基于特定关系模式下函数依赖集的闭包的研究

肖治军，　彭小宁*

（怀化学院 计算机科学与技术系，湖南 怀化　418008）

摘　要：通过实例研究，定义了一个特定的关系模式，并称它为传递依赖关系模式．在传递依赖关系模式下，得出了计算函数依赖集的闭包中函数依赖的个数的公式，并依照具体实例设计了计算函数依赖集的闭包中所有函数依赖的算法，在此算法中，设计了多种计算属性集所有子集的算法．

关键词：函数依赖集的闭包；　Armstrong 公理；　子集

中图分类号：TP301　　**文献标识码**：A　　**文章编号**：1671－9743（2012）05－0027－04

0　引言

在关系数据库理论中，计算函数依赖集的闭包是一个相当复杂且困难的问题．为了解决这个问题，人们把计算函数依赖集的闭包简化为计算属性集的闭包．然而，此种方法只是从计算属性集的闭包的角度局部研究了函数依赖集的闭包．本文通过研究参考文献［1］中的实例，引申出一个特定关系模式，并把它定义为传递依赖关系模式．在传递依赖关系模式下，计算函数依赖集的闭包变得有方法可依，并且可以从计算函数依赖集的闭包的角度整体研究函数依赖集的闭包．

1　传递依赖关系模式的定义

我们先来看参考文献［1］的一个实例．

例 1　设有关系模式 R（X，Y，Z）与它的函数依赖集 F＝｛X→Y，Y→Z｝，求函数依赖集的闭包 F^+．

解：根据函数依赖的推理规则，可推出 F 的闭包 F^+ 有 43 个函数依赖，它们如图 1 所示．

$$F^+ = \begin{cases} X\to\varnothing & XY\to\varnothing & XZ\to\varnothing & XYZ\to\varnothing & Y\to\varnothing & YZ\to\varnothing & Z\to\varnothing & \varnothing\to\varnothing \\ X\to X & XY\to X & XZ\to X & XYZ\to X & & & & \\ X\to Y & XY\to Y & XZ\to Y & XYZ\to Y & Y\to Y & YZ\to Y & & \\ X\to Z & XY\to Z & XZ\to Z & XYZ\to Z & Y\to Z & YZ\to Z & Z\to Z & \\ X\to XY & XY\to XY & XZ\to XY & XYZ\to XY & & & & \\ X\to XZ & XY\to XZ & XZ\to XZ & XYZ\to XZ & & & & \\ X\to YZ & XY\to YZ & XZ\to YZ & XYZ\to YZ & Y\to YZ & YZ\to YZ & & \\ X\to XYZ & XY\to XYZ & XZ\to XYZ & XYZ\to XYZ & & & & \end{cases}$$

图 1

在例 1 中，函数依赖集 F 中涉及到关系模式 R 中的属性只有 3 个，我们把它进行推广，引申出特定的关系模式．

定义 1　设有关系模式 R（U，F），U 是关系模式 R 的属性集，F 是 R 上成立的只涉及到 U 中属性的函数依赖集．如果在关系模式 R（U，F）中，U＝｛X_1，X_2，X_3，…，X_n｝（$n\in N^+$），F＝｛$X_1\to\varnothing$｝（n＝1），F＝｛$X_i\to X_{i+1}$｝（i＝1，2，3，…，n－1 且 $n\geqslant 2$），我们

收稿日期：2012－03－26

基金项目：怀化学院重点学科建设项目资助．

作者简介：肖治军（1991－），男，怀化学院计算机系 2009 级学生．

＊通讯作者：彭小宁（1962－），男，湖南长沙人，怀化学院教授，主要研究领域数字图像处理与信息安全技术、数据库技术等．

就称关系模式 R（U，F）为传递依赖关系模式，并把它记作 T（U，F）.

2 函数依赖集的闭包中函数依赖的个数

定义 2 设 F 是在关系模式 R（U）上成立的函数依赖集合，X，Y 是属性集 U 的子集，X→Y 是一个函数依赖 . 如果从 F 中能够推导出 X→Y，即如果对于 R 的每个满足 F 的关系 r 也满足 X→Y，则称 X→Y 为 F 的逻辑蕴涵（或 F 逻辑蕴涵 X→Y），记为 F| = X→Y.

定义 3 设 F 是函数依赖集，被 F 逻辑蕴涵的函数依赖的全体构成的集合，称为函数依赖集 F 的闭包，记为 F^+ . 即：

$F^+ = \{X \to Y | F| = X \to Y\}$

定理 1 在 T（U，F）中，F^+ 中函数依赖的个数是 $\frac{2^{2n+1}+1}{3}$（$n \in N^+$）.

证明： 根据 Armstrong 公理之自反律和传递律，可得 $\{X_i\} \to \{X_j\}$ 和 $\{X_i\} \to \emptyset$，其中 i = 1，2，3，…，n；j = i，i + 1，i + 2，…，n.

当 k = 1 时，令集合 A = U，集合 B = U - $\{X_1\}$；当 k = 2，3，4，…，n 时，令集合 A = U - $\{X_1, X_2, X_3, \cdots, X_{k-1}\}$，集合 B = U - $\{X_1, X_2, X_3, \cdots, X_k\}$.

设 P_i 为集合 A 的任一子集，其中 i = 1，2，3，…，2^{n-k+1}. 根据 Armstrong 公理之合并律，显然有 $\{X_k\} \to P_i$，其中 i = 1，2，3，…，2^{n-k+1}.

设定 Q_j 为集合 B 的任一子集，其中 j = 1，2，3，…，2^{n-k}. 根据 Armstrong 公理之增广律，可得 $\{X_k\} \cup Q_j \to P_i \cup Q_j$，其中 i = 1，2，3，…，$2^{n-k+1}$；j = 1，2，3，…，$2^{n-k}$. 再根据 Armstrong 公理之分解律，可得 $\{X_k\} \cup Q_j \to P_i$，其中 i = 1，2，3，…，2^{n-k+1}；j = 1，2，3，…，2^{n-k}.

最后，显然有 Ø→Ø. 我们猜想 $F^+ = \{\{X_k\} \cup Q_j \to P_i, \emptyset \to \emptyset\}$，其中 i = 1，2，3，…，$2^{n-k+1}$；j = 1，2，3，…，$2^{n-k}$. 下面，我们证明此猜想是正确的 .

令集合 S = $\{\{X_k\} \cup Q_j \to P_i, \emptyset \to \emptyset\}$，其中 i = 1，2，3，…，$2^{n-k+1}$；j = 1，2，3，…，$2^{n-k}$. 设集合 X 和 Y 是属性集 U 的子集，并设 Ø 的下标为 n + 1.

对于 $\forall X \to Y \in S$，根据定义 3，显然 $\forall X \to Y \in F^+$. 因此，$S \subseteq F^+$.

对于 $\forall X \to Y \in F^+$，当 X 是 Ø 时，显然，只有 Ø→Ø 成立，且 Ø→Ø∈S. 因此，当 X 是 Ø 时，$F^+ \subseteq S$；当 X 不是 Ø 时，我们先把集合 X 拆分成单个元素 X_{min} 和集合 T 两部分，其中，X_{min} 代表 X 中下标最小的元素，T = X - X_{min}. 依据函数依赖集 F 中各个属性之间的函数依赖关系，我们可以得出 T 中所有元素的下标都大于 min，集合 Y 中所有元素下标都大于等于 min. 因此，当 k = min 时，T 必存在于 Q_j 中，而 Y 也必存在于 P_i 中，其中 i = 1，2，3，…，2^{n-k+1}；j = 1，2，3，…，2^{n-k}. 那么，当 X 不是 Ø 时，显然有 $\forall X \to Y \in S$，则 $F^+ \subseteq S$. 因此，对于 $\forall X \to Y \in F^+$，有 $F^+ \subseteq S$.

由上可得 $F^+ = S$，即 $F^+ = \{\{X_k\} \cup Q_j \to P_i, \emptyset \to \emptyset\}$，其中 i = 1，2，3，…，$2^{n-k+1}$；j = 1，2，3，…，$2^{n-k}$. 因此，猜想是正确的 .

综上所述，在 T（U，F）中，F^+ 中函数依赖的个数是：

$$\sum_{k=1}^{n} 2^{2n-2k+1} + 1 = \frac{2^{2n+1}+1}{3} (n \in N^+)$$

3 属性集所有子集的求解算法

在传递依赖关系模式 T（U，F）下，我们来设计求解属性集 U 的所有子集的算法 . 注意：如果没有特殊说明，本文所有算法中集合元素的相对顺序不能改变 . 本文也规定：集合中任何元素与 Ø 组合仍为元素本身 .

算法 1 下面是算法 1 的详细步骤 .

（1）创建集合 A 和集合 R，并令集合 A = U = $\{X_1, X_2, X_3, \cdots, X_n\}$，集合 R 为 Ø.

（2）建立一张 2^n 行，n + 1 列的表格 T. 表格 T 的第 0 行称为表头，第 1 ~ 2^n - 1 行称为表体 . 表格 T 的具体描述如下：

①在表头中，所有单元格填入的都是表格的列名 . 在表格 T 中，第 0，1，2，…，n - 1 列的列名分别是 X^1，X^2，X^3，…，X^n；第 n 列的列名是属性集 .

②在表格 T 的第 1 ~ 2^n - 1 行、第 0 ~ n - 1 列这块区域中，每行的第 0，1，2，…，n - 1 个单元格依次填入此行行号所对应的 n 位二进制数的第 n - 1，…，2，1，0 位 .

③在表体的每行中，从左至右将二进制位为 1 的

单元格所在列的列名填入属性集列的单元格中．

(3) 令 count = 1.

(4) 如果 count = n + 1，将 Ø 加入到集合 R 的首位，然后输出集合 R，算法结束；否则，转 (5)．

(5) 依次扫描属性集列的第 2^n-1，2^n-2，2^n-3，…，1 个单元格，如果某一个单元格中的列名个数等于 count，则将此单元格中的所有列名作为集合 R 的一个元素加入到集合 R 的末尾．

(6) count = count + 1，转 (4)．

算法 2　下面是算法 2 的详细步骤．

(1) 创建集合 S，并令集合 S 为 Ø.

(2) 创建一个与算法 1 一样的表格 T，表格 T 增加第 n + 1 列．第 n + 1 列的列名是起始号．在表体的每行中，把首个二进制位为 1 的单元格所在列的列号填入起始号列的单元格中．

(3) 令 column = 1.

(4) 令 count = 1.

(5) 如果 count = n − column + 2，转 (8)，否则，转 (6)．

(6) 依次扫描表格 T 的第 2^n-1，2^n-2，2^n-3，…，1 行．如果在某行中，属性值列的单元格中列名的个数等于 count，并且起始号列的单元格中的列号等于 column，则将此行的属性集列的单元格中的所有列名作为集合 S 的一个元素加入到集合 S 的末尾．

(7) count = count + 1，转 (5)．

(8) column = column + 1，如果 column = n + 1，将 Ø 加入到集合 S 的末尾，然后输出集合 S，算法结束；否则，转 (4)．

算法 3　下面是算法 3 的详细步骤．

(1) 创建集合 R，并使 R = U = $\{X_1, X_2, X_3, \cdots, X_n\}$．集合 R 中每一个元素用自身下标进行标志．如果 n = 1，则将集合 R = $\{Ø, X_1\}$ 输出，算法结束；否则，令 i = 1，转 (2)．

(2) 如果集合 R 中第 i 个元素的标志 (记为 x) 小于 n，转 (3)；否则，转 (7)．

(3) 令 j = x + 1.

(4) 在集合 R 中，第 i 个元素与第 j 个元素组合成标志为 j 的新元素 N，并将 N 加入到集合 R 的末尾．

(5) j = j + 1. 如果 j = n + 1，转 (6)；否则，转 (4)．

(6) 如果集合 R 中第 i 个元素的属性个数和标志都等于 n − 1，则将 Ø 加入到集合 R 的首位，同时输出集合 R，算法结束；否则，转 (7)．

(7) i = i + 1. 转 (2)．

算法 4　下面是算法 4 的详细步骤．

(1) 创建集合 S，并使集合 S = U = $\{X_1, X_2, X_3, \cdots, X_n\}$．令 i = 1.

(2) 如果 i 小于 n，就删除集合 S 中第 1 个元素 (记为 X)，然后利用算法 3 计算出集合 S 的所有子集的集合 A，转 (3)；否则，先将集合 S 的第 1 个元素放到集合 S 的末尾，再将 Ø 加入到集合 S 的末尾，接着，输出集合 S，算法结束．

(3) 集合 A 中的每一个元素都与 X 组合起来，从而形成新的集合 A，然后将集合 A 中所有元素作为一个整体加入到集合 S 的末尾．

(4) i = i + 1，转 (2)．

4　函数依赖集的闭包的求解算法

算法 5　下面是算法 5 的详细步骤．

(1) 分别依据算法 1 和 2 (或算法 3 和 4)，计算出属性集 U 的子集集合 R 和 S. 创建集合 W，并令集合 W 为 Ø.

(2) 对集合 R 和 S 中的元素进行标志，标志即为集合元素中最小的属性下标．特别地，Ø 的标志为 n + 1.

(3) 从集合 R 的第 1 个元素起，依次将集合 R 中的单个元素 (记为 r) 与集合 S 中的所有元素按照从左到右的顺序进行比较．如果集合 S 中某一元素 (记为 s) 的标志小于等于 r 的标志，则将函数依赖 s→r 加入到集合 W 中．

(4) 集合 W 即为 F^+，输出集合 W，算法结束．

显然，依据算法 5，我们可以很容易求解例 1 了．

5　算法实例演示

本文共设计了 5 个算法，限于篇幅，我们不能对每个算法都进行实例演示．本文只对算法 1 进行实例演示．

依据算法 1，计算属性集 U = {X, Y, Z} 的子集

集合 R，具体步骤如下．

（1）依据算法 1 的第（1）和（2）步，可以创建表 1．

表 1

X	Y	Z	属性集
0	0	1	Z
0	1	0	Y
0	1	1	YZ
1	0	0	X
1	0	1	XZ
1	1	0	XY
1	1	1	XYZ

（2）依据算法 1，只要自底向上扫描表 1 的属性集列即可．第一次扫描，如果属性集列的单元格中列名的个数为 1，则将此单元格中的所有列名作为集合 R 的一个元素加入到集合 R 的末尾，由此可得 R = {X, Y, Z}；第二次扫描，如果属性集列的单元格中列名的个数为 2，则将此单元格中的所有列名作为集合 R 的一个元素加入到集合 R 的末尾，由此可得 R = {X, Y, Z, XY, XZ, YZ}；第三次扫描，如果属性集列的单元格中列名的个数为 3，则将此单元格中的所有列名作为集合 R 的一个元素加入到集合 R 的末尾，由此可得 R = {X, Y, Z, XY, XZ, YZ, XYZ}．

（3）将 Ø 加入到集合 R 的首位，可得属性集 U 的子集集合 R = {Ø, X, Y, Z, XY, XZ, YZ, XYZ}．

6 结语

本文通过对参考文献［1］中实例的深入研究，发掘出传递依赖关系模式，传递依赖关系模式是一种特殊的关系模式．在传递依赖关系模式下，我们可以利用 Armstrong 公理去全面研究函数依赖集的闭包，而不是通过研究属性集的闭包去局部研究函数依赖集的闭包．我们从本文的研究结果中可以看出，对于一般的关系模式，通过研究属性集的闭包去研究函数依赖集的闭包是十分明智的．

参考文献：

[1] 陈志泊，王春玲．数据库原理及应用教程（第二版）[M]．北京：人民邮电出版社，2008：140－148．

[2] 王珊，萨师煊．数据库管理系统概论（第四版）[M]．北京：高等教育出版社，2006：169－173．

[3] Thomas Connolly, Carolyn Begg．数据库系统—设计、实现与管理（第四版）[M]．北京：电子工业出版社，2008：285－290．

Study on the Closure of Functional Dependency Set Based on a Specific Relational Schema

XIAO Zhi－jun, PENG Xiao－ning*

(*Department of Computer Science and Technology, Huaihua University, Huaihua, Hunan 418008*)

Abstract: This paper defines a specific relational schema that is called transitive dependency relational schema, according to studying a case. Based on the transitive dependency relational schema, the paper proves a formula for computing the number of functional dependencies in the closure of functional dependency set and designs an algorithm for computing all functional dependencies in the closure of functional dependency set. In the algorithm, the paper also designs some algorithms for computing all subsets of attribute set.

Key words: closure of functional dependency set; Armstrong's axioms; subset