

基于时态数据库的极小子结构逻辑系统

刘冬宁^{1),2)} 汤 庸³⁾ 滕少华¹⁾ 林 哲²⁾

¹⁾(广东工业大学计算机学院 广州 510006)

²⁾(中山大学逻辑与认知研究所 广州 510275)

³⁾(华南师范大学计算机学院 广州 510631)

摘 要 逻辑是知识表达的重要方法,但由于时序性知识与时间属性知识交叉应用的复杂性,其对时态数据库支撑一直不尽如人意.目前时态数据运算体系不完备,时态关系演算缺乏系统和有力的逻辑与代数理论支持.为此,文中从子结构逻辑出发,针对时态数据库及其信息处理中关键的知识推理、时态操作与函数依赖等内容,构建了一个极小的(最小的)子结构逻辑系统 TDL_{min} ,其恰好等价于传统的数据库函数依赖 Armstrong 规则. TDL_{min} 系统能在逻辑语义模型中对时序性、时间属性进行表达,而在句法逻辑系统将时间剥离,从而既表达了时态知识,对时态操作进行处理,又降低了逻辑系统的复杂度,使得逻辑系统的时间复杂度为 $P\text{-time}(O(n^2))$. 而该逻辑系统还可与传统的 Allen 方法进行对接,使得相关时态查询所需的时间代价为传统非时态查询的时间加上一个复杂度仅为 $O(n)$ 的线性时间,从而使得系统具有更强的普适性和应用前景.

关键词 时态数据库;子结构逻辑;时态查询;函数依赖

中图法分类号 TP301 **DOI号** 10.3724/SP.J.1016.2013.01592

A Minimal Substructural Logic in Temporal Database

LIU Dong-Ning^{1),2)} TANG Yong³⁾ TENG Shao-Hua¹⁾ LIN Zhe²⁾

¹⁾(Faculty of Computer, Guangdong University of Technology, Guangzhou 510006)

²⁾(Institute of Logic and Cognition, Sun Yat-Sen University, Guangzhou 510275)

³⁾(School of Computer, South China Normal University, Guangzhou 510631)

Abstract At present temporal data arithmetic system is imperfect, moreover, it is short of powerful supported by logic and algebra for the relational calculation of temporalities. Based these, we put forward a minimum substructural logic (TDL_{min}), which is aimed at the critical area of research fields of temporality inference, temporality operating, temporal relational algebra and functional dependency of temporal database and its information processing. As substructural logic and its universal algebra model are logical tools, temporal knowledge is represented only in the semantics model of TDL_{min} , and the syntactic logic has been deprived in presentation of temporality, which is cut down the complexity of logic system so that TDL_{min} is P-time and its time complexity is only $O(n^2)$. Moreover the syntactic logic of TDL_{min} is properly equal to the Armstrong Rules of functional dependency of traditional database, while the semantics model can build up a method corresponding to Allen's Method of traditional temporal query, which led to the time cost of temporal query is equal to traditional query without temporality plusing a linear time whose complexity is only $O(n)$.

Keywords temporal database; substructural logic; temporal query; functional dependency

收稿日期:2012-03-19;最终修改稿收到日期:2012-10-10. 本课题得到国家自然科学基金(61272067,60970044,61104156)、广东省自然科学基金(10451009001004804)、广东省自然基金团队研究项目(S2012030006242)、广东省战略新兴产业项目(2011168005,2011A010801007)以及教育部重点实验室基金(110411)资助. 刘冬宁,男,1979年生,博士,副教授,中国计算机学会(CCF)会员,主要研究方向为人工智能逻辑、时态信息处理. E-mail: liudn@aliyun.com. 汤 庸,男,1964年生,博士,教授,主要研究领域为时态信息处理. 滕少华,男,1962年生,博士,教授,主要研究领域为网络安全、数据挖掘、协同计算. 林 哲,男,1983年生,博士研究生,主要研究方向为子结构逻辑、形式语言.

1 引言

时态信息的需求与技术一直伴随着数据库技术的发展而产生和发展。时态信息技术仍处于研究和阶段, 现有时态数据库(TDB)共性概念、研究方法仍存在不足, 主要是因为时态数据运算体系不完备、时态关系演算缺乏系统和有力的数学理论支持而引起^[1-2]。

逻辑是知识表达的一种重要方法, 然而逻辑学对时态数据库与时态信息处理方面的支撑却一直不如人意。1994 年 Gabbay 等人论证了公理化时态逻辑及其证明论方法不适合信息系统中的时态数据库建模^[3], 至此信息系统若采用时态逻辑作为时态知识的表达方法, 一般采用的是非公理化时态逻辑^[4], 但其符号演算和推理能力强, 信息处理能力弱, 与时态数据库和时态信息处理研究相脱离, 没有涉及时态知识的数据库模型。此外, 除了非公理化时态逻辑外, 也有许多一阶逻辑、区间逻辑、描述逻辑和动态逻辑等方面针对时态知识的数据库模型及相关数据操纵的逻辑学研究与应用^[5-8]。但这些研究主要基于不同的研究内容, 研究背景与应用也不尽相同, 没有形成统一的观点、模型与方法论, 使得针对时态数据库、时态信息处理与时态数据运算体系的逻辑学研究在系统化方面仍然是一个开问题^[9]。

这是由于时态知识的复杂性造成的, 时态知识一般分为时序性知识和时间属性知识。其中, 时序性知识表示了知识表达、推理和运算过程中的序性关系^[10], 这类时态知识与时态推理中所应遵循的外部秩序有关。时间属性知识则表示了在推理与计算中, 时间作为内因的表现形式。时序性知识与时间属性知识的交叉出现, 导致了时态知识的复杂性, 使得在时态数据库方面, 难以使用统一的逻辑工具作为表述, 这也是由于传统逻辑方法止步于 Hilbert 风格的研究, 忽略了知识内部结构, 及表达与推理的结构特性与资源敏感特性, 由此影响了系统的判定性和计算复杂性而造成的。而不可判定或计算复杂度高、代价大的知识表达方法不能被用于信息系统。

子结构逻辑是一类构造性逻辑, 并将逻辑系统分为结构规则和操作规则。其 Gentzen 式演算能把逻辑系统“掰”碎了, 通过对模态规则、结构规则与逻辑算子(类似时序性知识)的灵活增删, 并借助代数语义模型与类型论做类型映射(类似时间属性知识), 在表达能力、判定性和计算复杂性方面对逻辑

系统进行综合考量^[11], 子结构逻辑在语义上的近似性能进一步转化为关系代数及相关语义, 从而有利于构建时态数据运算体系和数据库建模。

为此, 本文从子结构逻辑出发, 面向时态数据库及其信息处理中关键的知识推理、时态操作与函数依赖等内容, 构建了一个极小的(最小的)子结构逻辑系统(我们将其命名为 $TDL_{\min}$, 意为 Minimum Substructural Logic of Temporal Database)。“极小”主要在于此系统只存在一个类似于传统合取连接词“ $\wedge$ ”的连接词“ $\cdot$ ”和最少的结构规则(2 条), 并恰好等价于传统的数据库函数依赖 Armstrong 规则。能在逻辑语义模型中对时序性、时间属性进行表达, 而在句法逻辑将时间剥离, 从而既表达了时态知识, 对时态操作进行处理, 又降低了逻辑系统的复杂度。该逻辑系统还可与传统的 Allen 方法进行对接, 使得相关时态查询所需的时间代价为传统非时态查询的时间加上一个复杂度仅为 $O(n)$ 的线性时间。

2 逻辑系统及其语义模型

2.1 问题提出与分析

时态数据库中的逻辑研究主要集中在知识表达、时态查询和范式理论 3 个方面, 其中知识表达是基础, 又可进一步分为时态完整性约束和知识推理; 范式理论是时态数据库设计、抽象数据库模型的重要环节, 逻辑学在此起到的作用主要体现在函数依赖和概念设计两方面; 而时态查询问题是核心所在, 是当前时态数据运算体系不完备, 时态关系演算缺乏系统和有力的数学理论支持的具体体现, 其研究可进一步细分为查询操作和时态关系代数。

针对时态查询的逻辑表达方法, 我们举例说明。表 1 中查询“仅在一个企业并且仅在一个部门工作过的员工”, 如使用传统非公理化时态逻辑的表达方法, 可将其表示为时间快照(Snapshot)和时间戳(Timestamp)两种形式, 如下:

(1) Snapshot: $\Diamond \Diamond (\exists c, d. Workplace(x, c, d)) \wedge \neg \exists c', d'. (\Diamond \Diamond Workplace(x, c', d') \wedge c \neq c' \wedge d \neq d')$ (其中 c 表示 company, d 表示 department)。

(2) Timestamp: $\exists c, d, t. Workplace(t, x, c, d) \wedge \neg \exists c', d', t'. (Workplace(t', x, c', d') \wedge c \neq c' \wedge d \neq d')$ (其中 c 表示 company, d 表示 department)。

这两种方法是基于时间点的, 形式晦涩, 缺乏直观性, 而时态逻辑又具有较高的计算复杂性, 通常都

表 1 人员工作关系表 Per_Co

Person_ID	Person_Name	Co_ID	Company	Department	Time
A001	Lee	C001	IBM	Account	[2009, Now]
A002	Jack	C002	Oracle	Marketing	[2009,2009]∪[2011,Now]
A002	Jack	C003	Notel	Account	[2010, 2010]
A003	Tom	C003	Notel	IT	[2009, 2009]
A003	Tom	C003	Notel	Marketing	[2010,2010]
A003	Tom	C001	IBM	Account	[2011, Now]

是 PSTIME 或不可判定的^{[12]①}, 因此不适合对时态查询进行知识表达与推理.

传统非经典时态逻辑之所以形式晦涩、计算复杂性高, 主要原因是时态关系、时态算子以及时态规则引入句法逻辑, 使得逻辑系统在句法层面变得复杂, 尽管知识表达能力强, 但相应也使得系统更为复杂, 并影响了判定性和计算复杂性.

针对时态数据库, 传统逻辑方法主要在句法层面增加对时间的表示, 而语义模型则只简要地以相应的整数、自然数、实数和有理数等序关系, 构成时间轴语义模型与逻辑系统对应, 如 $(Z, \leq)$, $(N, \leq)$, $(R, \leq)$ 和 $(Q, \leq)$. 如能在语义模型中, 增加对时间的表示与运算, 将句法逻辑对时间的表示转移至语义模型, 甚至直接剥离对时间的表示, 则能降低句法逻辑系统的复杂度, 从而在判定性和计算复杂性方面得到更好的性质, 同时通过语义模型对时态知识的表达, 也保持了系统的知识表达能力.

2.2 代数语义模型

表 1 中, 人员表(Person)和企业表(Company)两个实体存在联系 Per_Co, 如图 1 所示, 其主键为 Person 表的 Person_ID 和企业表的 Co_ID, 时间(Time)只是属性(时态数据库中一般不存在时间实体). 但是有别于一般属性, 时间在此是具有约束性的. 例如: $\text{Person_ID} \wedge \text{Co_ID} \Rightarrow_T \text{Department}$, 表示在时间 T 的约束下, 通过 Person_ID 和 Co_ID 能确定人员工作部门, 这也是时态函数依赖 TFD 的一般表示方法^[13-14].

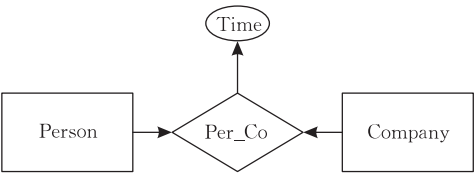


图 1 Person-Company-Workplace 关系的 ER 图

$\text{Person_ID} \wedge \text{Co_ID} \Rightarrow_T \text{Department}$ 的实质是将 $\text{Person_ID} \wedge \text{Co_ID} \Rightarrow \text{Department}$ 的值映射入相应的时间模型或者时间函数、时间域, 也即对时间的表示由时间模型来完成. 换言之如果时间模型有足够的时态知识表达能力, 那么可用其作为逻辑系统

的语义模型, 以减少句法逻辑对时间的表示, 从而降低句法逻辑的复杂性.

逻辑系统的语义模型种类有许多, 例如 Kripke 语义模型、领域语义模型和代数语义模型, 其中抽象程度最高的是代数语义模型. 针对时态数据库的一般应用(如上例), 可对其构建时间代数模型如下.

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 $\text{Vec}(W)$ 为 W 上的所有向量子集, 形如 $\text{Vec}(W) = \{[a_1, \dots, a_n] | t_1 \leq a_1 \leq a_n \leq t_n\}$.

显然整数时间轴 T 本身即为这样的一个向量集, 空集 $\emptyset$ 为这样的一个最小向量集, 且 $W \subseteq T$. 而 $W \in \text{Vec}(W)$ 且 $\emptyset \in \text{Vec}(W)$, 分别为 $\text{Vec}(W)$ 上的极大元和极小元.

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

通过定义 2, 我们实际得到了一个时间模型, 它以时间向量的向量幂集为论域, 同时满足偏序的时序关系, 符合最基本的时态数据库的时序关系, 该模型有且只有一个二元运算, 即“ $\cap$ ”. 这个运算与逻辑运算中的“ $\wedge$ ”合取运算或“ $\cdot$ ”product 运算有密切关系(在切割定理消除部分, 我们会对其进行介绍), 通过这个时间向量集半群以及“ $\cap$ ”运算, 我们可得到一个子结构逻辑系统 $\text{TDL}_{\min}$.

2.3 子结构逻辑句法系统

$\text{TDL}_{\min}$ 系统的公式由原子公式(类型 Tp) $p, q, r \dots$ 通过二元连接词“ $\cdot$ ”构成, 逻辑演算矢列形如 $\Gamma \Rightarrow A$, Gentzen 系统如下所示.

公理: $p \Rightarrow p$ (Id).

结构规则:

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

① Mark K, Gert S. A finite axiomatization of propositional type theory in pure lambda calculus [EB/OL]. http://www.ps.uni-sb.de/Papers/abstracts/KaminskiS_July_2008

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

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

操作规则:

$$\frac{\Gamma, A, B \Rightarrow C}{\Gamma, A \cdot B \Rightarrow C} (\cdot L);$$

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

TDL_{min} 系统的代数语义模型定义如下.

定义 3. TDL_{min} 系统的代数结构为 $G = (M, \cdot, \leq)$, G 是一个交换半群, $\leq$ 是 M 上的偏序关系, 并且对所有 $a, b, c \in M$ 满足:

$$(1) a \cdot a \leq a;$$

$$(2) \text{ if } a \leq c, \text{ then } a \cdot b \leq c.$$

易证得: $c \leq a$ and $c \leq b$ iff $c \leq a \cdot b$ (iff $c \leq b \cdot a$) 在 G 中成立.

证明.

情形 1. $c \leq a \cdot b$ iff $c \leq b \cdot a$ 由交换性可得;

情形 2. $c \leq a$ and $c \leq b$ iff $c \leq a \cdot b$

① 左 $\Rightarrow$ 右. 因为 $c \leq a$ 且 $c \leq b$, 所以 $c \cdot c \leq a \cdot b$, 据定义 3(1), 得 $c \leq a \cdot b$;

② 右 $\Rightarrow$ 左. 因为 $a \leq a$, 据定义 3(2), $a \cdot b \leq a$, 因为 $c \leq a \cdot b$, 所以 $c \leq a$.

据情形 1, 2, $c \leq a$ and $c \leq b$ iff $c \leq a \cdot b$ (iff $c \leq b \cdot a$) 在 G 中成立 (简称其为 (I)). 证毕.

定理 1(框架可靠性). TDL_{min} 系统相对于代数语义模型 $G = (M, \cdot, \leq)$ 是可靠的.

据定义 3 各算子的运算规则的对应性, 定理 1 易证得.

定理 2(框架完全性). TDL_{min} 系统相对于代数语义模型 $G = (M, \cdot, \leq)$ 是完全的.

证明. 给定一个序列 (Sequent) $\Gamma_0 \Rightarrow A_0$, 令 S 表示该序列中所有公式的子公式集合, p 为原子公式. 显然, S 是一个有穷集, S 上的语言为 S^* 的子集. 定义 $(M, \cdot, \leq)$ 上的典范赋值 f 如下所示:

$$f(p) = \{\Gamma \in S^* : \Gamma \Rightarrow p\}.$$

我们需要证明的是:

$$(II) f(A) = \{\Gamma \in S^* : \Gamma \Rightarrow A\}.$$

施归纳于 A 的长度:

情形 1. 当 $A = p$ 时, 结论显然.

情形 2. 当 $A = B \cdot C$ 时:

(1) 左 $\Rightarrow$ 右. 设 $\Gamma \in f(B \cdot C)$. 据 (I), $\Gamma \in f(B)$ 且 $\Gamma \in f(C)$, 据归纳假设 $\Gamma \Rightarrow B$ 且 $\Gamma \Rightarrow C$, 再据

$$(\cdot R), \Gamma \Rightarrow B \cdot C;$$

(2) 右 $\Rightarrow$ 左. 设 $\Gamma \Rightarrow B \cdot C$. 据 $(\cdot L)$ 与 Weakening 规则, 易证得 $\Gamma \Rightarrow B$ 且 $\Gamma \Rightarrow C$, 据归纳假设 $\Gamma \in f(B)$ 且 $\Gamma \in f(C)$, 据 (I), $\Gamma \in f(B \cdot C)$.

故据情形 1, 2, (II) $f(A) = \{\Gamma \in S^* : \Gamma \Rightarrow A\}$ 得证.

下证完全性. 设 $\Gamma_0 \not\Rightarrow A_0$, 据 (I), $\Gamma_0 \notin f(A_0)$, 又因为 $\Gamma_0 \Rightarrow \Gamma_0$, 再据 (II) $\Gamma_0 \in f(\Gamma_0)$, 所以 $\mu(\Gamma_0) \notin f(A_0)$, 所以 $\Gamma_0 \not\models A_0$. 故完全性定理得证. 证毕.

针对半群 $G = (M, \cdot, \leq)$ 与时间向量集半群 $S = (Vec(W), \cap, \subseteq)$, 我们可定义同态映射如下.

定义 4. μ 在 $G = (M, \cdot, \leq)$ 上的定义为 $Tp \rightarrow Vec(W)$, 并满足 $\mu(A \cdot B) = \mu(A) \cap \mu(B)$.

由此, 运算性得到保持, 我们得到 TDL_{min} 系统相对于 $S = (Vec(W), \cap, \subseteq, \mu)$ 的模型可靠性和模型完全性.

定理 3(模型可靠性). TDL_{min} 系统相对于时间向量集半群模型 $S = (Vec(W), \cap, \subseteq, \mu)$ 是可靠的.

定理 4(模型完全性). TDL_{min} 系统相对于时间向量集半群模型 $S = (Vec(W), \cap, \subseteq, \mu)$ 是完全的.

3 系统分析与证明

3.1 查询操作与函数依赖

3.1.1 查询操作

根据 TDL_{min} 系统, 及其代数结构与时间向量集半群模型, 我们实际上使得在赋值 $\mu: Tp \rightarrow Vec(W)$ 下, 句法逻辑中的公式 (类型) 映射到了时间模型中, 而其自身剥离了对时间的表示, 从而降低了句法逻辑系统的时间复杂度, 单纯由语义模型完成时间的表达.

如果将句法逻辑中的公式 (类型) 看作是时态数据库的数据表中的属性, 我们实际完成了将属性映射为类型, 类型映射为时间集, 即 “attributes as types, types as time set” 的工作. 由此, 时间向量集半群模型既通过偏序关系表达了基本的时序关系与时序性知识, 又通过属性-类型-时间集映射表示了时间属性知识, 使得相应的对时态知识的表达与操作可单纯由逻辑系统的语义模型完成.

据此, 我们在表 1 中查询 “仅在一个企业并且仅在一个部门工作过的员工”, 实际上就是比较

$\mu(\text{Person_ID} \cdot \text{Co_ID} \Rightarrow \text{Department}) = \mu(\text{Person_ID})$ 是否成立,其查询操作计算过程如下:

(1) 针对 Lee, $\mu(\text{A001} \cdot \text{C001} \Rightarrow \text{Account}) = \{[2009, \text{Now}]\} = \mu(\text{A001})$, $\mu(\text{Person_ID} \cdot \text{Co_ID} \Rightarrow \text{Department}) = \mu(\text{Person_ID})$ 成立,所以 Lee 满足条件;

(2) 针对 Jack, $\mu(\text{A002} \cdot \text{C002} \Rightarrow \text{Marketing}) = \{[2009, 2009], [2011, \text{Now}]\}$, $\mu(\text{A002}) = \{[2009, 2009], [2011, \text{Now}]\} \cup \{[2010, 2010]\} = \{[2009, 2009], [2010, 2010], [2011, \text{Now}]\}$, $\mu(\text{Person_ID} \cdot \text{Co_ID} \Rightarrow \text{Department}) = \mu(\text{Person_ID})$ 不成立,所以 Jack 不满足条件;

(3) 针对 Tom, $\mu(\text{A003} \cdot \text{C003} \Rightarrow \text{IT}) = \{[2009, 2009]\}$, $\mu(\text{A003}) = \{[2009, 2009]\} \cup \{[2010, 2010]\} \cup \{[2011, \text{Now}]\} = \{[2009, 2009], [2010, 2010], [2011, \text{Now}]\}$, $\mu(\text{Person_ID} \cdot \text{Co_ID} \Rightarrow \text{Department}) = \mu(\text{Person_ID})$ 不成立,所以 Tom 不满足条件;

(4) 综上,满足条件的只有 Lee,为查询操作结果.

属性-类型-时间集映射的方式,使得时态知识的表达与处理只需通过代数语义模型进行操作,而主要是线性地基于代数和集合的运算,显得直观而有效率.此外,基于子结构逻辑的构造性,我们实际还处理了时态数据库中经典的 Now 语义过大和过小问题.例如,我们要在表 1 中查询 Tom 在 2012 年、2009 年和 2014 年是否在 IBM 工作,实际上要比较的是这些时间是否在 $\mu(\text{Person_ID} \cdot \text{Co_ID})$ 中,过程如下:

(1) 针对 2012 年,因为 $\mu(\text{A003} \cdot \text{C003}) = \{[2010, 2010], [2011, \text{Now}]\}$,并且存在 $[2011, \text{Now}]$ 包含(contain)了 $[2012, 2012]$ ($[2012, 2012] \in \text{Vec}([2011, \text{Now}])$),所以返回结果为 true, Tom 在 2012 年在 IBM 工作;

(2) 针对 2009 年,对于 $\forall x \in \mu(\text{A003} \cdot \text{C003})$, $[2009, 2009] \notin \text{Vec}(x)$,但 $[2009, 2009] \in \text{Vec}(\mu(\text{A003}))$,所以返回结果为 false, Tom 在 2009 年不在 IBM 工作;

(3) 针对 2014 年,对于 $\forall x \in \mu(\text{A003} \cdot \text{C003})$, $[2014, 2014] \notin \text{Vec}(x)$,同时 $[2014, 2014] \notin \text{Vec}(\mu(\text{A003}))$,所以返回结果为 uncertain,所以无法判定 Tom 在 2014 年是否在 IBM 工作.

针对 Now 语义的问题,经典二值逻辑无法处理

其不确定性问题,但是基于子结构逻辑的构造型语义,不确定性得以表达和处理.

3.1.2 函数依赖

在代数语义模型对时态知识进行表达并做操作运算的同时,句法逻辑主要提供的是合理性支撑,即 $\mu(\text{Person_ID} \cdot \text{Co_ID} \Rightarrow \text{Department})$ 与 $\mu(\text{Person_ID})$ 这样的映射和比较是否合理.而 $\text{Person_ID} \cdot \text{Co_ID} \Rightarrow \text{Department}$ 实际是时态数据库在时间约束下的函数依赖,即 $\text{Person_ID} \cdot \text{Co_ID} \Rightarrow_T \text{Department}$,因此通过对时间向量半群模型的赋值与映射, $\mu(\text{Person_ID} \cdot \text{Co_ID} \Rightarrow \text{Department})$ 与 $\mu(\text{Person_ID})$ 的比较实际则是在时间模型约束下的函数依赖推理与计算.为证明合理性,我们将证明, $\text{TDL}_{\min}$ 系统恰好等价于 Armstrong 函数依赖规则,由此通过属性-类型-时间集的映射方式,以上述语义模型对时态知识进行表达和计算是合理的.

定义 5. Armstrong 函数依赖规则.

设有关系模式 $R(A_1, A_2, \dots, A_n)$ 和属性全集 $U = A_1 A_2 \dots A_n$, X, Y, Z, W 均是 U 的子集, F 是 R 的函数依赖集,Armstrong 推理规则如下:

自反律(Reflexivity). 若 $Y \subseteq X$,则 $X \rightarrow Y$;

增广律(Augmentation). 若 $X \rightarrow Y$,则 $XZ \rightarrow YZ$;

传递律(Transitivity). 若 $X \rightarrow Y, Y \rightarrow Z$,则 $X \rightarrow Z$.

欲证明 $\text{TDL}_{\min}$ 与 Armstrong 函数依赖规则等价,先定义 Φ 是形如 $A \Rightarrow B$ 的一个有穷序列集合,定义 λ 是一个从 $\text{TDL}_{\min}$ 上的类型到属性全集 U 的满射,形如: $\lambda(p) = \{p\}$ 且 $\lambda(A \cdot B) = \lambda(A) \cup \lambda(B)$. 现需证明如下定理.

定理 5 (等价性定理). $\Phi \vdash_{\text{TDL}_{\min}} A \Rightarrow B$ iff $\lambda(A) \rightarrow \lambda(B)$ 据 Armstrong 函数依赖规则.

证明. 从左至右的证明是平凡的,通过 Weakening 规则和 Contraction 规则,我们易证得, $\text{TDL}_{\min}$ 满足 Armstrong 函数依赖规则的自反律、增广律和传递律.我们需要证明的是右至左,以自反律为例证明.

如果 $\lambda(A) \rightarrow \lambda(B)$ 是通过自反律得到,则 $\lambda(B) \subseteq \lambda(A)$. 根据 λ 的定义, $\lambda^{-1}(B)$ 是 $\lambda^{-1}(A)$ 的子公式. 根据 Weakening 规则, $\Phi \vdash_{\text{TDL}_{\min}} \lambda^{-1}(A) \Rightarrow \lambda^{-1}(B)$, 即 $\Phi \vdash_{\text{TDL}_{\min}} A \Rightarrow B$. 证毕.

类似地,我们可以证明增广律和传递律,证明从略.

函数依赖是数据库设计理论的基础,是数据库设计理论中的核心概念,可以规范属性之间满足语

义约束,同时也是数据库中推理与运算的重要方式与工具. 由于时间维的引入,时态数据库中的函数依赖更为复杂,主要体现在时间约束下的数据依赖关系. 我们证明了 $TDL_{\min}$ 与传统数据库的函数依赖 Armstrong 推理规则是恰好等价的,一方面从句法逻辑上对代数语义模型的“属性-类型-时间集”映射与操作运算方式提供了合理性支持,另一方面,从“恰好”等价也说明了 $TDL_{\min}$ 是极小的、最小的.

3.2 计算复杂性

3.2.1 $TDL_{\min}$ 的判定性

$TDL_{\min}$ 系统在句法逻辑中剥离了对时间的表示,而改由语义模型完成对时态运算与操作的支持,使得逻辑系统的复杂性大大降低. 因此,使其不但具有良好的知识表达能力,而且还是可判定的,使得逻辑系统的时间复杂度为 P-time($O(n^2)$). 以下将证明其判定性与时间复杂度.

欲证明 $TDL_{\min}$ 系统是可判定的,首先要证明其是可切割消除的(Cut-free),欲证明切割消除定理,我们可先将系统转换成另一种形式,即以传统的“ $\wedge$ ”取代“ $\cdot$ ”算子和 Weakening 与 Contraction 规则.

易证得,在结构规则 Weakening 和 Contraction 下的“ $\cdot$ ”算子与经典的“ $\wedge$ ”是等价的.

证明.

(1) $\wedge \Rightarrow \cdot$

$$\frac{\frac{A \Rightarrow A \quad B \Rightarrow B}{A, B \Rightarrow A \cdot B} (\cdot R) \quad A \wedge B \Rightarrow A \quad A \wedge B \Rightarrow B (\wedge L)}{\frac{A \wedge B, A \wedge B \Rightarrow A \cdot B}{A \wedge B \Rightarrow A \cdot B} (\text{Contraction})} (\text{Cut}).$$

(2) $\cdot \Rightarrow \wedge$

$$\frac{\frac{A \Rightarrow A}{A, B \Rightarrow A} (\text{Weakening}) \quad \frac{B \Rightarrow B}{A, B \Rightarrow B} (\text{Weakening})}{\frac{A \cdot B \Rightarrow A}{A \cdot B \Rightarrow A} (\cdot L) \quad \frac{A \cdot B \Rightarrow B}{A \cdot B \Rightarrow B} (\cdot L)} (\wedge R).$$

证毕.

因此,我们的系统可以转化成只含“ $\wedge$ ”操作规则和 Cut 结构规则的逻辑系统(如下所示),并且我们可以证明 Cut 规则也是可以消除的.

公理: $p \Rightarrow p$ (Id).

结构规则:

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

操作规则:

$$\frac{\Gamma, A \Rightarrow C}{\Gamma, A \wedge B \Rightarrow C} (\wedge L);$$

$$\frac{\Gamma \Rightarrow A \quad \Gamma \Rightarrow B}{\Gamma \Rightarrow A \wedge B} (\wedge R).$$

定义 6. 切割度. 令 $d(x)$ 表示类型公式 x 中出现的连接词 $\wedge$ 的数量. 且 $d(x_1, x_2, \dots, x_n) = d(x_1) + d(x_2) + \dots + d(x_n)$. 则一个切割的度为 $d(\Delta) + d(\Gamma) + d(\Pi) + d(A) + d(B)$.

定理 6(切割消除定理). 在系统的任意切割中,当所有切割的前提不是由切割推演得到时,该切割的结论存在且只存在两种可能:(1) 与结论前提中的某一个一致;(2) 该切割可以被一个或两个切割度更小的切割替换.

证明(分 5 种情况).

情形 1. $\Delta \Rightarrow A$ 是公理的实例;

情形 2. $\Gamma, A, \Pi \Rightarrow B$ 是公理的实例;

情形 3. 证明 $\Delta \Rightarrow A$ 的最后一步推演中没有引入 A 中的主连接词;

情形 4. 证明 $\Gamma, A, \Pi \Rightarrow B$ 的最后一步推演中没有引入 A 中的主连接词;

上述 4 种情况的切割消除证明与经典子结构逻辑系统 Lambek 演算相似,我们需要证明的是以下一种情况.

情形 5. 切割前提推演的最后一步都引入了 A 的主连接词 $\wedge$, $A = A' \wedge A''$. 可将原切割

$$\frac{\frac{\Delta \Rightarrow A' \quad \Delta \Rightarrow A''}{\Delta \Rightarrow A' \wedge A''} (\wedge R) \quad \frac{\Gamma, A' (\text{or } A''), \Pi \Rightarrow B}{\Gamma, A' \wedge A'', \Pi \Rightarrow B} (\wedge L)}{\Gamma, \Delta, \Pi \Rightarrow B} (\text{Cut})$$

置换为

$$\frac{\Delta \Rightarrow A' (\text{or } A'') \quad \Gamma, A' (\text{or } A''), \Pi \Rightarrow B}{\Gamma, \Delta, \Pi \Rightarrow B} (\text{Cut}).$$

新的切割比原切割有更小的度.

据情形 1~5,我们证明了在 $TDL_{\min}$ 系统中,切割定理是可以被消除的. 证毕.

据切割消除性,我们可证明 $TDL_{\min}$ 系统具有有穷判定性.

定理 7(判定性定理). 在上述系统中,给定一个 Sequent: $\Gamma \Rightarrow A$,其是可判定的.

证明. 设给定一个 Sequent: $\Gamma \Rightarrow A$,我们可对其构造一棵自底向上的证明树,该树的每一分支为一证明. 对于该树的构造,可使用 $(\wedge L)$ 和 $(\wedge R)$ 规则(注意到 Cut 规则已经被消除). 因此我们可将每次回溯过程构造为树中的一个点,形成一棵新树. 在新树中,每一个向上的步骤,都能消除连接词 $\wedge$ 的一次出现,注意到这样的回溯总是有穷的,因此该树是有穷的. 据此 Sequent: $\Gamma \Rightarrow A$ 是可判定的当且仅当

该树的某一支能证明之。

证毕。

3.2.2 TDL_{min}的计算复杂性

TDL_{min}系统的时间复杂度为 P-time($O(n^2)$), 我们将使用有向图的方式对其进行证明^[15]. 在此之前, 我们需先证明一个引理和一个推论。

引理 1. 对 $\forall p_i \in A$ (p_i 为原子公式), $\Phi \vdash_{\text{TDL}_{\min}} \Gamma \Rightarrow A$ iff $\Phi \vdash_{\text{TDL}_{\min}} \Gamma \Rightarrow p_i$.

证明.

情形 1. 当 $A = p_i$ 时, 结论显然;

情形 2. 当 $A = B \cdot C$ 时:

(1) 左 $\Rightarrow$ 右. 据 Weakening 规则和 $(\cdot L)$, $B \cdot C \Rightarrow B$ 且 $B \cdot C \Rightarrow C$, 再据 Cut 规则, $\Gamma \Rightarrow B$ 且 $\Gamma \Rightarrow C$, 据归纳假设 $\Gamma \Rightarrow p_i$;

(2) 右 $\Rightarrow$ 左. 对 $\forall p_i \in A$, 如果 $\Gamma \Rightarrow p_i$, 据有穷次是用 $(\cdot R)$ 和 Contraction 规则, 易得 $\Gamma \Rightarrow A$.

故据情形 1, 2, $\forall p_i \in A$, $\Phi \vdash_{\text{TDL}_{\min}} \Gamma \Rightarrow A$ iff $\Phi \vdash_{\text{TDL}_{\min}} \Gamma \Rightarrow p_i$ 得证. 证毕。

推论 1. $\forall p_i \in A$ (p_i 为原子公式), $\Phi \vdash_{\text{TDL}_{\min}} \Gamma \Rightarrow A$ iff $\Phi^* \vdash_{\text{TDL}_{\min}} \Gamma \Rightarrow p_i$, 其中 Φ^* 表示由 Φ 产生的所有形如 $A \Rightarrow p_i$ 的公式集 (Horn 子句)。

定义 7 (有向图). 有向图 G 由三元组 $G = (Gn, R, W)$ 组成, 其中 Gn 表示图中结点的集合, R 表示图中有向边的集合. 一条有向边由二元序对 (a_i, a_j) 组成, 其中 $a_i, a_j \in Gn$, $w_i \in W$ 为每条边上的权值. G 上从结点 a 到 b 的路径为 R 中元素的序列, 形如 $(a_1, a_2), (a_2, a_3), \dots, (a_{n-1}, a_n)$, 其中 $a_1 = a$ 且 $a_n = b$, $w(a, b) = \prod w(a_i, a_j)$, 其中 $(0 \leq i \leq j \leq n)$. 两条路径 (a, b) 和 (c, d) 是分离的 (disjoint) 当且仅当 $(a, b) \not\subseteq (c, d)$ 且 $(c, d) \not\subseteq (a, b)$.

据此, 我们可以根据上述定义, 构造 TDL_{min}(Φ^*) 的有向图, 过程如下:

(1) 令 $Gn = \forall p_i \in \Phi$, p_i 为原子公式;

(2) 对 Φ^* 任意序列, 如 $p_1 \cdots p_n \Rightarrow q$, 将 $(q, p_1), \dots, (q, p_n)$ 加入到 R 中, 其权值为 $(q, p_1) = \dots = (q, p_n) = 1/n$. 如图 2 所示。

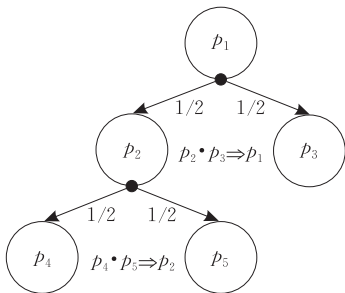


图 2 TDL_{min}(Φ^*) 有向图示例

这个构造过程是 P-time 的, 其时间复杂度为 $O(n)$. 特殊的, 针对时态数据库, Φ 可以为其函数依赖集. 以下利用有向图证明 TDL_{min} 系统判定算法的时间复杂度。

根据推论 1, 欲证明 $\text{Sequent}; \Gamma \Rightarrow A$ 在给定公式集 Φ 是否能被判定, 仅需证明在给定公式集 Φ^* 中, 是否对 $\forall p_i \in A, \Gamma \Rightarrow p_i$. 为此可构建 TDL_{min}(Φ^*) 有向图, 并作扫描如下:

(1) 对 $\forall p_i \in A$, 依次扫描;

(2) 如果 $p_i \in \Gamma, \Gamma \Rightarrow p_i$, 结论显然;

(3) 如果 $p_i \notin \Gamma, \Gamma \Rightarrow p_i$ 当且仅当存在若干条分离的路径, 形如 $(p_i, a_1), \dots, (p_i, a_n)$ 使得 $a_1, \dots, a_n \in \Gamma$ 且 $\sum w(p_i, a_i) \geq 1 (1 \leq i \leq n)$;

(4) 如果 $\exists p_i \in A$ 不存在步 3 中的路径, 则 $\Gamma \Rightarrow A$ 在给定公式集 Φ 下不可判定; 否则 $\Gamma \Rightarrow A$ 可判定。

显然, 这个扫描过程是 P-time 的, 由于经过了两个嵌套循环的扫描 (一次为 $p_i \in A$, 另一次为 $a_i \in \Gamma$), 易知其最坏时间复杂度为 $O(n^2)$. 这个时间复杂度对于绝大部分逻辑系统 (包括模态逻辑、线性逻辑、子结构逻辑等) 来说, 已经是非常好的结果, 经典一阶逻辑扫描真值表的时间复杂度也为 $O(n^2)$.

4 TDL_{min}与 Allen 方法的对接

TDL_{min} 系统对时间的表示在语义模型中实现, 而在句法逻辑中剥离对时间的表示, 从而降低了逻辑系统的复杂性, 提高了直观性. 这样也使得时态查询将分两步进行 (如图 3). 第 1 步为传统的非时态查询, 其可兼顾时态知识和非时态知识, 对于时态知识在第 1 步查询中不对时间进行演算, 留到第 2 步进行; 第 2 步为时态查询, 即针对子结构逻辑语义模型的时间表达和演算对时态知识进行二次计算。

在第 2 步查询中, 实际上我们要完成的事情, 是根据语义模型从类型到时间向量集半群的赋值 $(\mu; Tp \rightarrow \text{Vec}(W))$, 对时间进行运算和比较, 而这些运算和比较是可以划分成若干类型进行的. 在 3.1 节的两例中, 我们实际完成了两种类型的运算:

(1) Equals. 判断两个时间向量集是否等价, 如查询“仅在一个企业并且仅在一个部门工作过的员工”;

(2) During and Contains. 判断一个时间向量集中是否存在时间向量, 而这个时间向量包含另一个时间向量集, 如查询“Tom 在 2012 年、2009 年和

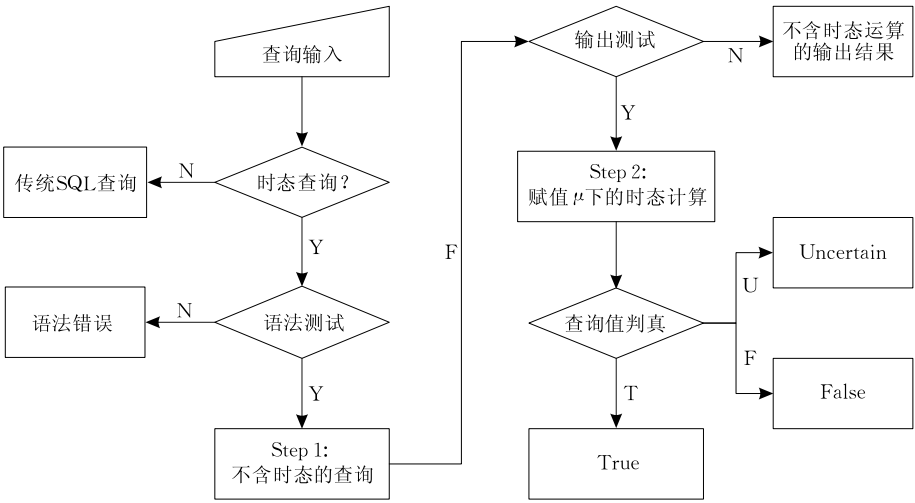


图 3 “两步式”时态查询流程图

2014 年是否在 IBM 工作”的例子。

由于时态数据库中,较多地采用线性时间而非分支时间(Branching Time),线性时间又具有单向性,因此时间向量集又是一个顺序集,完成上述两种操作的时间复杂度仅为 $O(n)$ 。因此两步式查询完成上述两种操作的时间复杂度为: $O(T_Q)+O(n)$,其中 $O(T_Q)$ 为第 1 步查询的时间复杂度,即传统的非时态查询的时间复杂度。

这表明,我们完成相关查询与推理计算时,所需要的时间代价是在传统非时态查询的基础上,加上一个时间复杂度为 $O(n)$ 的时间向量集运算。由于时间复杂度 $O(n)$ 是线性的,所以实际整个查询的时间复杂度仅取决于 $O(T_Q)$,这是一个较为理想的结果。而除了 Equals 和 During(Contains) 运算外,由于时间向量集中的元素主要采取的也是时间区间的表示方法,根据不同查询与应用,我们还能完成的运算有

- (1) Before and after;
- (2) Overlaps and Overlapped-by;
- (3) Meets and Met-by;
- (4) Starts and Started-by;
- (5) Finishes and Finished-by.

这实际和传统的基于区间逻辑和时间区间运算的 Allen 方法^[16]是完全对接的。1983 年 Allen 提出了 13 种时态关系运算:“before”,“after”,“during”,“contains”,“overlaps”,“overlapped-by”,“meets”,“met-by”,“starts”,“started-by”,“finishes”,“finished-by”和“equals”,相关研究结果被应用于时态查询语言 TSQL2 中^[17]。

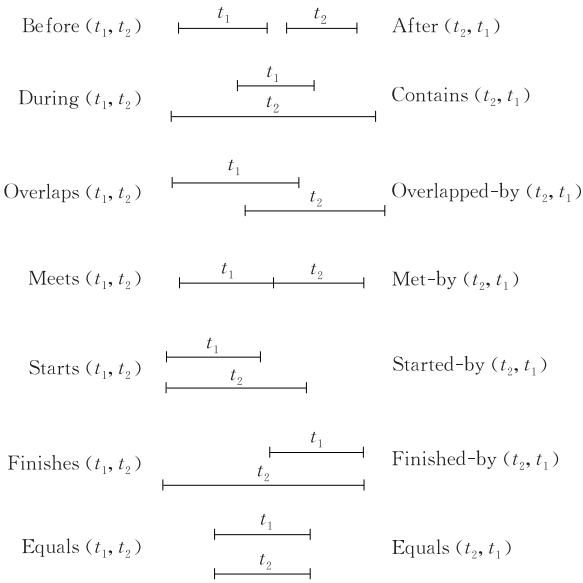


图 4 Allen 方法的 13 种时态区间运算

这种对接也使得 TDL_{min} 系统具有更强的普适性和应用前景。首先,其恰好等价于 Armstrong 函数依赖规则(极小性源出于此),由此通过属性-类型-时间集的映射方式,为单纯由语义模型对时态知识进行表达和计算的方法提供了合理性支撑;其次,其在句法逻辑中剥离了时间的表示,降低了逻辑系统的复杂度,使得通过该逻辑系统进行的表达与推理运算,时间复杂度是 P-time($O(n^2)$)的,而且还能完成不确定性推理;最后, TDL_{min} 系统能与经典的时态数据库查询语言 TSQL2 进行对接,匹配经典的时态关系运算 Allen 方法,而同时相应的查询时间复杂度接近于传统非时态查询时间复杂度,仅为 $O(T_Q)+O(n)$ 。

5 结束语

子结构逻辑是一类构造性逻辑,采用构造性证明,具有较强的计算性.从证明论角度出发,子结构逻辑将逻辑系统分为结构规则和操作规则.直观上来说,其 Gentzen 式的风格有别于传统 Hilbert 风格,能把逻辑系统“掰”碎了,通过对模态规则、结构规则与逻辑算子(类似时序性知识)的灵活增删,映射并借助代数语义与类型论(类似时间属性知识),在表达能力、判定性和计算复杂性方面对逻辑系统进行综合考量,其在语义上的近似性能进一步转化为关系代数及相关语义,从而有利于构建时态数据运算体系和数据库建模.

为此,我们从子结构逻辑出发,针对时态数据库及其信息处理中关键的知识推理、时态操作与函数依赖等内容,构建了一个极小的子结构逻辑系统 TDL_{min} ,其只存在一个类似于传统合取连接词“ $\wedge$ ”的连接词“ $\cdot$ ”和最少的结构规则(2 条),并恰好等价于传统的数据库函数依赖 Armstrong 规则. TDL_{min} 系统能在逻辑语义模型中对时序性、时间属性进行表达,而在句法逻辑系统将时间剥离,从而既表达了时态知识,对时态操作进行处理,又降低了逻辑系统的复杂度,使得逻辑系统的时间复杂度为 $P\text{-time}(O(n^2))$. 而该逻辑系统还可与传统的 Allen 方法进行对接,使得相关时态查询所需的时间代价为传统非时态查询的时间加上一个复杂度仅为 $O(n)$ 的线性时间.

由于子结构代数语义模型与关系代数的相似性,在后续研究中,我们希望将 TDL_{min} 系统或相关变形系统能与时态关系代数对接,在语义模型层面形成等价的关系代数模型或可被关系代数运用的子代数模型.此外,针对多粒度的时态函数依赖(TMVD),我们也希望能进一步推广 TDL_{min} 系统的相关方法,并做相关工作.

参 考 文 献

[1] Tang Yong. Temporal Information Technology and Its Applications. Springer/TUP, 2010. 07

[2] Papadakis N, Plexousakis D, Papadakis M, Manifavas H. The ramification problem in temporal databases: An approach with conflicting constraints. Journal of Intelligent Information Systems, 2012, 38(1): 41-94

[3] Gabbay D M, Hodkinson I, Reynolds M. Temporal Logic: Mathematical Foundations and Computational Aspects. Dxford: Oxford University Press, 1994

[4] Chomicki J, Saake G. Logics for Databases and Information Systems: Temporal Logic Information Systems. Boston/Dordrecht/London: Kluwer Academic Publishers, 1998

[5] Máté J, Safarik J. Transformation of relational databases to transaction-time temporal databases//Proceedings of the 2011 Second Eastern European Regional Conference on the Engineering of Computer Based Systems (ECBS-EERC'11). Bratislava, 2011: 27-34

[6] Damaggio E, Deutsch A, Zhou Dayou. Querying contract databases based on temporal behavior//Proceedings of the 2011 International Conference on Management of Data (SIGMOD'11). ACM, New York, 2011: 397-408

[7] Jiang Yun-Cheng, Tang Yong, Wang Ju, Ji Gao-Feng. Description-logic-based temporal ER model with attribution dependencies. Journal of Computer Research and Development, 2007, 44(10): 1765-1773(in Chinese)

(蒋永承, 汤庸, 王驹, 冀高峰. 基于描述逻辑的带属性依赖时序 ER 模型. 计算机研究与发展, 2007, 44(10): 1765-1773)

[8] Liu Dong-Ning, Tang Yong. Dynamic logic model of time axes in temporal database. Journal of Software, 2010, 21(4): 694-701(in Chinese)

(刘冬宁, 汤庸. 时态数据库时间轴的动态逻辑模型. 软件学报, 2010, 21(4): 694-701)

[9] Saltzer J, Kaashoek M F. Principles of Computer System Design. ACM New York: Morgan Kaufmann, 2009

[10] Goble Lou. The Blackwell Guide to Philosophical Logic. USA: Blackwell Pulishers Inc., 2001

[11] Galatos N, Jipsen P, Kowalski T, Ono H. Residuated Lattices: An Algebraic Glimpse at Substructural Logics. Elsevier, San Diego, USA, 2007

[12] Buszkowski W. Type logics and pregroups. Studia Logica, 2007, 87(2/3): 145-169

[13] Hao Zhong-Xiao. Theory of Temporal DataBase Design. Beijing: Science Press, 2009(in Chinese)

(郝忠孝. 时态数据库设计理论. 北京: 科学出版社, 2009)

[14] Xu Chunyan, Jiang Feng; Liu Guixia. The study of temporal scheme in temporal database//Proceedings of the 2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC). Jilin, China, 2011: 1473-1476

[15] Buszkowski W, Lin Zhe, Moroz K. Pregroup grammars with letter promotions: Complexity and context-freeness. Journal of Computer and System Science, 2012, 78(6): 1899-1909

[16] Allen J F. Maintaining knowledge about temporal intervals. Communications of the ACM, 1983, 26(11): 832- 843

[17] Snodgrass R T. The Temporal Query Language. Norwell, MA, USA: Kluwer Academic Publishers, 1995



LIU Dong-Ning, born in 1979, Ph. D. , associate professor. His research interests include artificial intelligent, procession of temporal information.

TANG Yong, born in 1964, Ph. D. , professor. His research interests include procession of temporal information.

TENG Shao-Hua, born in 1962, Ph. D. , professor. His research interests include network security, data mining, cooperative computing.

LIN Zhe, born in 1983, Ph. D. candidate. His research interests include substructural logic, formal language.

Background

Logic is an important method of knowledge representation, but because of the complexity of the cross application in tense and temporal attributes knowledge, it has been disappointing on the temporal database support.

In the fields of temporal knowledge presentation, axiomatic systems and proof method for temporal logic have found so far relatively few applications in the inference modeling of temporal database in 1994 by Gabbay. Thus, non-axiomatic temporal logics are used in temporal inference nowadays. But because of run short of intuitionist and axiomatic style's restriction, non-axiomatic system does not be used in knowledge inference. On the other hand, in 1983, Allen established one kind of interval-based temporal logic with thirteen temporal calculus, which now be used in TSQL2. But Allen's method can only explain algorithm of time, it could not reflect from time to attribute, so it could not be used in attributes' inference too. Furthermore, there are still many open problems in research of temporal database, such as uncertain semantics of Now in temporal querying and inference.

Based these, we put forward a minimum substructuaral

logic(TDL_{min}), which is aimed at the critical area of research fields of temporality inference, temporality operating, temporal relational algebra and functional dependency of temporal database and its information processing. As substructural logic and its universal algebra model are logical tools, temporal knowledge is represented only in the semantics model of TDL_{min} , and the syntactic logic has been deprived in presentation of temporality, which is cut down the complexity of logic system so that TDL_{min} is P-time and its time complexity is only $O(n^2)$. Moreover the syntactic logic of TDL_{min} is properly equal to the Armstrong Rules of functional dependency of traditional database, while the semantics model can build up a method corresponding to Allefs Method of traditional temporal query, which led to the time cost of temporal query is equal to traditional query without temporality plusing a linear time whose complexity is only $O(n)$.

This paper is supported by the National Natural Science Foundation of China under Grant Nos. 61272067, 60970044, 61104156; the Natural Science Foundation of Guangdong Province of China under Grant Nos. 10451009001004804, S2012030006242.