

Deduction Graphs Based-on Functional Dependency Structure

Ji Yue Yao Qingda Dong Zuming

Computer Software Institute Lingnan (University) College

Abstract

A deduction graph (DG) for representing the structural relationship between functional dependencies(FD) or functional-free Horn formula (extended from Horn classes) in a relational database or rule-based expert system is defined. In addition to graphically solving the inference problem by DG's, many logic queries can be answered by DG's with substitutions for unifying expressions. This graphic approach provides an alternative method for conducting inferences and processing logic queries, which is neither forward-chaining-oriented nor backward-chaining-directed using the inference rule of resolution and has many advantages to execute a logic program, These advantages are achieved because a DG is never infinite and constructing a DG is independent from the computation and search rules and also independent from the order of the given FD's or HC's.

Keywords deduction graph, functional dependency, Horn clause, inference rule, relational database

1. Introduction

It is known that a relational database (RDB) [5] corresponds to a logic program consisting solely of ground unit clauses [11] as the individual tuples in the RDB. It is also known that performing inferences or logical deductions has been the fundamental unit of computation, since the only interest in theorem proving by the resolution-refutation technique is to demonstrate a logical consequence of a logic program and a goal clause [11].

We alternatively solve the important inference problem by a new graphic technique that does not rely on the resolution-refutation principle as commonly used in a backward-chaining control system. Inferences accomplished by means of the proposed graphic method play a central role in RDB's, rule-based expert systems (RBES), logic programming (LP), and artificial intelligence (AI). Although the inference

Received 10 January, 1990

problem for producing a new functional dependency (FD) or function-free Horn clause (HC) from existing ones can be also solved by other methods, such as truth tables [4], [16], derivation sequences [16], logical equivalence [14], [16], inference rules including resolution [11], [16].

The graphic representations of FD's, which play an important role in inferences, are of various types including trees, such as Bernstein's S-based derivation tree for a given S of FD's [2], [3]; directed acyclic graphs (DAG), such as Maier's S-based DAG [12]; the FD-graph of Ausiello et al. [1]; Date's FD-diagram[6]; Martin's bubble chart [13]; Smith's dependency diagram [15]; Yang's deduction graph

Here we propose deduction graphs (DG) based-on Functional Dependency Structure (FDS)[18].

2. Preliminaries

We briefly review relevant terminology and notations conventionally used in the areas of RDB's and the HC logic.

Let U be a non-empty set of attributes, which is called a relation scheme and conventionally represented by a string of its attributes. When X and Y are subsets of U, the union of X and Y is conventionally represented by short-hand string notation XY.

A functional dependency (FD) of the term $X \rightarrow A$ for X being a subset of U and A being an attribute in U is a constraint on (a subset of) U such that a relation $r(U)$ satisfies the FD $X \rightarrow A$ if an identical X-value of every two tuples in $r(U)$ implies the same A-value. It is well known that $X \rightarrow A$ is equivalent to the formula $(\neg X \vee A)$ [14] where the symbols " \neg " and " \vee " stand for "negation" and "disjunction", respectively. In this formula, A is viewed as an atom or more specifically, the predicate $A(s, t)$ is an atom whose value is true if $t_1(A) = t_2(A)$ and is false otherwise during an interpretation of using two tuples t_1 and t_2 in $r(U)$ to substitute the tuple variables s and t in A. Each attribute in X is an atom. Formula X is the conjunction of the atoms in X. Let "conjunction" be denoted by the symbol " \wedge ". Thus, if $X = B_1 \dots B_k$, where $B_1 \dots B_k$ is the short hand string notation of the formula $(B_1 \wedge \dots \wedge B_k)$, then $(\neg X \vee A)$ can be written as

$$(\forall s)(\forall t)(A(s,t) \vee \dots \vee \neg B_1(s,t) \vee \dots \vee \neg B_k(s,t)) \tag{1a}$$

where the arguments s and t have scheme U and are called tuples variables s and t, (1a) can be understood as

$$A \vee \neg B_1 \vee \dots \vee \neg B_k \tag{1b}$$

which is called a headed Horn clause [3] and is equivalent to the following form as conventionally represented in LP

$$A:-B_1,\dots,B_k \quad (1c)$$

In (1c), the symbol “:-” stands for “if”, commas mean conjunctions, the left side of :- is the head, and the right side is the body. If no body or no head occurs in (1c), the clause is, respectively, called a unit clause or a goal clause. If the arguments of a unit clause are all constants, the clause is ground and called an assertion or a fact corresponding to a tuple in a RDB.

For defining a sound and complete set of inference rules for the Horn logic as similar to that for FD's [16], we need to extend HC's to Horn formulas (HF). An HF is defined as follows:

- 1) Each HC is an HF.
- 2) If $A:-X$ and $B:-X$ with unifiable X 's [11] are HC's, then $A, B:-X$ is an HF.

Let X, Y, Z be formulas that are conjunctions of atoms. With the above extension, We have inference rules for HF's.

- 1) (Reflexivity): If Y is a subformula of X , then $Y:-X$.
- 2) (Transitivity): If $Y:-X$ and $Z:-Y$ with unifiable Y 's, then $Z:-X$.
- 3) (Conjunction): If $Y:-X$ and $Z:-X$ with unifiable X 's then $Y, Z:-X$.

These rules for function-free HF's are sound and complete, since so are the counterparts for FD's as proven by Armstrong [1], [16].

3. Functional Dependency Structure and Deduction Graph Based-on FDS

In the following discussion, we make use of the relational scheme $R(U, F)$ to show our ideas, where attribute universe $U = ABCDEFGH$, and functional dependency set $F = \{ E \rightarrow H, F \rightarrow BC, C \rightarrow D, BD \rightarrow G, G \rightarrow A, A \rightarrow BC \}$.

Definition 3.1. Given FD set F , we define a set composed of all the left hand set of dependencies in F . This is called Left-Hand-Side of F , denoted by $LHS(F) = \{ X | f: X \rightarrow Y \text{ in } F \}$. Similarly, we define Right-Hand-Side of F as $RHS(F) = \{ Y | f: X \rightarrow Y \text{ in } F \}$.

Definition 3.2. Given FD set F , and $f: X \rightarrow Y, g: U \rightarrow V$ are in F , then

- (1) If $F \models X \rightarrow U$ and $F \models U \rightarrow X$, we say f Strongly-Connects g , denoted by SCR (f, g),
- (2) If $F \models X \rightarrow U$ but $F \not\models U \rightarrow X$, we say f Weakly-Connects g , denoted by WCR (f, g),

(3) If $F \models X \rightarrow U$ and $F \models U \rightarrow X$, we say f Not-Connects g , denoted by $NCR(f, g)$.

Proposition 3.1. Given F and $f: X \rightarrow Y, g: U \rightarrow V$ are in F , thus

- (1) $SCR(f, g)$ holds, iff $X_F^+ = Y_F^+$,
- (2) $WCR(f, g)$ holds, iff $X_F^+ \supseteq Y_F^+$,
- (3) $NCR(f, g)$ holds, iff $X_F^+ - Y_F^+ \neq \phi$ and $Y_F^+ - X_F^+ \neq \phi$.

Example 3.1. For the given F , we have the following:

- $(E)_F^+ = EH,$
- $(F)_F^+ = FBCDGA,$
- $(C)_F^+ = CD,$
- $(BD)_F^+ = BDGAC,$
- $(G)_F^+ = GABCD,$
- $(A)_F^+ = ABCDG,$ So we have
- $SCR(BD \rightarrow G, G \rightarrow A),$
- $WCR(F \rightarrow BC, G \rightarrow G),$
- $NCR(E \rightarrow H, C \rightarrow D).$

Theorem 3.1. Given FD set F , let $f: X \rightarrow Y, g: U \rightarrow V, h: S \rightarrow T$ are in F , then we have the following:

- (1) For any f , $SCR(f, f)$ holds,
- (2) If $SCR(f, g)$ holds, then $SCR(g, f)$ holds,
- (3) If $SCR(f, g)$ and $SCR(g, h)$ holds, then $SCR(f, h)$ holds.

Definition 3.3. Given FD set F , we say I is a Strongly-Connecting-Class (SCC) of F , if for any $f: X \rightarrow Y$ and $g: U \rightarrow V$ in I , also $SCR(f, g), F \models X \rightarrow U$ and $F \models U \rightarrow X$ hold.

Proposition 3.2. Given FD set F , then $F = \bigcup_{i=1}^p I_i$, and for $i \neq j$ $I_i \cap I_j = \phi$ where I_i is a SCC of F , we denote $\{I_i\}$ by F/SCR .

Example 3.2. For the given F , we have four SCCs as follows:

- $I_1 = \{E \rightarrow H\},$
- $I_2 = \{F \rightarrow BC\},$
- $I_3 = \{C \rightarrow D\},$
- $I_4 = \{BD \rightarrow G, G \rightarrow A, A \rightarrow BC\}.$

Definition 3.4. Given FD set F , let I, J be in F/SCR . Suppose f in I and g in J . If $WCR(f, g)$, then we say I Weakly-Connects J , denoted by $WC(I, J)$.

Proposition 3.3. Given FD set F , let I, J in F/SCR , then $WC(I, J)$

holds, iff there exist f in I and g in J such that $WC(f, g)$ holds.

Theorem 3.2. Given FD set F , let I, J, K all in F/SCR , then we have:

- (1) For any I , $WC(I, I)$ holds,
- (2) If $WC(I, J)$ and $WC(J, I)$ hold, then $I=J$,
- (3) If $WC(I, J)$ and $WC(J, K)$ hold, then $WC(I, K)$.

So far, we obtain a partially-ordered set $\langle F/SCR, WC \rangle$ for the given FD set F , which is called the Functional Dependency Structure (FDS) for F .

Example 3.3. For the given F , its FDS can be shown as follows:

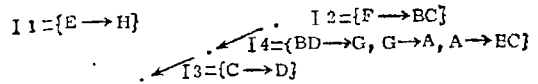


Figure 1. The FDS for the given F

Definition 3.5. For the given FD set F , its deduction graph $\langle N, E \rangle$ is defined as follows:

$N = LHS(F) + RHS(F)$,

E is a set of $N \times N = \{ (n_1, n_2) \}$, which is constructed as follows:

- (1) for each FD: $X \rightarrow Y$, then (X, Y) is in E ,
- (2) for each SCCs I with $LHS(I) = \{ X_1, X_2, \dots, X_k \}$, then $(X_1, X_2), (X_2, X_3), \dots, (X_k, X_1)$,
- (3) for SCCs I and J satisfying $WC(I, J)$ and there exists no other SCC K such that $WC(I, K)$ and $WC(K, J)$, let X_i be in $LHS(I)$ and X_j be in $LHS(J)$, then (X_i, X_j) is in E .

Example 3.4. For the given F , its deduction graph based-on FDS is shown as follows:

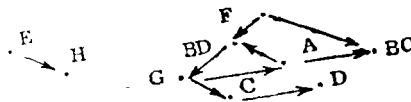


Figure 2. The deduction graph based-on FDS for F

4. Conclusions

FDS has been put into the design of database schemes, computation of candidate keys, etc. The introduction of deduction graph based-on FDS will contribute to the implementation of this ideas. DG proposed here can also be applied to HF in Logic Programming and Expert Sys-

tems. The properties about DG based-on FDS need to be investigated further.

References

- [1] Ausiello G, *J. ACM*, 30(1983), 752~766
- [2] Beeri C, *ACM TODS*, 4(1979), 30~59
- [3] Bernstein P A, *ACM TODS*, 1(1976), 277~298
- [4] Chang C L, *Symbolic Logic and Mechanical Theorem Proving*, New York: Academic, 1973
- [5] Codd E F *CACM*, 13(1970), 6, 377~387
- [6] Date C J, *An Introduction to Database Systems*, 4th ed. Reading, MA: Addison-Wesley, 1985
- [7] Fagin R, *Proc. ACM SIGMOD Conf*, 1979, 153~160
- [8] Fagin R, *J. ACM*, 29(1982), 3, 952~985.
- [9] Hayes-Roth F, *CACM*, 28(1985), 9, 921~932
- [10] Ling T W, *ACM TODS.*, 6(1985), 2, 329~346
- [11] Liloyd J W, *Foundations of Logic Programming*, New York:Springer-Verlag, 1984
- [12] Maier D, *J. ACM*, 27(1980), 4, 664~674
- [13] Marti J, *Computer Data-Base Organization*, 2nd ed. Englewood Cliffs, NJ:Prentice-Hall, 1977
- [14] Sagiv Y, *J. ACM*, 28(1981), 3, 435~453
- [15] Smith H C, *CACM*, 28(1981), 3, 826~838
- [16] Yang C C, *Relational Databases*. Englewood Cliffs, NJ: Prentice-Hall, 1986
- [17] Yang C C, *IEEE TOSE*, 15(1989), 1, 60~67
- [18] Ji Yue, *J.Huazhong Univ. of Sci. & Tech.* 17(1989), 3, 71~76

函数依赖结构与演绎图

纪 岳* 姚卿达 董祖明

摘 要

本文论证了函数依赖与无函数Horn公式之间的等价性,并提出了基于函数依赖结构的演绎图,为逻辑查询及推理处理提供了一个新方法,它与运用规则的次序无关,因而具有很多相应的好处.

关键词 演绎图, 函数依赖, Horn子句, 推理规则, 关系数据库

• 计算机软件研究所