A Top-down Inference Procedure for Template Dependencies

L. J. HENSCHEN* AND B. P. WEEMS†

*Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL 60201, USA

†Department of Computer Science Engineering, The University of Texas at Arlington, P.O. Box 19015, Arlington, TX 76019, USA

The chase procedure, for determining if a given dependency must hold in any relation where a set of dependencies is known to hold, operates bottom-up by constructing a hypothetical relation known as a tableau. A top-down counterpart, based on resolution theorem proving techniques, is introduced in which, through factoring, resolvents which contain function terms may be avoided. A literal numbering technique, which limits the number of resolvents, is also examined.

Received June 1986

INTRODUCTION

In the dependency inference problem for relational databases, one must determine if a given dependency must hold in any relation where a set of dependencies is known to hold. The chase procedure[1] operates bottom-up by constructing a hypothetical relation known as a *tableau*. This notes examines a top-down counterpart based on resolution theorem-proving techniques.[3] Through factoring, resolvents which contain function terms may be avoided. A literal numbering technique which limits the number of resolvents is also examined. The procedure is not intended for the entire class of template dependencies, since the inference problem is known to be only semi-decidable,[8] but this procedure can be applied to various subclasses of the template dependencies. In some experiments involving embedded multivalued dependencies, this procedure finds a proof much faster than the chase.[1] In one instance, our procedure found a proof in less than a minute (VAX 11/780) when the chase had not succeeded in over three hours. We assume familiarity with relational databases as described in Ref. 7 and automated deduction methods as described in Ref. 3.

Top-Down Inclusion Dependency Inference Procedure

Before examining template dependencies, the simpler case of inclusion dependencies is considered.[2]

Definition: An *inclusion dependency*, $R[x_1 \dots x_n] \subseteq S[y_1 \dots y_n]$, where R and S are relations and $x_1 \dots x_n$ and $y_1 \dots y_n$ are sequences of attributes from R and S, respectively, is a logical statement indicating that for each tuple in R there exists a tuple in S such that the value for y_j is identical to the value for x_j for $1 \leqslant j \leqslant n$.

Example Inclusion Dependency

$R[ABC] \subseteq S[DEF]$

states that for any tuple in R there must also be a tuple in S whose D-value is the same as the A-value, whose E-value is the same as the B-value, and whose F-value is the same as the C-value.

The chase may be used for inclusion dependencies. As an example, assume a database has relations $R(A,B,C)$, $S(D,E,F)$ and $T(G,H,I)$. Also assume that the inclusion dependencies $R[AC] \subseteq S[DE]$, $S[DF] \subseteq T[GH]$, and $T[G] \subseteq R[C]$ are known to hold. $R[A] \subseteq R[C]$ is to be proven to also hold. Initially, the only tuple is $R(a,b,c)$. By $R[AC] \subseteq S[DE]$, we must also have tuple $S(a,c,d)$, where d is a value arbitrarily chosen (it cannot be chosen again). Similarly, by $S[DF] \subseteq T[GH]$, there must be a tuple $T(a,d,e)$, where e is a value arbitrarily chosen. Finally, by $T[G] \subseteq R[C]$, there must be a tuple $R(f,g,a)$, but this last *required* tuple proves that $R[A] \subseteq R[C]$ must also hold.

The described chase operates *forward*, that is, it works from elementary facts towards the goal, in this case the tuple which must be present. The top-down counterpart operates *backward*, that is, it works from the goal towards the elementary facts. For the same instance as above, we start with a goal of $R(x,y,a)$ (x and y are variables and unify with any term). From $T[G] \subseteq R[C]$, this goal can be satisfied if $T(a,w,z)$ can be satisfied. Further, from $S[DF] \subseteq T[GH]$, $T(a,w,z)$ can be satisfied if $S(a,v,w)$ can be satisfied. Finally, from $R[AC] \subseteq S[DE]$, $S(a,v,w)$ can be satisfied if $R(a,u,v)$ can be satisfied. But since we are testing $R[A] \subseteq R[C]$ and we are assuming a tuple $R(a,b,c)$, this last goal is satisfied.

Top-Down Template Dependency Inference Procedure

Definition: A (typed) *template dependency* for a relation R is represented as:

$$\begin{array}{c} h_{11} \dots h_{1n} \\ \vdots \quad\; \vdots \\ h_{m1} \dots h_{mn} \\ \hline c_1 \dots c_n \end{array}$$

where $h_{ij} \neq h_{kl}$ *if* $j \neq l$, $c_i \neq h_{jk}$ *if* $i \neq k$, and $c_i \neq c_j$ if $i \neq j$. The row $h_{i1} \dots h_{in}$ is an *hypothesis row*. The row $c_1 \dots c_n$ is the *conclusion*. A template dependency is interpreted by considering the symbols to be variables. If the template dependency holds on a relation R with arity n, then if a set of tuples from R can consistently replace the pattern represented in the hypothesis, some tuple in R must satisfy the required conclusion.

A conclusion row symbol which does not appear in the hypothesis is an *embedded position*. A conclusion row symbol which does appear in the hypothesis is a *context position*.

Example Template Dependency

$$\begin{array}{c} A\ B\ C\ D \\ A\ E\ F\ G \\ \hline A\ B\ F\ H \end{array}$$

states that if there are two tuples which agree in the first attribute, then there must also be a tuple with the same value for the first attribute, its second value from the first tuple, and its third value from the second tuple. Note that the fourth position is embedded and is left arbitrary.

The chase can also be applied to template dependencies. Suppose we know that in a relation the following template dependencies must hold:

$$\begin{array}{llllll} (1) & A\,B\,C\,D\,E & (2) & A\,B\,C\,D\,E & (3) & A\,B\,C\,D\,E \\ & \underline{A\,b\,c\,d\,e} & & \underline{a\,B\,c\,d\,e} & & \underline{a\,b\,C\,d\,e} \\ & A\,B\,F\,d\,H & & F\,B\,C\,G\,e & & A\,b\,C\,D\,e \end{array}$$

We would like to prove that:

$$\begin{array}{l} A\,B\,C\,D\,E \\ \underline{A\,b\,c\,d\,e} \\ A\,B\,F\,d\,E \end{array}$$

must also hold. The steps in constructing the tableau are given. Initially, the tableau contains rows which fulfil the pattern in the hypothesis of the dependency to implicate:

(1) $A\,B\,C\,D\,E$

(2) $A\,b\;\;c\;d\,e$

Row 3 is created to fulfil dependency 1, the two hypothesis rows are fulfilled by rows 1 and 2. 1 and 2 are symbols arbitrarily chosen for the embedded positions.

(3) $A\,B\,1\,d\,2$

Row 4 is created to fulfil dependency 2, the two hypothesis rows are fulfilled by rows 3 and 1, in that order. 3 and 4 are, again, arbitrarily chosen for the embedded positions.

(4) $3\,B\,1\,4\,E$

Row 5 is created to fulfil dependency 3, the two hypothesis rows are fulfilled by rows 3 and 4.

(5) $A\,B\,1\,d\,E$

Row 5 matches the conclusion of the template dependency to be proven in its context positions, thus implying that it must also hold.

A top-down proof without function terms is induced by the chase proof. We start with the conclusion of the template dependency to be proven as our goal. For sake of readability, '_' represents a distinct variable which appears only once in the clause (i.e. like Prolog). From the hypothesis of the dependency to be proven, we know the goals ($A\;B\;C\;D\;E$) and ($A\;b\;c\;d\;e$) can be satisfied.

$(A\,B\,\_\,d\,E)$

This goal is satisfied if the following goal derived from applying dependency 3 can be satisfied:

$(A\,\_\,X\,d\,\_)\;\;(\_\,B\,X\,\_\,E)$

One possibility for this new goal being satisfied is to derive the following goal from applying dependency 2 to the second subgoal:

$(A\,\_\,X\,d\,\_)\;\;(\_\,B\,X\,\_\,\_)\;\;(\_\,B\,\_\,\_\,E)$

This new goal may be satisfied if the first two subgoals are satisfied by the same dependency, which can be represented by *factoring*:

$(A\,B\,X\,d\,\_)\;\;(\_\,B\,\_\,\_\,E)$

This goal may be satisfied if the goal resulting from applying dependency 1 to the first subgoal can be satisfied:

$(A\,B\,\_\,\_\,\_)\;\;(A\,\_\,\_\,d\,\_)\;\;(\_\,B\,\_\,\_\,E)$

Once again, we may factor, this time the first and third subgoals:

$(A\,B\,\_\,\_\,E)\;\;(A\,\_\,\_\,d\,\_)$

We may now see that these two subgoals are satisfied by the rows ($A\;B\;C\;D\;E$) and ($A\;b\;c\;d\;e$) from the hypothesis of the dependency to be proven.

If template dependencies are described using Horn clauses,[4] the variable for each embedded position is existentially quantified, which leads to Skolem functions. In positive hyper-resolution, the automated deduction version of the chase, these functions replace our use of arbitrarily chosen values. In a top-down proof, through the use of factoring as in the above example, resolvents with function terms may be deleted.

From Ref. 4 a template dependency may be written in the form:

$$\forall x_1 \ldots x_m \, \exists y_1 \ldots y_n \, \forall z_1 \ldots z_p \; R(\,) \wedge \ldots \wedge R(\,) \Rightarrow R(\,)$$

where the x_i correspond to symbols appearing in both the hypothesis and conclusion (the antecedent and conclusion of the implication), the y_i correspond to symbols appearing only in the conclusion, and the z_i correspond to symbols appearing only in the hypothesis. The y_i will be replaced by Skolem functions over all x_i. As is noted in Ref. 9, for (typed) template dependencies all y_i may be replaced by the same function. As an example, the following TD:

$$\begin{array}{l} A\,B\,C\,D\,E \\ \underline{A\,b\,c\,d\,e} \\ A\,B\,F\,d\,H \end{array}$$

may be written as:

$$R(x_1, x_2, z_1, z_2, z_3) \wedge R(x_1, z_4, z_5, x_3, z_6) \Rightarrow R(x_1, x_2, f(x_1, x_2, x_3), x_3, f(x_1, x_2, x_3))$$

using a Skolem function.

Theorem 1

Resolution with factoring and the following restrictions is complete for template dependency inference.

(*a*) Set-of-support with conclusion from dependency to be proven given support (i.e. only resolve with this clause or its descendants).

(*b*) All resolvents containing a function term are deleted.

Proof

Recall that the chase (or positive hyper-resolution) is complete. We will show that given a chase proof, our procedure generates a corresponding top-down proof. The generation of each new row in the chase method gives a total ordering on the generated symbols. In the top-down proof, our goal is free of functions. Each step in a top-down proof will:

(1) Remove subgoal corresponding to the most recent row from the chase proof which has not been removed.

(2) Add new subgoals from the hypothesis of the template dependency which forced the chase row to be created. Factor subgoals which correspond to the same row in the chase. This guarantees that a function term can never unify with a variable which appears in another subgoal. If during execution a function term appears, then we have replaced a row (subgoal) prior to accounting for its use in an hypothesis to create a later row, which contradicts the ordering induced by the chase proof. Q.E.D.

Corollary

Resolution without factoring, but with the following restrictions is complete for template dependency inference.

(1) Set-of-support with conclusion from dependency to be proven given support.

(2) All resolvents containing a nested function term are deleted.

Proof

Similar to main theorem, but all subgoals corresponding to the same chase row are removed consecutively using the same dependency. Q.E.D.

Enhancements

Additional conditions, besides those on function terms, can be used to delete more resolvents while still maintaining completeness. Factoring is known to increase the combinatorics and is often omitted when solving difficult problems with a theorem prover (and possibly sacrificing completeness). For this problem, any subgoal may factor with any other subgoal, making the situation worse. A simple literal numbering strategy will make factoring more reasonable.

Within each clause (i.e. set of subgoals), each subgoal is numbered uniquely. The initial goal from the dependency to be proven is assigned 1. When a subgoal numbered i is replaced by the k subgoals corresponding to an hypothesis, they are numbered ni, $ni+1, \ldots$ $ni+k-1$, where n is the maximum number of hypothesis rows of all template dependencies which are known to hold. This guarantees that subgoals are numbered uniquely.

When a resolvent is generated, the number of the replaced subgoal is recorded. In the future, only subgoals which are numbered higher may be replaced. We refer to these as *replaceable subgoals*, the others are *unreplaceable subgoals*. Unreplaceable subgoals are still eligible for factoring.

Factoring is also restricted, but so as to maintain completeness. When a resolvent is generated via some template dependency, each new subgoal may be factored with no more than one previously existing subgoal and any number of the new subgoals. The resulting subgoal is numbered with the highest number of the factored subgoals (and will be replaceable). Completeness can be shown by examining what factoring did in our proof of the top-down procedure. At any time there is no need to maintain more than one subgoal to correspond to a row generated in the chase. So by applying factoring as soon as subgoals are produced, we never allow two subgoals corresponding to the same chase row to exist. Finally, this fact also guarantees that for a proof thus produced there is a chase proof which performs the identical replacements, but in the opposite order.

To further restrict allowable resolvents, we define connected variables.

Definition. The variables in a replaceable subgoal are said to be *connected variables*. If x and y are connected variables and y appears in another subgoal, then all variables in that subgoal are also connected to x (i.e. connected is an equivalence relation).

If an unreplaceable subgoal does not contain a variable connected to a variable in a replaceable subgoal, it is due to one of two reasons: either (1) there is another resolvent differing only in the replaceable/unreplaceable designations which does replace this literal (i.e. factoring is not necessary for success); or (2) the subgoal is satisfied by one of the hypothesis rows for the dependency being proven. In the first case the resolvent should be deleted. In the second case the subgoal is removed (ordinarily all subgoals corresponding to the hypothesis of the dependency being proved (input-positive units) will be factored together and replaced as the last steps in the proof). Factoring is also restricted by only permitting factoring of subgoals which have a connection of variables. This lowers the number of factors which must be tested.

Subsumption can be used,[3] but must be modified due to the distinction between replaceable and unreplaceable subgoals. If a clause C subsumes a clause D, there must be a substitution which makes the set of subgoals of C a subset of the subgoals of D. To ensure completeness, a subgoal of C can correspond to a subgoal of D as long as the subgoal of C is replaceable or the subgoal of D is unreplaceable (in other words, an unreplaceable subgoal cannot correspond to a replaceable subgoal).

Subsumption is applied in particular way. Forward subsumption, or deleting a new clause which is subsumed by an older clause, is always allowed. Backward subsumption, or deleting an old clause which is subsumed by a new clause, is only allowed in the same level of breadth-first search. If applied to clauses in the preceding level, a proof may be blocked.

Application to a Result of Sagiv and Walecka

The decidability of the inference problem for embedded multivalued dependencies (EMVD) is an open problem in dependency theory. An EMVD is a template dependency with the following properties. (1) There are two hypothesis rows. (2) If a symbol appears in a column of both hypothesis rows, then it must appear in the conclusion in the same column.

Example EMVD

$$\begin{array}{cccc} A & B & C & D \\ A & b & c & d \\ \hline A & B & c & e \end{array}$$

which is usually written as $A \rightarrow\rightarrow B \mid \subset$. An EMVD $X \rightarrow\rightarrow Y \mid Z$, where X, Y and Z are sets of attributes, indicates that the X symbols appear identically in both hypothesis rows and the conclusion, the Y attribute symbols appear identically in the first hypothesis row and the conclusion, and the Z attribute symbols appear identically in the second hypothesis row and the conclusion.

Regardless of the 'simple' structure of these dependencies, most results have been negative. Sagiv and Walecka[6] examine a subclass of the EMVDs which do have a decidable inference problem. Their class, the Z-EMVDs, has the property that all EMVDs known to hold in an instance have the form $X \twoheadrightarrow Y|Z$, where Z is the same for all dependencies. We show that a top-down procedure which corresponds to their chase-style procedure is easily derived.

Suppose that $A \twoheadrightarrow CE|F$ is to be inferred in the relation scheme $R(ABCDEF)$. Z is just the attribute F. The goal is:

$(A \_ C \_ E f)$

The two positive units from the hypothesis which can satisfy goals are:

(A,B,C,D,E,F)
(A,b,c,d,e,f)

Now, suppose that $CD \twoheadrightarrow AE|F$ is known to hold (recall that F must appear this way in all EMVDs in the instance). This replacement gives the subgoals:

$(A \_ C\ 1\ E \_)\quad (\_ \_ C\ 1 \_ f)$

Replacing the first subgoal will always lead to a clause which will be subsumed by its parent clause. Consider replacement using $D \twoheadrightarrow ACE|F$:

$(A \_ C\ 1\ E \_)\quad (\_ \_ \_\ 1 \_ \_)\quad (\_ \_ C\ 1 \_ f)$

This clause is clearly subsumed. Even replacements allowing function terms will be subsumed, leaving factoring or satisfying the subgoal with an hypothesis fact. Factoring certainly does not help, since the original goal is obtained. Satisfying the subgoal with an hypothesis fact corresponds exactly to Sagiv and Walecka's procedure.

Complete Example of Top-Down Procedure for EMVDs

A more complete example of top-down inference is now examined. For the given instance, a reasonably efficient chase program did not find the proof after three hours on a VAX 11/780. The top-down procedure described herein, without the checks for connected variables, found a proof in less than a minute. Of course, the useful rows in the chase tableau may be determined from the top-down proof. The assumed EMVDs are:

(1) $I \twoheadrightarrow A|B$
(2) $E \twoheadrightarrow B|I$
(3) $G \twoheadrightarrow C|D$
(4) $DF \twoheadrightarrow C|B$
(5) $B \twoheadrightarrow D|G$
(6) $D \twoheadrightarrow B|F$
(7) $BD \twoheadrightarrow A|C$
(8) $C \twoheadrightarrow B|E$

The EMVD to be inferred is: $C \twoheadrightarrow A|B$.

Before giving the proof using our procedure, the chase proof (constructed using our top-down proof, not a chase program) is given (see Fig. 1). Each row is numbered by the subgoal for which it corresponds in the top-down proof. After the row, we give a triple $(X\ Y\ Z)$ where X is the EMVD used, Y is the number of the row matching the first hypothesis row, and Z is the number of the row matching the second hypothesis row. Instead of using function terms, the row number replaces the generated term.

Only the clauses actually needed for the proof are given. Sixty clauses generated by resolution or factoring were not deleted. The initial goal is:

$\underline{1}(A, b, C, \_, \_, \_, \_, \_, \_)$

The goals which can be satisfied by the hypothesis of $C \twoheadrightarrow A|B$ are:

(A,B,C,D,E,F,G,H,I)
(a,b,C,d,e,f,g,h,i)

Note that replaceable subgoals are underlined. Variables which do not appear elsewhere are represented with '_', otherwise a number is used. The first generated clause is from replacing the goal by using EMVD 7 $(BD \twoheadrightarrow A|C)$:

$\underline{2}(A, b, \_, 1, \_, \_, \_, \_, \_)\quad \underline{3}(\_, b, C, 1, \_, \_, \_, \_, \_)$

Next, subgoal 3 is replaced using EMVD 4 $(DF \twoheadrightarrow C|B)$:

$2(A, b, \_, 1, \_, \_, \_, \_, \_)\quad \underline{6}(\_, \_, C, 1, \_, 2, \_, \_, \_)$
$\underline{7}(\_, \mathrm{b}, \_, 1, \_, 2, \_, \_, \_)$

Subgoal 7 is now replaced using EMVD 6 $(D \twoheadrightarrow B|F)$:

$2(A, b, \_, 1, \_, \_, \_, \_, \_)\quad 6(\_, \_, C, 1, \_, 2, \_, \_, \_)$
$\underline{14}(\_, b, \_, 1, \_, \_, \_, \_, \_)\quad 15(\_, \_, \_, 1, \_, 2, \_, \_, \_)$

Factoring is now used to combine subgoals 15 and 6, and subgoals 14 and 2:

$\underline{14}\ (A, b, \_, 1, \_, \_, \_, \_, \_)\quad \underline{15}(\_, \_, C, 1, \_, \_, \_, \_, \_)$

501	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	Hypothesis
500	*a*	*b*	*C*	*d*	*e*	*f*	*g*	*h*	*i*	Hypothesis
250	250	*b*	*C*	250	*E*	250	250	250	250	(8 500 501)
125	125	*b*	125	125	*E*	125	125	125	*I*	(2 250 501)
62	*A*	*b*	62	62	62	62	62	62	*I*	(1 501 125)
31	31	*b*	31	62	31	31	250	31	31	(5 62 250)
15	15	15	*C*	62	15	15	250	15	15	(3 250 31)
7	7	*b*	7	62	7	15	7	7	7	(6 62 15)
3	3	*b*	*C*	62	3	15	3	3	3	(4 15 7)
1	*A*	*b*	*C*	62	1	1	1	1	1	(7 62 3)

Fig. 1. Chase proof.

Subgoal 15 is now replaced using EMVD 3 ($G \rightarrow\rightarrow C|D$):

$$14(A,b,\_,1,\_,\_,\_,\_,\_) \quad \underline{30}(\_,\_,C,\_,\_,\_,2,\_,\_) \quad \underline{31}(\_,\_,\_,1,\_,\_,2,\_,\_)$$

Subgoal 31 is now replaced using EMVD 5 ($B \rightarrow\rightarrow D|G$):

$$14(A,b,\_,3,\_,\_,\_,\_,\_) \quad 30(\_,\_,C,\_,\_,\_,2,\_,\_) \quad \underline{62}(\_,1,\_,3,\_,\_,\_,\_,\_) \quad \underline{63}(\_,1,\_,\_,\_,\_,2,\_,\_)$$

Factoring is again used, this time to combine subgoals 63 and 30, and subgoals 62 and 14:

$$\underline{62}(A,b,\_,\_,\_,\_,\_,\_,\_) \quad \underline{63}(\_,b,C,\_,\_,\_,\_,\_,\_)$$

Subgoal 62 is now replaced using EMVD 1 ($I \rightarrow\rightarrow A|B$):

$$\underline{63}(\_,b,C,\_,\_,\_,\_,\_,\_) \quad \underline{124}\ (A,\_,\_,\_,\_,\_,\_,\_,1) \quad \underline{125}(\_,b,\_,\_,\_,\_,\_,\_,1)$$

Subgoal 125 is now replaced using EMVD 2($E \rightarrow\rightarrow B|I$):

$$63(\_,b,C,\_,\_,\_,\_,\_,\_) \quad 124(A,\_,\_,\_,\_,\_,\_,\_,2) \quad \underline{250}(\_,b,\_,\_,1,\_,\_,\_,\_) \quad \underline{251}(\_,\_,\_,\_,1,\_,\_,\_,2)$$

Subgoals 250 and 63 are factored, and subgoals 251 and 124 are factored:

$$\underline{250}(\_,b,C,\_,1,\_,\_,\_,\_) \quad \underline{251}(A,\_,\_,\_,1,\_,\_,\_,\_)$$

Subgoal 250 is replaced using EMVD 8 ($C \rightarrow\rightarrow B|E$):

$$\underline{251}(A,\_,\_,\_,1,\_,\_,\_,\_) \quad \underline{500}(\_,b,C,\_,\_,\_,\_,\_,\_) \quad \underline{501}(\_,\_,C,\_,1,\_,\_,\_,\_)$$

Finally, subgoals 501 and 251 are factored:

$$\underline{500}(\_,b,C,\_,\_,\_,\_,\_,\_) \quad \underline{501}(A,\_,C,\_,\_,\_,\_,\_,\_)$$

This clause corresponds to the hypothesis of $C \rightarrow\rightarrow A|B$.

REFERENCES

1. A. V., Aho, C. Beeri and J. D. Ullman, The theory of joins in relational databases. *ACM Transactions on Database Systems* **4** (3), 297–314 (1979).
2. M. A. Casanova, R. Fagin and C. H. Papadimitriou, Inclusion dependencies and their interaction with functional dependencies. *ACM Symposium on the Principles of Database Systems 1982*, Los Angeles, California, March 1982, 171–176.
3. C.-L. Chang and R. C.-T. Lee. *Symbolic Logic and Mechanical Theorem Proving*. Academic Press, London (1973).
4. R. Fagin, Horn clauses and database dependencies. *Journal of the ACM* **29** (4), 952–985 (1982).
5. F. Sadri and J. D. Ullman, Template dependencies: a large class of dependencies in relational databases and its complete axiomatization. *Journal of the ACM* **29** (2), 363–372 (1982).
6. Y. Sagiv and S. F. Walecka, Subset dependencies and a completeness result for a subset of embedded multivalued dependencies. *Journal of the ACM* **29** (1), 103–117 (1982).
7. J. D. Ullman, *Principles of Database Systems*, 2nd ed. Computer Science Press (1982). Rockville, Maryland, USA.
8. M. Vardi, The implication and finite implication problems for typed template dependencies. *ACM Symposium on the Principles of Database Systems 1982*, Los Angeles, California, March 1982, 230–238.
9. B. P. Weems, On the Embedded Multivalued Dependency Membership Problem, *Ph.D. Dissertation*, Northwestern University, Evanston, Illinois, USA (1984).