Acta Informatica 28, 607–610 (1991)

Acta Informatica

Addendum to Null values in nested relational databases

Mark A. Roth[1], **Henry F. Korth**[2], **and Abraham Silberschatz**[2]

[1] Air Force Institute of Technology, AFIT/ENG, Wright Patterson AFB, OH 45433, USA
[2] Department of Computer Sciences, University of Texas at Austin, Austin, TX 78712, USA

Abstract. We define a new operator, decomposition projection, and show that extended projection is a precise generalization of decomposition projection with respect to unnesting, and that null-extended projection is a precise generalization of decomposition projection with respect to unnesting and PNF possibility function POSS*.

1. Introduction

Levene and Loizou [1] note that due to the impreciseness of extended union and null-extended union, extended projection and null-extended projection will, respectively, be imprecise as well, contrary to Propositions 5 and 11 of [3]. They suggest that a decomposition projection operator needs to be defined, which, similar to decomposition union, takes into account the join dependency which models the nested structure of the relation. In this addendum, we provide this definition and the correct statement and proof of Propositions 5 and 11.

2. Motivation for decomposition projection

Consider the very simple nested relation *emp* on scheme

$$\begin{aligned} \text{Emp} &= (\text{employee, Children}),\\ \text{Children} &= (\text{name, Games, Foods}),\\ \text{Games} &= (\text{game}),\\ \text{Foods} &= (\text{food}). \end{aligned}$$

which is shown in Fig. 1a. In this example, employee "Smith" has one child, "Bill", who likes to play "ball" and likes to eat "ice cream", and employee "Jones" who also has one child named "Bill", who likes to play "tag" and eat "pie". What should the result of $\pi^e_{Children}\ emp$ be? As in [3], we believe that all relations should be in Partitioned Normal Form (PNF) and thus, the result should merge the two Children nested tuples since they now agree

Employee	Children		
	Name	Games	Foods
		game	food
Smith	Bill	ball	ice cream
Jones	Bill	tag	pie

(a)

Employee	Name	Game
Smith	Bill	ball
Jones	Bill	tag

Employee	Name	Food
Smith	Bill	ice cream
Jones	Bill	pie

(b)

Fig. 1. a Nested relation on Emp scheme, and **b** equivalent 4NF relations on Emp_1 and Emp_2 schemes

on the *name* attribute. Since we have projected away the information needed to separate the two "Bill"s in the relation, the projection can only mean that children named "Bill" like to play "ball" and "tag" and like to eat "ice cream" and "pie". This may, at first, seem strange, but consider the traditional Fourth Normal Form (4NF) version of this same database. We would have two schemes:

$$\text{Emp}_1 = (\text{employee, name, game})$$
$$\text{Emp}_2 = (\text{employee, name, food}).$$

The data of Fig. 1a with these schemes is shown in Fig. 1b. If now we project out the *employee* attribute in both of the 4NF tables and then join the results we in fact get four tuples since for the only remaining common attribute, *name*, the value of "Bill" is the same in both relation's tuples.

This interaction can be formally expressed in terms of the multi-valued dependencies or, equivalently, the join dependency that should hold in the completely unnested 1NF relation that corresponds to a nested relation. The extended operators defined in [3] automatically maintain PNF and preserve the underlying join dependency when performing algebraic operations. If, however, we completely unnest a relation and then apply corresponding 1NF algebra operators, we do not get the same semantics. In [3] we defined *decomposition union* and *decomposition difference* operators to take into account the join dependency in standard union and difference operators. We now provide a similar definition for *decomposition projection.*

3. Formal development

The definition and proofs in this section are numbered to correspond with [3].

Definition 22a The *decomposition projection* or *Δ-projection* of a 1NF relation r on scheme R onto attributes Y is

$$\pi_Y^\Delta(r) = \bigcup_{t \in r[Y]}^{\Delta}(t)$$

where $\cup^\Delta$ is the Δ-union operator[1] defined in Definition 18 of [3].

Proposition 5 (revised) *Extended projection is a precise generalization of Δ-projection with respect to unnesting.*

Proof. We show $\mu^*(\pi_X^e(r)) = \pi_{X'}^\Delta(\mu^*(r))$, where X' are all of the attributes of the completely unnested scheme X. Let $\{t_1, t_2, \ldots, t_n\}$ be the tuples of r. By the definitions of extended projection and Δ-projection we have

$$(1) \qquad \mu^*(\pi_X^e(r)) = \mu^*\left(\bigcup_{t \in r[X]}^{e}(t)\right) = \mu^*(t_1[X] \cup^e t_2[X] \cup^e \ldots \cup^e t_n[X])$$

and

$$(2) \qquad \pi_{X'}^\Delta(\mu^*(r)) = \bigcup_{t \in \mu^*(r)[X']}^{\Delta}(t) = \mu^*(t_1)[X'] \cup^\Delta \mu^*(t_2)[X'] \cup^\Delta \ldots \cup^\Delta \mu^*(t_n)[X']) = \mu^*(t_1[X]) \cup^\Delta \mu^*(t_2[X]) \cup^\Delta \ldots \cup^\Delta \mu^*(t_n[X]).$$

By a straightforward extension of the proof of Proposition 2 of [3] which states that extended union is precise generalization of Δ-union, or $\mu^*(p \cup^e q) = \mu^*(p) \cup^\Delta \mu^*(q)$, one can show the equivalence of Eq. (1) and (2). □

Proposition 11 (revised) *Null-extended projection is a precise generalization of Δ-projection with respect to unnesting and PNF possibility function POSS*.*

Proof. We show $\mu^*(POSS^*(\pi_X'^e(r))) = \pi_{X'}^\Delta(\mu^*(POSS^*(r)))$, where X' are all of the attributes of the completely unnested scheme X. By Proposition 5 (revised), we know that extended projection is a precise generalization of Δ-projection, and so $\pi_{X'}^\Delta(\mu^*(POSS^*(r))) = \mu^*(\pi_X^e(POSS^*(r)))$. Thus, we only need to show that $POSS^*(\pi_X'^e(r)) = \pi_X^e(POSS^*(r))$. By the definitions of extended projection and null-extended projection we have

$$(3) \qquad POSS^*(\pi_X'^e(r)) = POSS^*\left(\bigcup_{t \in r[X]}^{e'}(t)\right) = POSS^*(t_1[X] \cup^{e'} t_2[X] \cup^{e'} \ldots \cup^{e'} t_n[X])$$

---

[1] The join dependency utilized in the application of the Δ-union will contain those schemes that are the non-empty intersections of the path set of the scheme tree of R [2] with attributes Y. Let $*(Z_1, Z_2, \ldots, Z_m)$ be this join dependency. An equivalent definition of Δ-projection is

$$\pi_Y^\Delta(r) = \bowtie(r[Z_1], r[Z_2], \ldots, r[Z_m])$$

and

$$(4)\quad \pi^e_X(POSS^*(r)) = \bigcup^e_{t\in POSS^*(r)[X]} (t)$$
$$= POSS^*(t_1)[X] \cup^e POSS^*(t_2)[X] \cup^e \ldots \cup^e POSS^*(t_n)[X])$$
$$= POSS^*(t_1[X]) \cup^e POSS^*(t_2[X]) \cup^e \ldots \cup^e POSS^*(t_n[X])$$

In Proposition 8 of [3], we proved that $POSS^*(r \cup^{e'} q) = POSS^*(r) \cup^e POSS^*(q)$ and a straightforward extension of that proof shows the equivalence of Eq. (3) and (4).

References

1. Levene, M., Loizou, G.: Correction to "Null values in nested relational databases" by M.A. Roth, H.F. Korth, and A. Silberschatz. Acta Inf. **28**, 603–605 (1991)
2. Meral Özsoyoğlu, Z., Yuan, L.Y.: A new normal form for nested relations. ACM Trans. Database Syst. **12**, 111–136 (1987)
3. Roth, M.A., Korth, H.F., Silberschatz, A.: Null values in nested relational databases. Acta Inf. **26**, 615–642 (1989)