It is obvious that these changes do not affect the synchronization and the response time is $T_1(n)$. Some minor variations of this system may be obtained by introducing a new processor in order to avoid an overloading of DP.

A new two-dimensional systolic array can be obtained by applying the results presented in the previous section. The structure and the performances are preserved.

REFERENCES

1. Baudet, G. M. — *Design and complexity of VLSI algorithms*, Fundations of Computer Science, Part 1, Algorithms and Complexity, Mathematical Centre Amsterdam, 1983, pp. 49—74.
2. Foster, M. J., Kung, H. T. — *Design of special purpose VLSI chips: examples and opinions*, Techn. Rep. *CMU-CS-79-147*, Dept. of Computer Science, Carnegie Mellon University, Pittsburgh, 1979.
3. Kramer, M. R., Leeuwen, J. van — *Systolic computation and VLSI*, Fundations of Computer Science, Part, 1, Algorithms and Complexity, Mathematical Centre Amsterdam, 1983, pp. 75—103.
4. Leiserson, C. E. — *Area-Efficient VLSI Computation*, MIT Press, Massachusetts 1983.
5. Ullman, J. D. — *Computational aspects of VLSI*, Computer Science Press, Inc. 1984.

Received 28.III. 1985

Computer Center
Polytechnical Institute
6600 Iasi, Romania

Analele ştiinţifice ale Universităţii „Al. I. Cuza" din Iaşi
Tomul XXXI, s. I a, Matematică, 1985, f. 3.

ON A NEW KIND OF DEPENDENCIES IN RELATIONAL DATABASES

BY

C. MASALAGIU and H. LUCHIAN

§ 1. **Introduction.** We suppose the reader familiar with the usual approach for the theory of dependencies in relational databases ([[4]], [[1]], [[7]], [[10]]) .

Let $\mathcal{U}$ be a denumerable set of *attributes* and $M=\{., c\}$ the set of marks. Let $\mathcal{U}_M=\mathcal{U}\times M$.

In the sequel, we shall use the following notations : $[m]$ will denote the set $\{1, 2,\dots,m\}$, $m\in N$; for $a=(u, m)\in\mathcal{U}_M$, we shall consider supp $(a)=u$ and for $\{a_1,\dots,a_n\}\subseteq\mathcal{U}_M$, supp $(\{a_1,\dots,a_n\})=\{\text{supp }(a_1),\dots,\text{supp }(a_n)\}$; for $\{A_1,\dots,A_m\}\subseteq\mathcal{U}$ we shall use the shorthand $A_1, . . A_m$.

A *relation scheme* over $\mathcal{U}$ is any finite subset of $\mathcal{U}$. From now on, we shall reffer to a fixed relation scheme $R=\{A_1,\dots, A_n\}$, $n\geqslant 2$; $R\times M$ will be denoted by R_M.

For each $A\in\mathcal{U}$, let the set D_A be the *domain* of A ; card $D_A\leqslant\aleph_0$. For each $a\in\mathcal{U}$, let $\alpha_A\subseteq D_A\times D_A$ be a reflexive, nontrivial relation ($\alpha_A\neq\neq D_A\times D_A$).

Let $\text{Tup}_R\mathcal{U}=\{u/u : [n]\to\bigcup_{i=1}^{n} D_{A_i}, u(\text{i})\in D_{A_i}\}$ be the set of tuples over R (we shall alternatively use $u(i)$ or $u(A_i)$, D_i or D_{A_i}, α_i or α_{A_i}, a.s.o.).

Let Q be any set $[n]$ or N itself. A *Q-instance* for R is an injective function $I_R : Q\to\text{Tup}_R\mathcal{U}$. The set of all Q-instances for R will be denoted by $\mathcal{I}_{R, Q}$.

An *instance* for R is a Q-instance, for some Q. If I is a Q-instance for R and $Q_1\subseteq Q$, any Q_1-instance I_1: $Q_1\to\text{Tup}_R\mathcal{U}$ such that $I/Q_1=I_1$, will be called *subinstance* of I (Q_1 will be either any $[n]$, or N itself). The set of all instances for R is denoted by $\mathcal{I}_R$.

Remark. The instances in the definition above appear to be ordered sets of tuples ; we note that this „ordering" depends on what ordering (if any!) is considered on N. A forth coming paper will examine closely the necessity of changing some classical definitions. For the moment let us only note that there are cases where ordered instances must be explicitely taken into account.

As usual, only a subset of $\mathcal{I}_R$ will be considered, which we shall call *the set of admissible instances* ($\mathcal{A}_R$). The admissible instances will obey certain specific requirements for $Im(I_R)$, $I_R\in\mathcal{I}_R$. For example: $\mathcal{A}_R^1=\bigcup_{m\geqslant 2}\mathcal{I}_{R, [m]}$, $A_R^2=$ $=\bigcup_{m\leqslant n}\mathcal{I}_{R, [m]}$, $\mathcal{A}_R^3=\mathcal{I}_{R, N}$, $\mathcal{A}_R=\mathcal{A}_R^1\cup\mathcal{A}_R^3$, a.s.o.

Definition. *A relational dependency over R is an ordered pair, $\sigma = (X, Y)$, of subsets of R_M.*

We shall denote σ by $X \to Y$. A Q-instance $I \in \mathcal{A}_R$ *satisfies* σ (written $I \models \sigma$) if the following formula is true $\bigwedge_{i=1}^{r} (I(k)(X_i) \Box I(k+1)(X_i)) \Rightarrow \bigwedge_{i=1}^{s} (I(k)(Y_i) \Box I(k+1)(Y_i))$, for any $k \in [p]$ when $Q = [p+1]$ and for any $k \in N$ when $Q = N$, where $X = X_1 \ldots X_r$ and $Y = Y_1, \ldots Y_s$, $r, s \geq 1$. $I(k)(z) \Box I(k+1)(z)$ stands for $(I(k)(j), I(k+1)(j)) \in \alpha_{A_j}$, if $z = (A_j, .)$ and for $(I(k)(j), I(k+1)(j)) \notin \alpha_{A_j}$, if $z = (A_j, c)$ (for all $z \in X \cup Y$). '$\bigwedge$' and '$\Rightarrow$' have the usual meaning from mathematical logic.

Remark. The usual definition of the instance (unordered *set* of tuples) — which can be easily formulated in our framework — would permit the definition of functional dependencies as a particular type of relational dependencies. Moreover, order dependencies of [[1]] can be expressed in this approach, by suitable definitions of M and α_{A_i}.

Let Γ be a set of relational dependencies, supp $\Gamma \subseteq R$ and σ a relational dependency, supp $\sigma \subseteq R$.

Definition. *The instance $I \in \mathcal{A}_R$ satisfies Γ (Γ is satisfied by I) if $I \models \gamma$, for any $\gamma \in \Gamma$. The usual notation $I \models \Gamma$ will be used.*

Definition. *σ is a consequence of Γ ($\Gamma \models \sigma$) if, for any $I \in \mathcal{A}_R$ which satisfies Γ, I also satisfies σ.*

The set of al dependencies that are consequences of Γ will be denoted by $\overline{\Gamma}$.

§ 2. A sound and complete set of inference rules for relational dependencies. Given Σ a set of relational dependencies for a relation scheme R, a well-known problem in the theory of relational databases is to find the set $\overline{\Sigma}$. This is usually done by considering Σ to be a set of 'axioms' and by providing a set of 'inference rules' INF such that $\overline{\Sigma} = \Sigma^{+}_{\mathrm{INF}}$, $\Sigma^{+}_{\mathrm{INF}}$ being the set of all the dependencies that can be obtained from Σ using INF. Usually, $\sigma \in \Sigma^{+}_{\mathrm{INF}}$ is denoted by $\Sigma \vdash_{\mathrm{INF}} \sigma$. For details, the reader is refered to [[4]], [[3]], [[2]], [[7]], [[8]], [[9]], [[10]] and for standard notions of mathematical logic, to [[5]], [[6]].

Let $R = A_1 A_2 \ldots A_n$ be our fixed relation scheme. For any dependency σ, supp $\sigma \subseteq R$, we shall construct a propositional calculus formula $\underline{\sigma}$ over the set of propositional variables $A_1, \ldots, A_n$ in the following way: if $\sigma = X_1 \ldots X_r \to Y_1 \ldots Y_s$ then $\underline{\sigma} = \underline{X_1} \ldots \underline{X_r} \to \underline{Y_1} \ldots \underline{Y_s}$, where, for any $Z \in X_1 \ldots X_r Y_1 \ldots Y_s$, $\underline{Z} = A_i$ if $Z = (A_i, .)$ and $\underline{Z} = \neg A_i$ if $Z = (A_i, c)$. Here, '$\neg$' '$\to$' and concatenation stand for logical 'non', 'implies' and 'and', respectively. If Σ is a set of dependencies over R, let $\underline{\Sigma} = \{\underline{\sigma} / \sigma \in \Sigma\}$.

Let σ be a dependency for R and $I \in \mathcal{J}_{R, [2]}$.

Lemma 1. *I satisfies σ iff $\underline{\sigma}$ is true for the „special" truth assignment ψ_I defined as follows: for any variable $A_i \in R$, $\psi_I(A_i) = 1$ iff $(I(1)(i), I(2)(i)) \in \alpha_i$.*

Proof. Suppose I satisfies $\sigma = X_1 \ldots X_r \to Y_1 \ldots Y_s$. Suppose also $\underline{X_1} \ldots \underline{X_r}$ is true under the special truth assignment ψ_I. Hence $\psi_I(\underline{X_j}) = 1$, $\forall j = \overline{1, r}$.

That is, $(I(1)(i), (I(2)(i)) \in \alpha_i$ if $\underline{X_j} = A_i$ and $(I(1)(i), I(2)(i)) \notin \alpha_i$ if $\underline{X_j} = \neg A_i$. By the definition of satisfaction, it follows that $(I(1)(i), I(2)(i)) \in \alpha$ if $Y_k = (A_i, .)$ and $(I(1)(i), I(2)(i)) \notin \alpha_i$ if $Y_k = (A_i, c)$; $\forall k = \overline{1, s}$. It follows that $\underline{Y_1} \ldots \underline{Y_s}$ is true, that is $\underline{\sigma}$ is true. A similar argument is used for the converse. □

Let Σ be a set of dependencies for R and σ a dependency for R.

Lemma 2. Let $I \in \mathcal{A}_R$, with $I \models \Sigma$ and $I \not\models \sigma$. Then there is a two-tuple subinstance I_1 of I which satisfies Σ and does not satisfy σ.

Proof. It follows directly from the appropriate definition. □

If R, Σ, σ are as above, we shall write $\underline{\Sigma} \models \underline{\sigma}$ if $\underline{\sigma}$ is a logical consequence from $\underline{\Sigma}$.

Theorem 1. *The following properties are equivalent:*

(i) $\Sigma \models \sigma$.

(ii) $I \models \Sigma$ *implies* $I \models \sigma$ *for any* $I \in \mathcal{J}_{R, [2]}$

(iii) $\underline{\Sigma} \models \underline{\sigma}$.

Proof. (i)⇔(ii). The „⇒" part is trivial and the converse follows from Lemma 2.

(ii)⇒(iii). Assume that (ii) is true and (iii) false. Then there exists an assignment ψ for which the formulas in $\underline{\Sigma}$ are true and $\underline{\sigma}$ is false. We construct a two-tuple instance $I : \{1, 2\} \to \mathrm{Tup}_R \mathcal{U}$ as follows: $I(1)(A) = \underline{1}$, $\forall A \in R$ and $I(2)(A) = \begin{cases} \underline{0} \text{ if } A \text{ is assigned } 0 \text{ in } \psi \\ \underline{1} \text{ if } A \text{ is assigned } 1 \text{ in } \psi \end{cases}$ (for each A, $\underline{0}$ and $\underline{1}$ are fixed elements in D_A such that $(\underline{1}, \underline{0}) \notin \alpha_A$). Let us note that ψ is the special truth assignment, for $I((\underline{1}, \underline{1}) \in \alpha_A$ by reflexivity). From Lemma 1 it follows that $I \models \Sigma$ and $I \not\models \sigma$, a contradiction.

(iii)⇒(ii). Assume $\underline{\Sigma} \models \underline{\sigma}$. Let $I \in \mathcal{J}_{R, [2]}$ be any instance such that $I \models \Sigma$ and let ψ_I be the special truth assignment for I. From Lemma 1 and $\underline{\Sigma} \models \underline{\sigma}$ it follows that $\underline{\sigma}$ is true in ψ_I and $I \models \sigma$. □

In the sequel, X, Y, Z, will be nonempty sets of marked attributes T, U, W — sets of marked attributes, $\overline{A}, \overline{B}, \overline{C_i}, \overline{B_j}$ $(i \in [n+1],\ j \in [k+1])$ — marked attributes; $\overline{A}^c$ will denote (A, c) if $\overline{A} = (A, .)$ and $(A, .)$ if $\overline{A} = (A, c)$ (for any $A \in R$).

The following set of inference rules will be refered to as INF:

(1) $$\overline{X \to Y} \quad \forall X, Y, Y \subseteq X$$

(2) $$\frac{X \to Y}{XV \to YW} \quad \forall X, Y, V, W \quad W \subseteq V$$

(3) $$\frac{X \to Y,\ Y \to Z}{X \to Z} \quad \forall X, Y, Z$$

(4) $$\frac{T\bar{A}\to\bar{B}}{T\bar{B}^c\to\bar{A}^c}\quad \forall T,\ \bar{A},\ \bar{B},$$

(5) $$\frac{\bar{C}_1\ldots\bar{C}_i\ldots\bar{C}_n\to\bar{C}_{n+1}\ ;\ \bar{B}_1\ldots\bar{B}_j\ldots\bar{B}_k\to\bar{B}_{k+1}}{\bar{C}_1\ldots\bar{C}_{i-1}\bar{C}_{i+1}\ldots\bar{C}_n(\bar{C}_{n+1})^c\bar{B}_1\ldots\bar{B}_{j-1}\bar{B}_{j+1}\ldots\bar{B}_k\to\bar{B}_{k+1}}$$

$\forall\bar{C}_1\ldots\bar{C}_{n+1},\ \bar{B}_1,\ldots,\bar{B}_{k+1},\ n\geqslant 1,\ k\geqslant 1$ where $\bar{C}_i$ coincides with $(\bar{B}_j)^c$ and $\bar{C}_{n+1}\neq\bar{B}_{k+1}$ if

$$\bar{C}_1\ldots\bar{C}_{i-1}\bar{C}_{i+1}\ldots\bar{C}_n=\bar{B}_1\ldots\bar{B}_{j-1}\bar{B}_{j+1}\ldots\bar{B}_k$$

(6) $$\frac{X\bar{A}\to Y,\ X\bar{A}^c\to Y}{X\to Y}\quad \forall X,\ Y,\ A$$

Remark. We shall prove that INF is a sound and complete set of inference rules for relational dependencies. From Theorem 1 we see that the proof can be done in a fragment of propositional logic, any dependency σ being translated into the corresponding formula $\underline{\sigma}$ and any inference rule — into an inference rule for formulas (we shall denote the translated inference rules PINF). We refer to [[3]] for a more detailed presentation of these ideas. □

Theorem 2. PINF *is sound, that is if* $\underline{\Sigma}\vdash_{\overline{\text{PINF}}}\underline{\sigma}$, then $\underline{\Sigma}\models\underline{\sigma}$ (or $\underline{\Sigma}^+_{\text{PINF}}\subseteq\underline{\Sigma}$).

Proof. It is not difficult to see that for any rule in PINF, if the hypotheses are true, the conclusion is also true.

Remark. Given $\underline{\Sigma}$, the set of 'axioms', we can transform it in a set $\underline{\Sigma}'$ in the following way : replace any $\underline{\sigma}=X\to\bar{B}_1\ldots\bar{B}_p\in\underline{\Sigma}$ by $\{X\to\bar{B}_1,\ldots,X\to\bar{B}_p\}$ (using (1) and (3)) ; then, apply (5), (6), (4) as many times as possible. It follows that $\underline{\Sigma}'$ does not contain complementary literals (that is, there do not exist $\sigma'\neq\sigma''$ in $\underline{\Sigma}'$, $\sigma'=X_1\bar{B}\to\bar{C}_1$ and $\sigma''=X_2(\neg\bar{B})\to\bar{C}_2$).

Theorem 3. *A formula* $\underline{\sigma}=Y\to\bar{B}$ *is a logical consequence from* $\underline{\Sigma}$ *iff it is a logical consequence from* $\underline{\Sigma}'$.

Proof. (sketch) : If $\underline{\sigma}$ is a logical consequence from $\underline{\Sigma}'$, it is also a logica consequence from $\underline{\Sigma}$, since $\underline{\Sigma}'$ consists of formulas that are logical consequences from $\underline{\Sigma}$. For the converse implication, using the 'derived' inference rule ([[4]], [[3]]) $\dfrac{X\to Y,\ X\to Z}{X\to YZ}$, we must prove only that : if $p_1=\bar{B}_1V\ \ldots\ V\bar{B}_n$ and $p_2=\bar{C}_1V\ldots V\bar{C}_k$ are clauses $(k, n\geqslant 2)$ and $p=\bar{B}_2V\ldots V\bar{B}_n\ V\bar{C}_2V\ldots V\bar{C}_k$ is a resolvent of p_1 and p_2 (that is $\bar{B}_1=\neg\bar{C}_1$), a clause q (with at least two literals) is a logical consequence from $\{p\}$ if it is a logical consequence from $\{p_1, p_2\}$. It can be shown that for q to be a logical consequence from $\{p_1, p_2\}$ but not from $\{p\}$, q must have a specific form ; we can find then an assignment for which p_1 and p_2 are true and q is false, — a contradiction.

Theorem 4. PINF *is complete (that is, if* $\underline{\Sigma}\models\underline{\sigma}$, *then* $\underline{\Sigma}\vdash_{\overline{\text{PINF}}}\underline{\sigma}$ *or equivalently,* $\underline{\Sigma}\subseteq\underline{\Sigma}^+_{\text{PINF}}$).

Proof. Using Theorem 3, $\underline{\Sigma}$ can be replaced by $\underline{\Sigma}'$. We shall show that if σ cannot be proved from $\underline{\Sigma}'$ using PINF, it cannot be a logical consequence from $\underline{\Sigma}'$. Firstly, let us note that if $\underline{\sigma}=X\to Y$ cannot be proved from $\underline{\Sigma}'$, then there must exist a variable $\bar{A}\in Y$ such that $X\to\bar{A}$ cannot be proved from $\underline{\Sigma}'$. We show that there is an assignment ψ for the variables $B\in R$ such that the elements of $\underline{\Sigma}'$ are true and $\underline{\sigma}$ is false. Namely, if $\bar{\sigma}=\bar{A}_1\ldots\bar{A}_t\to\bar{A}$, we put $\psi(\bar{A})=0$ (that is, $\psi(A)=0$ if $A=\bar{A}$ and $\psi(A)=1$ if $\bar{A}=\neg A$) and $\psi(\bar{A}_i)=1$, $i=\overline{1,t}$ (that is $\psi(A_i)=1$ if $\bar{A}_i=A_i$ and $\psi(A_i)=0$ if $\bar{A}_i=\neg A_i$). Let us note that it is not possible to have in $\underline{\sigma}$ $\bar{A}_i\equiv\bar{A}$ or $\bar{A}_{ii}\equiv A_j$ because otherwise $X\to\bar{A}$ could be proved from $\underline{\Sigma}'$ (that is, until now ψ is well-defined). To have the complete definition of ψ, for any other variable $\bar{B}$ (different from $\bar{A}_i$, $\bar{A}$), we put $\psi(\bar{B})=0$ if $\bar{B}$ appears in a left member of a formula in $\underline{\Sigma}'$. For the remaining variables $\bar{B}$ we put $\psi(\bar{B})=1$ (ψ is still well defined because $\underline{\Sigma}'$ does not contain complementary literals). All the formulas in $\underline{\Sigma}'$ are true in ψ(otherwise $\underline{\sigma}$ could be proved from $\underline{\Sigma}'$).

REFERENCES

1. Ginsburg, S., Hull, R. — *Order Dependency in the Relational Model*, Research Report, Univ. of Southern California, 1982.
2. Thalheim, B. — *A Complete Axiomatization for full Join Dependencies in Relations*, Bulletin of the EATCS, No. 24, October 1984, pp. 109–115.
3. Sagiv Y., Delobel, C., Scott D., Parker, jr., Fagin R. — *An Equivalence Between Relational Database Dependencies and a Fragment of Propositional Logic*, JACM, vol. 28, no. 3, July 1981, pp. 435–453
4. Ullman, J. D. — *Principles of Database Systems*, Computer Science Press, 1980
5. Yasuhara, A. — *Recursive Function Theory and Logic*, Academic Press, 1971
6. Chang, C. L., Lee, R. C., — *Symbolic Logic and Mechanical Theorem Proving*, Academic Press, 1973
7. Minker, J. — *On Theories of Definite and Indefinite Databases*, TR. 1250, Univ. of Maryland
8. Reiter, R. — *Towards a Logical Reconstruction of Relational Database Theory* — In „On Conceptual Modelling" — Springer Verlag, 1984, pp. 191–238
9. Kowalski, R. — *Logic as a Database Language*. In „The Proceedings of the Third British National Conference on Databases", Cambridge University Press, 1984, pp. 103–132
10. Fagin, R., Vardi, Y. M. — *The Theory of Data Dependencies*. A Survey — Research Report, IBM Research Laboratory, San Jose, California, 1984

Received 28.V. 1985

Computer Center
University of Iaşi
6600 Iaşi, Romania