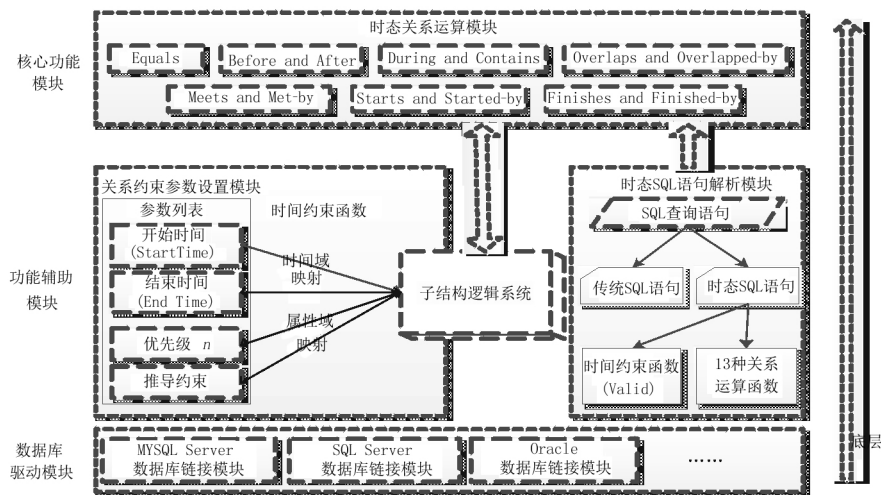


图 1 基于子结构逻辑的不确定性语义时态查询中间件技术的整体框架

上述各模块具体实施方式,采用以下技术方案予以实现。数据库连接驱动模块通过 JDBC 方式与非时态数据库管理系统成功建立连接之后,时态数据库配置模块通过对指定数据库表格进行字段参数配置,完成从类型(type)到子结构逻辑的句法映射,从而配置后的数据库表格可以支持时态查询操作;时态 SQL 语句解析模块通过解析获取的具有时态查询特征的 SQL 语句,提取出时态查询所特有的约束函数,并将处理结果交付给不确定性语义的时态关系运算模块;最后,通过对应于子结构逻辑的时间代数模型对时态 SQL 进行运算,完成带不确定性的三值语义演算,得出时态查询的最终查询结果。

时态数据库配置模块,具体包括以下几个方面:通过 Key-Value 方式,对主键字段的参数配置,对推导出的主键字段(约束主键)的参数配置,对时间约束字段的参数配置;其中,关系约束参数具体包括:(i) 开始时间(StartTime);(ii) 结束时间(EndTime);(iii) 优先级  $n$ ;(iv) 推导约束,以上参数变量值映射入相应的子结构逻辑模型的时间域、属性域。

时态 SQL 语句解析模块,具体包括:采用正则表达式解析、提取具有时态特征的 SQL 语句中时态查询所特有的约束函数,其中时态查询约束函数包括:Valid(时间约束函数)和 Allen 的 13 种时态关系运算,如图 2 所示,分别为:Equals、Before、After、During、Contains、Overlaps、Overlapped-by、Meets、Met-by、Starts、Started-by、Finishes 和 Finished-by。

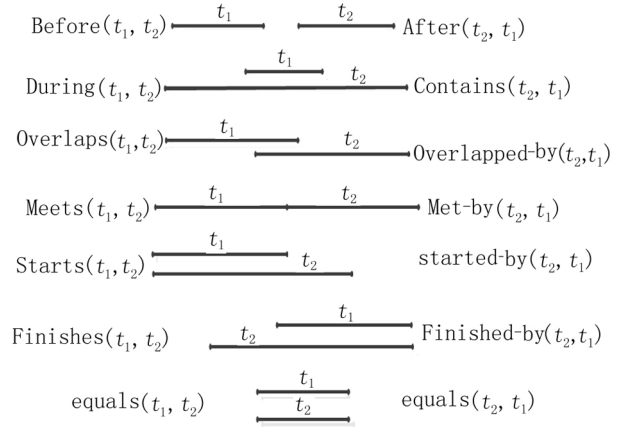


图 2 Allen 方法 13 种时态区间运算

表 1 人员-企业-部门关系表 Per\_Co\_Com

Person_ID	Person_Name	Co_ID	Company	Dep_ID	Department	Time
P001	Ken	C001	IBM	D001	Account	[2017 ,Now]
P002	Jeff	C002	Oracle	D002	Marketing	[2014 ,2014] $\cup$ [2016 ,Now]
P002	Jeff	C003	Google	D001	Account	[2015 ,2015]
P003	Bob	C003	Google	D003	IT	[2015 ,2015]
P003	Bob	C003	Google	D002	Marketing	[2016 ,2016]
P003	Bob	C001	IBM	D001	Account	[2017 ,Now]

其中,Equals: 判断 2 个时间向量集是否等价; Before and After: 判断一个时间向量集是否在另一个向量集之前、之后; During and Contains: 判断一个时间向量集是否包含(被包含于)另一个时间向量集; Overlaps and Overlapped-by: 判断一个时间向量

集与另一个时间向量集是否有交集; Meets and Met-by: 判断一个时间向量集与另一个时间向量集是否始末相接; Starts and Started-by: 判断一个时间向量集是否包含(被包含于)另一个时间向量集,并且 2 个时间向量集的起始时间点相同; Finishes and Fin-

ished-by: 判断一个时间向量集是否包含( 被包含于) 另一个时间向量集, 并且 2 个时间向量集的结束时间点相同.

不确定性语义的时态关系运算模块, 具体包括: 通过子结构逻辑系统对时态约束函数进行关系运算, 将有效时间( Valid) 的表示和运算增加至子结构逻辑模型中, 并且在句法逻辑中抽离对时间属性的表示, 从而可以对时态知识进行准确的表达, 并能处理不确定性三值语义( True、False 和 Uncertain), 由于时态关系运算操作均为线性时间运算比较, 即对连续时间或非连续时间段进行线性比较, 其时间复杂度仅为  $O(n)$ .

### 3 时态查询运算流程与实验分析

在中间件原型验证实验中, 本文分别对 Allen 方法的 13 种时态关系区间运算在 4 张数据库表格上进行 1 000 次实验, 对实际查询时间取平均值, 其中 4 张数据库表格的基本表结构为表 1 所示. 表 1 是按该文中提出的方法所实现的原型系统中的真实数据库测试表格结构, 表 1 由 7 列属性构成, 为了体现实验结果的说服力, 测试表格 1 中的数据内容由无规律的自然数据组成, 并按人员 ID( Person\_ID) 升序排序, 由于篇幅原因, 表 1 仅展示少许记录数据. 实验所采用的 4 张数据库表格结构如表 1 所示, 仅在记录数目上有所不同, 其中这 4 张数据库表格中的记录数目分别为  $10^4$ 、 $10^5$ 、 $10^6$  和  $10^7$ , 具体实验结果统计如表 2 所示.

因为在时态数据库应用中, 对于时间类型( type) Time 实际可表示为一系列有序的时间顺序集合, 所以完成时态查询操作的时间复杂度仅为  $O(n)$ , 如图 3 所示. 其时间曲线随实验次数增加无曲率线性增长, 说明时间复杂度未出现高次变元参数( 如  $n^2$  等).

以下对其演算流程进行详述, 分为句法分析与查询语义演算两部分.

表 2 13 种时态关系区间运算时间统计数据

关系区间	平均时态查询时间 / ms			
	$10^4$	$10^5$	$10^6$	$10^7$
Equals	160	174	252	1 054
Before/After	45	51	87	451
During/Contains	43	47	92	464
Overlaps/Overlapped-by	50	59	102	470
Meets/Met-by	44	50	80	443
Starts/Started-by	44	50	97	455
Finishes/Finished-by	41	52	79	449

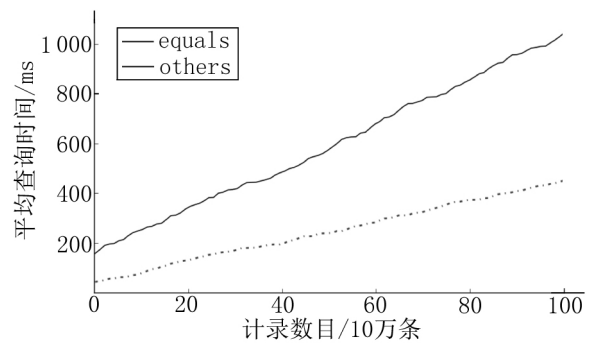


图 3 时态查询操作时间效率统计

#### 3.1 时态查询句法分析

在中间件句法分析中, 主要根据文献 [22] 中定义的  $L_{CTQ}$  范畴语法对具体的结构化时态查询语句进行演算处理, 其中  $L_{CTQ}$  范畴集的定义如下.

定义 3 ( $L_{CTQ}$  范畴集) 集合  $B$  表示基本范畴集合,  $t$  表示时间类型,  $\mu$  表示表属性类型,  $s$  表示终止符类型,  $f$  表示数据库表类型.  $CAT(B)$  为一最小集, 使得

- (i)  $B \subseteq CAT(B)$ ;
- (ii) 若  $A, B \in CAT(B)$ , 则  $A \leftarrow B \in CAT(B)$ ;
- (iii) 若  $A, B \in CAT(B)$ , 则  $A \rightarrow B \in CAT(B)$ ;
- (iv) 若  $A, B \in CAT(B)$ , 则  $A \cdot B \in CAT(B)$ ;
- (v) 若  $A, B \in CAT(B)$ , 则  $A \mid B \in CAT(B)$ .

具体的时态查询 SQL 语句示例, 如下所示( 以表 1 为例):

```
System
select r1. Person_ID
from Co_ID r1, Co_ID r2
where r1. Person_ID = r2. Person_ID
valid( r1. Person_ID, r1. Co_ID) equals valid( r2.
Person_ID).
```

该查询语句的具体查询需求为: 查询“只在唯一企业并且只在唯一部门中的职工”. 其中 Person\_ID、Co\_ID 分别为数据表主关键字( pk) 职工编号、公司编号.

根据  $L_{CTQ}$  范畴, 对上述具体的时态查询 SQL 语句建立查询词汇表, 如表 3 所示.

表 3 范畴语法词汇表

词项	类型	简要描述
select	$(n \rightarrow s) \leftarrow a$	动作类型
r1. Person_ID	$a$	属性类型
from	$(s \rightarrow s) \leftarrow f$	介词类型
Co_ID r1	$f$	表类型
Co_ID r2	$f$	表类型
where	$(s \rightarrow s) \leftarrow s$	连词类型
valid( )	$a \leftarrow t / t \rightarrow a$	时态属性类型
equals	$t \mid (a \rightarrow (s \leftarrow a)) \mid t$	时态运算类型
,	$f \rightarrow (f \leftarrow f)$	连词类型
=	$a \rightarrow (s \leftarrow a)$	运算类型
System	$n$	系统公设类型

上述具体各部分时态类型的演算步骤具体如下:

- (i) “System”: 系统公设, 类型为  $n$ ;
- (ii) “select  $r1, Person\_ID$ ”类型运算为  $((n \rightarrow s) \leftarrow a) \cdot a \vdash n \rightarrow s$ ;
- (iii) “from  $Co\_ID\ r1, Co\_ID\ r2$ ”类型运算为  $((s \rightarrow s) \leftarrow f) \cdot ((f \leftarrow f) \cdot f) \cdot (f \rightarrow (f \leftarrow f)) \cdot ((f \leftarrow f) \cdot f) \vdash s \rightarrow s$ ;
- (iv) “where  $r1, Person\_ID = r2, Person\_ID$ ”类型运算为  $((s \rightarrow s) \leftarrow s) \cdot a \cdot (a \rightarrow (s \leftarrow a)) \cdot a \vdash s \rightarrow s$ ;
- (v) “valid( $r1, Person\_ID\ r1, Co\_ID$ ) equals valid( $r2, Person\_ID$ )”类型运算为  $((s \rightarrow s) \leftarrow s) \cdot (a \leftarrow t) \cdot (t \mid (a \rightarrow (s \leftarrow a)) \mid t) \cdot (t \rightarrow a) \vdash s \rightarrow s$ ;
- (vi) 整句查询语句的最终类型运算结果为  $n \cdot (n \rightarrow s) \cdot (s \rightarrow s) \cdot (s \rightarrow s) \cdot (s \rightarrow s) \vdash s$ .

根据上述的各部分时态查询演算的结果最终能够推导出终止符类型  $s$ , 能够使得句法分析演算追中停止并得到最后的正确结果. 而对于不符合该句法结构的时态查询语句, 则无法推导出终止符类型  $s$ .

### 3.2 时态查询演算流程及结果分析

时态查询逻辑流程的具体实施步骤如下: (i) 接收普通(非时态)/时态 Sql 查询需求; (ii) 时态解析模块对 sql 进行处理判断, 若为时态查询需求, 执行 (iv), 否则执行 (iii); (iii) 执行传统 SQL 查询; (iv) 执行时态查询, 通过子结构逻辑语义模型对时态知识进行二次计算, 完成从类型(type)到子结构逻辑的向量集半群的映射  $(\mu: T_p \rightarrow Vec(W))$ , 对时间进行运算和比较; (v) 查询值判真: True、False 或 Uncertain.

表1为人员-企业-部门关系表 Per\_Co\_Com, 以此表为例, 对基于子结构逻辑的不确定性语义时态查询中间件进行具体时态查询实例的逻辑处理流程及结果分析.

1) 关系约束模块完成表1中从类型(type)到子结构逻辑的句法映射, 即依次将主关键字(primary key)  $Person\_ID$ 、 $Co\_ID$ 、 $Dep\_ID$  向优先级 1、2 及推导参数赋值, 完成  $Person\_ID \wedge Co\_ID \Rightarrow Dep\_ID$  值向时间域的映射; 对于类型 Time, 将拆分对应为数据库视图 View 中的 2 个属性域 StartTime 和 EndTime, 即将一个时间向量拆分为 2 个时间点.

2) 对于查询请求“只在唯一企业并且只在唯一部门中的职工”, 对应具体的时态查询语句为:

```
select distinct Person_ID
```

```
from per_co_dep
```

```
where
```

```
valid(Person_ID, Co_ID) equals valid(Person_ID)
```

其中 valid 为有效时间函数, equals 为时间区间约束函数. 该查询需求本质上是判断  $Person\_ID \wedge Co\_ID \Rightarrow TDep\_ID$  是否等于 TimePerson\_ID, 表1只有 Ken 满足条件为最终查询结果.

3) 对于时态查询中的不确定性问题, 例如查询“Jeff 在 2016 年至 2018 年期间是否在 Oracle 公司工作?”对应具体的时态查询语句为:

```
select *
```

```
from per_co_dep
```

```
where
```

```
valid(Person_ID = "P002", Co_ID = "C002")
```

```
contains
```

```
valid("2016", "2018")
```

由于  $(Person\_ID = "P002" \wedge Co\_ID = "C002" = \{ [2014, 2014] \cup [2016, Now] \}, Time = \{ [2016, Now] \cup [Now, 2018] \})$ , 其中  $[Now, 2018] \notin Person\_ID = "P002" \wedge Co\_ID = "C002"$ , 因此最终查询结果为 Uncertain.

## 4 结语

在本文中, 为解决不确定性语义下的时态查询及其计算资源配置的优化问题, 使用子结构逻辑对时态演算机制进行了完善并构建了查询中间件原型. 相关方法与中间件演算流程主要是基于构造性逻辑语义的, 将时态属性映射为类型语义, 使针对时间属性的运算从句法演算中剥离, 在保证表达能力的前提下, 仅进行语义计算, 加速并优化了查询演算流程. 实验结果表明, 该时态查询方法是线性且无曲率的, 演算方法与查询中间件原型可行、高效, 具有共性, 为时态信息处理提供理论与技术支撑.

在后续研究中, 将对演算机制的进一步优化和调整, 分析对非时态查询与各类时态查询的甄别性及分步操作, 避免提前或过早地介入时态演算而牺牲计算复杂性之外的流程时间与运算资源, 优化演算流程.

## 5 参考文献

- [1] 汤娜, 叶小平, 汤庸, 等. 偏序时态 XML 索引 TempPartialIndex? [J]. 软件学报, 2016, 27(9): 2290-2302.

- [2] 陈海明, 崔莉. 面向服务的物联网软件体系结构设计 with 模型检测 [J]. 计算机学报, 2016, 39(5): 853-871.
- [3] Álvarez M R, Félix P, Carriñena P. Discovering metric temporal constraint networks on temporal databases [J]. Artificial Intelligence in Medicine, 2013, 58(3): 139-154.
- [4] Combi C, Oliboni B, Quintarelli E. Modeling temporal dimensions of semistructured data [J]. Journal of Intelligent Information Systems, 2012, 38(3): 601-644.
- [5] 尹浩, 乔波. 大数据驱动的网络信息平面 [J]. 计算机学报, 2016(1): 126-139.
- [6] Hans-Ulrich Krieger. A temporal extension of the Hayes and ter horst entailment rules for RDFS and OWL [EB/OL]. [2016-11-18]. <http://www.nifti.eu/publications/pdfs/hayesterhorstextension.pdf>.
- [7] Zhao Ran, Do Quanxuan, Roth Dan. A robust shallow temporal reasoning system [EB/OL]. [2016-11-18]. <http://www.anthology.aclweb.org/N/N12/N12-3008.pdf>.
- [8] Huang Yaping, Zhao Jiali, Liu Yunhui et al. Nonlinear dimensionality reduction using a temporal coherence principle [J]. Information Sciences, 2011, 181(16): 3284-3307.
- [9] 严丽, 马宗民, 刘健, 等. 基于关系数据库映射的模糊数据 XML 建模 [J]. 计算机学报, 2011, 34(2): 291-303.
- [10] 张翀, 唐九阳, 肖卫东, 等. 基于簇核心的 XML 结构聚类方法 [J]. 计算机研究与发展, 2011, 48(11): 2161-2176.
- [11] 杜方, 陈跃国, 杜小勇. RDF 数据查询处理技术综述 [J]. 软件学报, 2013, 24(6): 1222-1242.
- [12] Lu Wei, Gabriel Pui Cheong Fung, Du Xiaoyong et al. Approximate entity extraction in temporal databases [J]. World Wide Web, 2011, 14(2): 157-186.
- [13] 蒋运承, 汤庸, 王驹, 等. 基于描述逻辑的带属性依赖时序 ER 模型 [J]. 计算机研究与发展, 2007, 44(10): 1765-1773.
- [14] 刘冬宁, 汤庸. 时态数据库时间轴的动态逻辑模型 [J]. 软件学报, 2010, 21(4): 694-701.
- [15] Luca Anselma, Bela Stantic, Paolo Terenziani, et al. Querying now-relative data [J]. Journal of Intelligent Information Systems, 2013, 41(2): 285-311.
- [16] Jose Enrique Pons, Nicolás Marín, Olga Pons, et al. A relational model for the possibilistic valid-time approach [J]. International Journal of Computational Intelligence Systems, 2012, 5(6): 1068-1088.
- [17] Doder D, Grant J, Ognjanovic Z. Probabilistic logics for objects located in space and time [J]. Journal of Logic and Computation, 2013, 23(3): 487-515.
- [18] Ribaric S, Hrkac T. A model of fuzzy spatio-temporal knowledge representation and reasoning based on high-level Petri nets [J]. Information Systems, 2012, 37(3): 238-256.
- [19] 左亚尧, 汤庸, 舒忠梅. 基于粒度层次映射转换的时态粒点差运算方法 [J]. 计算机研究与发展, 2012, 49(11): 2320-2327.
- [20] Emonet R, Varadarajan J, Odobez J M. Temporal analysis of motif mixtures using dirichlet processes [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2014, 36(1): 140-156.
- [21] 刘冬宁, 汤庸, 滕少华, 等. 基于时态数据库的极小子结构逻辑系统 [J]. 计算机学报, 2013, 36(8): 1592-1601.
- [22] 刘冬宁, 汤庸, 黄昌勤, 等. 时态查询语言的并发 Lambek 演算及范畴语法 [J]. 智能系统学报, 2009, 4(3): 245-250.

## The Temporal Querying of Uncertain Semantics via Substructural Logic

TENG Shaohua, TU Hongjun, LIU Dongning

(School of Computer Science and Technology, Guangdong University of Technology, Guangzhou Guangdong 510006, China)

**Abstract:** Temporal querying of uncertainty semantics is an important and open problem. Its optimization in computing resource allocation is critical with respect to temporal data processing. A temporal calculating system via substructural logic has been put forward. It can deal with uncertain semantics thoroughly. Based on these, a querying middleware prototype has been constructed. It maps time attributes to types directly. Therefore, operations of time attributes can be divorced from syntax calculation, but still ensure knowledge represented. That is lead to the querying concisely and efficiently. Experimental results show that the method and middleware prototype are feasible and valid. And it contributes to provide system method to support temporal database, especially to temporal querying of uncertain semantics.

**Key words:** temporal database; temporal querying; uncertainty semantics; substructural logic

(责任编辑: 冉小晓)