

不确定时态信息表示及时态运算的扩展^{*})

林嘉宜 彭宏 谢嘉孟 郑启伦

(华南理工大学计算机科学与工程学院 广州 510641)

摘要 时态信息表示和查询是时态数据库的研究重点,但现有的时态数据库模型在处理不确定时态信息方面仍存在较大困难。而现实生活中很多时态信息都是不确定的,故提出了一种不确定时态信息的表示方法,并扩展时态运算使其支持不确定时态信息的处理和查询,从而扩展了时态数据库的适用范围。

关键词 不确定时态信息,时态数据库,时态运算

Uncertain Temporal Information Representation and the Extensions of Temporal Operation

LIN Jia-Yi PENG Hong XIE Jia-Meng ZHENG Qi-Lun

(College of Computer Science and Engineering, South China University of Technology, Guangzhou 510641)

Abstract Temporal information representation and query are the main research topics in temporal database. It still remains difficulties in handing uncertain temporal information in temporal database while in real life many events happen with uncertain temporal information. A new model for representing uncertain temporal information is proposed and the extensions of various temporal operations are also presented in this paper. The model and the extensions enrich the flexibility of temporal database.

Keywords Uncertain temporal information, Temporal database, Temporal operation

1 引言

时间这种基本的概念和信息科学中的“信息”一样难以精确定义,然而,我们却常以时间为依据来解释各种事实和数

据,因为各种事件和实体间的关系常常蕴涵在时态信息中。对时态信息表示的研究大约起于 20 世纪 50 年代末 60 年代初,当时,这一问题在人工智能界中开始并未引起重视。80 年代初,Allen^[1]提出基于时间区间的方法,定义了 13 种定性的确定时态关系,为区间的时态推理奠定了基础,后来不少学者对他的模型进行了扩展和改良。之后, Vilain 和 Kautz^[2]提出基于时间点的方法。到了 90 年代, Schwlb 与 Decher^[3]基于 Meiri^[4]模型研究了不确定约束的处理方法,使时态信息表示和时态推理得到较快的发展。

数据库是现今储存信息的重要手段。时态数据库 TDB (Temporal Database),是在传统的数据库基础上加上时间维,不仅能刻画某个时刻的数据,还能反映出其历史和揭示其未来^[5]。时态数据库的先前工作,主要是建立在有效时间的开始、终止都十分清楚的基础上。然而,在许多实际情况中,事件的时间界限并不十分清楚。因此,表示不确定时间,是时态数据库中目前仍缺少的一个重要的功能^[6]。而这一缺陷极大地约束了时态数据库的应用范围,影响了时态数据库的进一步推广。

本文给出了一种不确定时态信息的表示方法,着重说明其基本思想及意义,最后扩展时态运算使其支持不确定时态信息的处理和查询。该表示方法和时态运算的扩展可以解决时态数据库中处理不确定时态信息存在的一些问题,增强了

时态数据库对不确定时态信息处理的支持,丰富了时态数据库查询语言,从而扩展了时态数据库的适应范围。

2 时态数据库中的不确定时态问题

时态数据库中的时态关系模式属性集可以分为非时态属性集(值与时间无关)和时态属性集(值随时间变化)两部分。我们可以用属性-时间对来表示时态数据库中元组的某个属性的值和有效时间(Valid Time)的关系,如某元组的属性 F 有值(value, $\langle t_1, t_2 \rangle$),表示元组在有效时间 $\langle t_1, t_2 \rangle$ 内,属性 F 具有值 value。

在时态查询语言中,当满足比较关系的时候,时态比较运算符(如 $>_T$ 、 $<_T$ 、 $=_T$ 等)返回满足条件的有效时间集。而时态运算符 $OVERLAP_T$ 、 $UNION_T$ 和 NOT_T 则是在相应的运算符 $OVERLAP$ 、 $UNION$ 和 NOT 上进行的时态扩展,返回的也是有效时间集。我们用符号 0 表示数据库系统中时间的最小值;用符号 ∞ 表示数据库系统中时间的最大值。 $OVERLAP_T$ 返回两个时间的交集; $UNION_T$ 返回两个时间的并集; NOT_T 则返回时间在 $\langle 0, \infty \rangle$ 上的补集。例如: $\{\langle 9, 25 \rangle, \langle 30, 53 \rangle\} OVERLAP_T \{\langle 20, 33 \rangle\} = \{\langle 20, 25 \rangle, \langle 30, 33 \rangle\}$ 。

以上的时间区段的界限都是确定的,各种时态运算也都是基于确定时间的。到目前为止,关于历史和时态的数据库都是定义在确定的时间区段上,例如,某元组的时态属性 Play 具有值 $\{\text{Football}, \langle 59, 76 \rangle\}$ 表明相应的人在 59 到 76 这段时间内踢足球。然而,在现实中很多情况下,我们往往不能很明确知道事件的确切发生时间和结束时间,只知道事件大概是什么时候发生的或结束的,例如,我们可能只是模糊地知道这

^{*} 基金项目:广东省科技攻关项目(A10202001);广州市科技攻关项目(2004Z2-D0091)。林嘉宜 博士研究生,主要研究方向:智能计算和智能网络技术;彭宏 教授,博士生导师,主要研究方向:智能网络技术;谢嘉孟 博士研究生,主要研究方向:数据挖掘和智能交通;郑启伦 教授,博士生导师,主要研究方向:智能信息处理及其应用。

件事发生在“下午 2 点到 3 点之间”，“大概在这个月的上旬”等等。现实生活中比较典型的这类例子是医疗数据，病人的症状持续的开始时间和结束时间一般都只有一个大概的范围，无法精确地给出具体的时间，如果用传统的时态数据库来表示这种类型的时态信息，就会遇到很大困难。因此，我们扩展了时态数据库中时态信息的表示方法，使它能处理不确定时态信息。

3 不确定时态信息的表示方法

由于只涉及时态属性的不确定时态信息表示，所以这里只需定义三种时态元素：不确定时间点、不确定时区和时距。

3.1 时间点

定义 1 确定时间点是稠密时间轴上的一个点，在本模型中是不可再分的最小时间单元。

定义 2 不确定时间点 tp 是一个随机变量，其取值域为一个时间区间 $\langle ps, pf \rangle$ ，表示为 $tp = \langle ps, pf \rangle$ ，其中 ps 和 pf 分别是稠密时间轴上的确定时间点，且有 $ps \leq pf$ 。当 $ps = pf$ 时， tp 就成为一个确定的时间点。

根据上述定义，不确定时间点 tp 是在 ps 和 pf 之间取值，其值是不确定的。例如，某病人从 5:00 到 5:05 之间开始出现发烧症状，可表示为 $tp = \langle 5:00, 5:05 \rangle$ 。当 $ps = pf$ 时， tp 就成为一个确定的时间点，如张三在 8:00 离家上班，可表示为 $tp = \langle 8:00, 8:00 \rangle$ 。该定义中也包括了空时间点的情况，即无法确定该时间点上的事件是否发生，例如， $\langle 6, 9 \rangle$ 表示一个事件可能在时间 6 到时间 9 之间发生，但是仍然不能确定该事件是否真的发生过。

3.2 时区

定义 3 时区是描述事件成立的时间区间，由两个时间点所组成的一个二元组 $ti = \langle is, if \rangle$ 来表示，其中 $is = \langle iss, isf \rangle$ 和 $if = \langle ifs, iff \rangle$ 分别代表时间区间的开始点和结束点，且有 $iss < ifs$ 和 $isf < iff$ 。

当开始时间点和结束时间点都是确定时间点的时候，此时区就是一个确定时区；当两者之一或两者是不确定时间点时，此时区是一个不确定时区。例如，李四每天早上 8:00 到 8:10 之间出门上班，由于交通的缘故，他到达单位的时间大概是 8:40 到 9:00 之间，这个时间区间可表示为 $ti = \langle \langle 8:00, 8:10 \rangle, \langle 8:40, 9:00 \rangle \rangle$ 。

同样，定义 3 描述的时区可能包括空时区（当开始时间点和结束时间点重叠的时候）的情况，即无法确定该时区上的事件是否发生。

3.3 时距

定义 4 时距是描述事件成立的时区长度或两个时间点之间的时间长度，时距 tl 是一个随机变量，其值分布在某个区间内，表示为 $tl = \langle ls, lf \rangle$ 。

我们用 $tl(ti)$ 表示时区长度，用 $tl(tp_1, tp_2)$ 表示两个时间点之间的时间长度，其中 ti 指不确定时区， tp_1, tp_2 指不确定时间点。

当时距表示时区的长度时， ls 和 lf 是两个非负实数，分别表示时区持续的最短和最长时间；当时距表示两时间点的时间距离时， ls 和 lf 是实数，分别表示时间点之间距离的最短和最长值，其正负号指明了两个时间点的时间轴上位置的前后关系。

设时区 $ti = \langle is, if \rangle = \langle \langle iss, isf \rangle, \langle ifs, iff \rangle \rangle$ 且 $isf \geq ifs$ ，如果再加一约束条件 $CP: tl(ti) = 0$ ，则该时区退化为时间点 $\langle ifs, isf \rangle$ 。可见，一定的时距定量约束能促使不确定时区转化为不确定时间点。

以上定义的三类时态元素可以表示不确定时态信息。下面，我们对时态运算进行扩展，使之能处理时态元素间的运算。

4 时态运算的扩展

这里的时态运算主要是针对各时态元素（时间点、时区）的取值域而进行的运算。在时态数据库中，时间的表示并不是任意的，由于存储长度的限制，时间的表示总有一个范围，存在一个最大最小值。应用中，我们将该时间表示的最小值定为时间轴的起点，用符号 0 表示，时间表示的最大值则用符号 ∞ 表示，那么下述各种时态运算的时态范围就是 $\langle 0, \infty \rangle$ 。设有时态元素 t_1 和 t_2 (t_1 和 t_2 可以是时间点或时区)，我们对各类时态运算扩展如下。

4.1 时态 OVERLAP_T 运算符

t_1 OVERLAP_T t_2 运算：返回的结果是时态元素 t_1 和 t_2 的时态交或可能的时态交。即， t_1 OVERLAP_T t_2 返回 t_1 和 t_2 之间重叠的部分或可能重叠的部分。分三种情况：

1) 时间点与时间点的交返回结果是空集（当两者无重叠部分时）或只有一个时间点的集合，设有两个时间点 $tp_1 = \langle ps_1, pf_1 \rangle, tp_2 = \langle ps_2, pf_2 \rangle$ ，则

$$tp_1 \text{ OVERLAP}_T tp_2 = \begin{cases} \text{i. } \phi (\text{if } pf_1 < ps_2 \text{ OR } pf_2 < ps_1) \\ \text{ii. } \{ \langle \text{Later}(ps_1, ps_2), \text{Earlier}(pf_1, pf_2) \rangle \} (\text{else}) \end{cases}$$

其中 Earlier(p_1, p_2) 操作返回两个参数中较早的时间点，Later(p_1, p_2) 操作返回两个参数中较迟的时间点。例如： $\langle 5, 12 \rangle$ OVERLAP_T $\langle 8, 16 \rangle$ 返回 $\{ \langle 8, 12 \rangle \}$ 。根据定义 2，返回的结果 $\langle 8, 12 \rangle$ 包括了空时间点的情况。

2) 时间点与时区的交返回两者的重叠部分，其结果是空集（当两者无重叠部分时）或只有一个时间点的集合（该结果时间点的上下界标识了两者的重叠区域）。例如： $\langle 3, 5 \rangle$ OVERLAP_T $\langle \langle 8, 16 \rangle, \langle 20, 23 \rangle \rangle$ 返回 ϕ ； $\langle 3, 17 \rangle$ OVERLAP_T $\langle \langle 8, 16 \rangle, \langle 20, 23 \rangle \rangle$ 返回 $\{ \langle 8, 17 \rangle \}$ 。

3) 时区与时区的交返回结果是空集（当两者无重叠部分时）、只有一个时间点的集合或只有一个时区的集合。设两个时区 $ti_1 = \langle \langle iss_1, isf_1 \rangle, \langle ifs_1, iff_1 \rangle \rangle, ti_2 = \langle \langle iss_2, isf_2 \rangle, \langle ifs_2, iff_2 \rangle \rangle$ ：

$$ti_1 \text{ OVERLAP}_T ti_2 = \begin{cases} \text{i. } \phi (\text{if } iff_1 < iss_2 \text{ OR } iff_2 < iss_1) \\ \text{ii. } \{ \langle ifs_1, isf_2 \rangle \} \\ \quad (\text{if } iff_1 = iss_2 \text{ OR } iff_2 = iss_1) \\ \text{iii. } \{ \langle \langle iss_3, isf_3 \rangle, \langle ifs_3, iff_3 \rangle \rangle \} (\text{else}) \end{cases}$$

其中 $iss_3 = \text{Later}(iss_1, iss_2)$ ； $isf_3 = \text{Earlier}(\text{Earlier}(\text{Later}(isf_1, isf_2), iff_1), iff_2)$ ； $ifs_3 = \text{Later}(\text{Later}(\text{Earlier}(ifs_1, ifs_2), iss_1), iss_2)$ ； $iff_3 = \text{Earlier}(iff_1, iff_2)$ 。

例如， $\langle \langle 5, 12 \rangle, \langle 18, 24 \rangle \rangle$ OVERLAP_T $\langle \langle 26, 28 \rangle, \langle 32, 37 \rangle \rangle$ 返回 ϕ ； $\langle \langle 5, 23 \rangle, \langle 26, 30 \rangle \rangle$ OVERLAP_T $\langle \langle 3, 9 \rangle, \langle 18, 24 \rangle \rangle$ 返回 $\{ \langle \langle 5, 23 \rangle, \langle 18, 24 \rangle \rangle \}$ 。

4.2 时态 UNION_T 运算符

t_1 UNION_T t_2 运算：返回的结果是时态元素 t_1 和 t_2 的时态并或可能的时态并。这里共有三种情况：

1) 时间点与时间点的并返回两个时间点的并集，其结果是时间点集。除非两点在最终确定时重叠，否则两个不确定

时间点的并不可能是一个时间点。例如： $\{\langle 5, 8 \rangle\} \text{UNION}_T \{\langle 15, 19 \rangle\}$ 返回 $\{\langle 5, 8 \rangle, \langle 15, 19 \rangle\}$ 。

2) 时间点与时区的并返回的结果是包含时间点和时区的集合或只包含时区的集合。设有时间点 $tp_1 = \langle ps_1, pf_1 \rangle$ ，时区 $ti_2 = \langle \langle iss_2, isf_2 \rangle, \langle ifs_2, iff_2 \rangle \rangle$ ，则：

$$tp_1 \text{UNION}_T ti_2 = \begin{cases} \text{i. } \{ti_2\} & (\text{if } isf_2 < ps_1 \text{ AND } pf_1 < ifs_2) \\ \text{ii. } \{tp_1, ti_2\} & (\text{else}) \end{cases}$$

例如： $\langle 17, 19 \rangle \text{UNION}_T \langle \langle 8, 16 \rangle, \langle 20, 23 \rangle \rangle$ 返回 $\{\langle \langle 8, 16 \rangle, \langle 20, 23 \rangle \rangle\}$ ； $\langle 7, 9 \rangle \text{UNION}_T \langle \langle 8, 16 \rangle, \langle 20, 23 \rangle \rangle$ 返回 $\{\langle 7, 9 \rangle, \langle \langle 8, 16 \rangle, \langle 20, 23 \rangle \rangle\}$ 。

3) 时区与时区的并的结果是包含一个或者两个时区的集合。设有两个时区 $ti_1 = \langle is_1, if_1 \rangle = \langle \langle iss_1, isf_1 \rangle, \langle ifs_1, iff_1 \rangle \rangle$ ， $ti_2 = \langle is_2, if_2 \rangle = \langle \langle iss_2, isf_2 \rangle, \langle ifs_2, iff_2 \rangle \rangle$ ，则：

$$ti_1 \text{UNION}_T ti_2 = \begin{cases} \text{i. } \{ \langle \langle \text{Earlier}(iss_1, iss_2), \text{Earlier}(isf_1, isf_2) \rangle, \langle \text{Later}(ifs_1, ifs_2), \text{Later}(iff_1, iff_2) \rangle \rangle \} \\ \quad (\text{if } isf_1 < isf_2 < ifs_1 \text{ OR } isf_2 < isf_1 < ifs_2) \\ \text{ii. } \{ \langle \langle iss_1, isf_1 \rangle, \langle ifs_1, \text{Earlier}(iff_1, iff_2) \rangle \rangle, \langle \langle \text{Later}(iss_2, ifs_1), isf_2 \rangle, \langle ifs_2, isf_2 \rangle \rangle \} \\ \quad (\text{if } ifs_1 < isf_2) \\ \text{iii. } \{ \langle \langle iss_2, isf_2 \rangle, \langle ifs_2, \text{Earlier}(iff_2, isf_1) \rangle \rangle, \langle \langle \text{Later}(iss_1, ifs_2), isf_1 \rangle, \langle ifs_1, ifs_1 \rangle \rangle \} \\ \quad (\text{if } ifs_2 < isf_1) \\ \text{iv. } \{ti_1, ti_2\} & (\text{else}) \end{cases}$$

如： $\langle \langle 8, 10 \rangle, \langle 20, 21 \rangle \rangle \text{UNION}_T \langle \langle 7, 9 \rangle, \langle 18, 23 \rangle \rangle$ 返回 $\{\langle \langle 7, 9 \rangle, \langle 20, 23 \rangle \rangle\}$ ； $\langle \langle 8, 10 \rangle, \langle 20, 23 \rangle \rangle \text{UNION}_T \langle \langle 18, 22 \rangle, \langle 27, 33 \rangle \rangle$ 返回 $\{\langle \langle 8, 10 \rangle, \langle 20, 22 \rangle \rangle, \langle \langle 20, 22 \rangle, \langle 27, 33 \rangle \rangle\}$ ； $\langle \langle 8, 10 \rangle, \langle 12, 15 \rangle \rangle \text{UNION}_T \langle \langle 16, 18 \rangle, \langle 20, 23 \rangle \rangle$ 返回 $\{\langle \langle 8, 10 \rangle, \langle 12, 15 \rangle \rangle, \langle \langle 16, 18 \rangle, \langle 20, 23 \rangle \rangle\}$ 。

4.3 时态 NOT_T 运算符

NOT_T(t₁) 运算：返回的时态元素是时态元素 t₁ 在 $\langle 0, \infty \rangle$ 上的可能时态补。这里分为两种情况：

1) 时间点补集，其结果是时区的集合

令 $tp_1 = \langle ps_1, pf_1 \rangle$ ，则

$$\text{NOT}_T(tp_1) = \{ti_1, ti_2\} = \{ \langle is_1, if_1 \rangle, \langle is_2, if_2 \rangle \} = \{ \langle \langle 0, 0 \rangle, \langle ps_1, pf_1 \rangle \rangle, \langle \langle ps_1, pf_1 \rangle, \langle \infty, \infty \rangle \rangle \}$$

且满足约束 CP： $tl(\langle if_1, is_2 \rangle) = 0$

其中结果集合中第二个时区的开始时间点和结束点两者在实际中间隔了操作数时间点 $\langle 12, 16 \rangle$ 所确定的那个时间点，所以满足约束 CP。

2) 时区补集，其结果是时区的集合。令

$ti_1 = \langle is_1, if_1 \rangle = \langle \langle iss_1, isf_1 \rangle, \langle ifs_1, iff_1 \rangle \rangle$ ，则

$$\text{NOT}_T(ti_1) = \{ti_2, ti_3\} = \{ \langle is_2, if_2 \rangle, \langle is_3, if_3 \rangle \} = \{ \langle \langle 0, 0 \rangle, \langle iss_1, isf_1 \rangle \rangle, \langle \langle ifs_1, iff_1 \rangle, \langle \infty, \infty \rangle \rangle \}$$

且满足约束 CP： $tl(\langle if_2, is_3 \rangle) = tl(ti_1) = tl(\langle is_1, if_1 \rangle)$

4.4 相关定理

在上述不确定时态表示方法的基础上进行的运算符扩展，满足以下定理。

定理 1 时态 OVERLAP_T, UNION_T, NOT_T 运算符在上述不确定时态表示方法下是封闭的。

证明：由于运算符 OVERLAP_T, UNION_T, NOT_T 会导致输出结果产生空集和空时态元素（即事件不发生）的情况，而我们在上面的定义中已经包含了空时态元素（空时间点和空时区）的表述。再从上面的时态运算的各项扩展定义可知，当用上述方法表示的不确定时态元素作为时态运算符（OVER-

LAP_T, UNION_T, NOT_T）的操作数时，所得到的结果一定能用上述表示方法中的时间点和时区所组成的集合来表示。所以这三类时态运算符在上述不确定时态表示方法下是封闭的。

定理 2 适用上述不确定时态表示方法扩展的 OVERLAP_T, UNION_T, NOT_T 运算满足布尔运算的基本特性（迪莫根法则，分配率，结合率，交换率）。

1) 迪莫根法则

$$\text{NOT}_T(t_1 \text{OVERLAP}_T t_2) = (\text{NOT}_T(t_1)) \text{UNION}_T (\text{NOT}_T(t_2))$$

$$\text{NOT}_T(t_1 \text{UNION}_T t_2) = (\text{NOT}_T(t_1)) \text{OVERLAP}_T (\text{NOT}_T(t_2))$$

2) OVERLAP_T 和 UNION_T 的分配率

$$t_1 \text{OVERLAP}_T (t_2 \text{UNION}_T t_3) = (t_1 \text{OVERLAP}_T t_2) \text{UNION}_T (t_1 \text{OVERLAP}_T t_3)$$

$$t_1 \text{UNION}_T (t_2 \text{OVERLAP}_T t_3) = (t_1 \text{UNION}_T t_2) \text{OVERLAP}_T (t_1 \text{UNION}_T t_3)$$

3) OVERLAP_T 和 UNION_T 的交换率

$$t_1 \text{OVERLAP}_T t_2 = t_2 \text{OVERLAP}_T t_1$$

$$t_1 \text{UNION}_T t_2 = t_2 \text{UNION}_T t_1$$

4) OVERLAP_T 和 UNION_T 的幂等率

$$t_1 = t_1 \text{OVERLAP}_T t_1$$

$$t_1 = t_1 \text{UNION}_T t_1$$

5) OVERLAP_T 和 UNION_T 的结合率

$$(t_1 \text{OVERLAP}_T t_2) \text{OVERLAP}_T t_3 = t_1 \text{OVERLAP}_T (t_2 \text{OVERLAP}_T t_3)$$

$$(t_1 \text{UNION}_T t_2) \text{UNION}_T t_3 = t_1 \text{UNION}_T (t_2 \text{UNION}_T t_3)$$

6) 双重否定

$$t_1 = \text{NOT}_T(\text{NOT}_T(t_1))$$

证明：由于上述几种时态运算在该表示方法中是封闭的，因此上述表达式的证明，只需要通过验证等式两边的时态元素集相同或者可能的时态元素集相同即可，这是不难得到的。限于篇幅，详细的证明过程略。

小结 本文讨论了时态数据库中处理不确定时间信息所存在的一些问题，并给出一种不确定时态信息的表示方法，在此基础上扩展了时态运算操作。当在时态数据库中用上述方法表示不确定时态信息时，扩展的时态运算能帮助用户获得与不确定时态查询相应的结果，从而丰富了时态数据库的应用范围。有关该不确定时态信息表示模型的更详细信息，在另文阐述。

参考文献

- Allen J F. Maintaining knowledge about temporal intervals. Communications of the ACM, 1983, 26(11): 832~843
- Vilain M, Kautz H. Constraint propagation algorithms for temporal reasoning. In: Kehler T, Rosenschein S et al. eds. Proc. of the 5th National Conference on Artificial Intelligence. SanMateo, CA: Morgan Kaufmann Publishers, Inc, 1986. 377~382
- Schwlb E, Dechter R. Processing disjunctions in temporal constraint networks. Artificial Intelligence, 1997, 93(1): 29~61
- Meiri I. Combining qualitative and quantitative constraints in temporal reasoning. Artificial Intelligence, 1996, 87(2): 343~385
- 黄楠, 刘爱琴. 时态数据库技术. 微机发展, 2002, 01: 24~26
- 李浩明. 基于不确定时间间隔的索引技术研究: [硕士学位论文]. 广州: 华南理工大学, 2004