Remarks on two new theorems of Date and Fagin

H.W.Buff, Swiss Reinsurance Company, 8022 Zürich, SWITZERLAND

Introduction

Date and Fagin give us two theorems (in [1]) that guarantee 4NF and 5NF respectively, if certain conditions are satisfied that can be expressed in terms of keys alone. This note generalizes one of them and then poses the question of the usefulness of such theorems for the practitioner. It also shows why theorems of this kind which would be really useful, cannot be expected.

Notation

The notation is as standard as possible. In the schema < U,Dep >, Dep is always a set of dependencies that apply to the set U of attributes (functional dependencies, FD's, multivalued dependencies, MVD's, and join dependencies, JD's). < U,Dep > |= c for a constraint c means that in all relations r for the schema (in which all dependencies of Dep are valid), c is also valid (Sorry, but in < U,Dep > |= c with c = "{AB} is a key", c is not a dependency, but a constraint). For a relation r, r| = c means that c is valid in r. X + Y is the union of X and Y.

An improved theorem

Definition. A subset C of U is a **cut** for the schema < U,Dep > if every key of < U,Dep > has a nonempty intersection with C and a nonempty intersection with U\C.

So a cut cuts every key into two nonempty pieces. Note that given all keys it is in most cases easy to check if there is a cut or not.

Theorem 1. If a relation schema is in BCNF and has no cut, then it is in 4NF.

Proof. Let < U,Dep > be a relation schema in BCNF which has no cut, and let X, Y and Z be pairwise disjoint subsets of U, Y and Z nonempty, such that their union equals U, and assume < U,Dep > |= X- > - > Y . Then X + Y is a superkey or X + Z is a superkey (otherwise every key would intersect Y and Z, and Y would be a cut). By coalescence ([2], page 132) we have < U,Dep > |= X-- > Y or < U,Dep > |= X-- > Z, so, by BCNF, X is a superkey. Therefore < U,Dep > is in 4NF.

Corollary 1 (Date/Fagin). If a relation schema is in BCNF and has a simple key, then it is in 4NF.

Proof. If a schema has a simple key, then it has no cut.

Example 1. Let the schema < {ABCD},Dep > be in BCNF and satisfy < U,Dep > |= exactly the sets {AB}, {AC}, {AD}, {BCD} are the keys. Then it is in 4NF (there is no cut).

Example 2. Let the schema < {A1,A2,...,An},Dep > be in BCNF and satisfy < U,Dep > |= exactly the sets {A1,A2}, {A2,A3}, ... , {An,A1} are the keys. If n is odd, then there is no cut and so the schema is in 4NF.

Note that both examples are not covered by corollary 1.

Is the theorem useful?

How useful are statements like theorem 1 or corollary 1 for the practitioner? If the designer is aware of all keys of a BCNF schema < U,Dep >, he can check in a minute whether the schema is 4NF by controlling whether for every nontrivial MVD X- > - > Y in Dep, X is a superkey (for the question of "hidden" MVD's see theorem 7.2 of [2], page 128). Similar remarks apply in the case of 5NF. The hard problem is to find all keys. Consider the following examples.

Example 3. Let U = {A1,A2,..,An} and Dep = {A1-- > U, A2-- > U} (n sufficiently large). Then < U,Dep > is 4NF. Then add A4- > - > A3. The resulting schema is not in 4NF. Then add

A3->->A2. The resulting schema is in 4NF again. Then add A6->->A5. The resulting schema is not in 4NF. Then add A5->->A4. The resulting schema is in 4NF again. And so on.

Example 4. The schema <U,Dep> with U={ABC} and Dep={A-->BC} is BCNF and there is a simple key. But now we add the MVD B->->C. Can we apply corollary 1? The answer is no, because the schema <U,Dep'> with Dep'=Dep+{B->->C} is no more BCNF (the MVD adds the FD B-->C, and B is not a superkey of <U,Dep'>).

Example 5. (This is a similar example for theorem 4.1 of [1], which says that a 3NF schema, in which every key is simple, is in 5NF.) The schema <U,Dep> with U={ABCDE} and Dep={A-->BCDE} is 3NF and all keys are simple. But if we add the join dependency JD{ABC,CD,DE}, the resulting schema <U,Dep'> with Dep'=Dep+{the JD} is no more in 3NF (the JD introduces the FD D-->E, among others, without D being a superkey or E being a prime attribute of <U,Dep'>).

These two examples show that neither the property "BCNF + there is a simple key" nor the property "3NF + all keys are simple" must necessarily remain true if the schema is extended. But the designer would like to have theorems with properties that are extendible.

What can be expected?

Definition. A schema <U,Dep'> is an extension of the schema <U,Dep> if Dep is a subset of Dep'. A schema <U,Dep> is in **hereditary 4NF** if <U,Dep> and all its extensions <U,Dep'> are in 4NF.

So we would be interested in properties of the set of keys of a schema, that guarantee hereditary 4NF. But the next theorem shows that nothing of interest can be expected.

Theorem 2. Let a relation schema be in BCNF. Then the schema is in hereditary 4NF if and only if all attributes are superkeys.

Proof. The if-part is almost trivial and left to the reader. Let <U,Dep> be a schema in BCNF and A in U an attribute so that {A} is not a superkey (which means that neither {A} nor the empty set is a key of <U,Dep>). We assume that U has at least two different attributes (else the theorem is trivially true). Let X be the empty set and Y:={A}. Then the MVD X->->Y is nontrivial. Let Dep':=Dep+{X->->Y} and Z:=U\{A}. We define a relation r for <U,Dep'> with two tuples t1 and t2 as follows: t1(B):=1 for all B in U, t2(A):=1 and t2(B):=2 for all B in Z. Every key of Dep intersects Z and <U,Dep> is BCNF, therefore r|=Dep. But r|=X->->Y and X is not a superkey of Dep' (because r|=Dep' and X is not a superkey of r). Therefore <U,Dep'> is not in 4NF and <U,Dep> not in hereditary 4NF.

Corollary 2. A BCNF schema is in hereditary 4NF if and only if it is in hereditary 5NF.

A final remark

Discussions like this once again show the importance of using a design language which makes it unnecessary for the designer to occupy himself with relational normalization. See [3] for this point.

References

[1] C.J.Date, R.Fagin, Simple Conditions for Guaranteeing Higher Normal Forms in Relational Databases; ACM Transactions on Database Systems, Vol. 17, No.3, September 1992, pages 465-476

[2] D.Maier, The theory of relational databases, Computer Science Press, 1983

[3] H.W.Buff, The Relational Model contra Entity Relationship?, ACM SIGMOD RECORD Vol. 21, No. 3, Sept. 1992, pages 33 - 34