Combinatorial Construction of Time-Stamp Systems and Interpolation Systems *

VALÉRIE ALBIERO ERIC SOPENA

Université Bordeaux I
Laboratoire Bordelais de Recherche en Informatique
Unité associée C.N.R.S. 1304
351, cours de la Libération
F-33405 TALENCE - FRANCE

{albiero,sopena}@labri.u-bordeaux.fr

Abstract. A time-stamp system allows to maintain in a distributed way a total order among a set of objects by assigning them some labels or time-stamps. An elementary operation consists in moving one of these objects from the ith position to the first one by only modifying its time-stamp. We introduce the notion of interpolation system, which generalizes the previous notion by allowing the objects to move from any position to any other position. Time-stamp systems and interpolation systems can be described as directed graphs (whose vertices stand for time-stamps) satisfying some special properties. We study in this paper different construction mechanisms leading to such systems.

Résumé. Un système d'estampillage permet de gérer de façon distribuée un ordre total sur des objets en leur attribuant des étiquettes ou estampilles. Une action élémentaire consiste à faire passer un objet d'une position i quelconque en position 1 en modifiant uniquement l'estampille de cet objet. Nous introduisons ici la notion de système d'interpolation qui généralise la notion précédente en permettant de faire passer un objet d'une position i quelconque en une position j quelconque. Les systèmes d'estampillage et d'interpolation peuvent être représentés sous forme de graphes orientés (dont les sommets constituent les estampilles) satisfaisant certaines propriétés. Nous étudions ici différents mécanismes de construction permettant d'obtenir de tels systèmes.

1 Introduction

One of the frequently addressed problems in distributed computing consists in maintaining some structure on a set of objects (data, events) by means of some additional information, called *time-stamp*, associated with each of these objects. For instance, a distributed implementation of a single piece of data, or variable, can be achieved as follows : each pro-

*With the support of the European Basic Research Action ESPRIT No 3166 (ASMICS), the European Community Cooperative Action IC-1000 (ALTEC) and the inter-PRC Project AQSI.

cess holds a multi-reader-single-writer register [6] associated with this variable. If a process wants to determine the actual contents of the variable, it scans all the registers and, thanks to the time-stamps, compute the last modified value. In order to modify the contents of the variable, a process must be able to compute a new time-stamp such that all processes will consider that time-stamp as the most recent one. The register then writes the new value of the variable, as well as the computed time-stamp, in its own register. Depending on whether overlapping actions of processes are allowed or not, one speaks about *concurrent* or *sequential* systems.

We consider in this paper sequential time-stamp systems and generalize them by introducing the new concept of interpolation system. The general time-stamping problem can be illustrated as follows :

THE MAILBOX PROBLEM : *A set of users share a common mailbox. At any time, the mailbox may contain at most one message per user : when a user wants to put a message in the mailbox he first removes his old own message, if necessary. Every time a message is put in the mailbox, it is associated with a time-stamp. The aim of this time-stamping mechanism is the following : by looking at any two time-stamps currently in the mailbox one must be able to determine the real-time order of their respective deposit dates.*

Such a system allows to maintain a total order (here the temporal order) among a set of objects (the messages). The time-stamps allow to compare, with respect to that order, any two messages in the mailbox. We are thus able to retrieve the total order among all the current messages. A system such that the time-stamps simply allow us to retrieve the last created message is called a *weak* time-stamp system [2, 11] (such a system gives us a solution for the distributed implementation of a variable discussed before). In [9], Saks and Zaharoglou considered a time-stamp system, which could be called *global*, allowing to retrieve the total order among the messages by looking at *all* the time-stamps at the same time. This system does not allow to compare any given pair of messages.

Israëli et Li [5] have shown that the time-stamping problem can be solved by using a *finite* set of time-stamps when the set of users is bounded. They proposed a combinatorial solution to that problem by means of a directed antisymmetric graph (whose vertices are the time-stamps) satisfying some specific property.

Another distinction can be made depending on whether we require the time-stamp to contain the identity of the user or not. In this case, we speak about *signed* or *unsigned* systems. In the signed case, each user has his own set of time-stamps. In the unsigned case, each time-stamp may be indifferently used by any of the users. The time-stamping problem in the signed case has been optimally solved by Zielonka [13].

We introduce here the notion of *interpolation system*, which allows a user to insert his new message in *any position* within the set of current messages. The total order among the messages thus obtained does no longer necessarily reflect the temporal order of their deposit dates. Such systems may also be described by means of antisymmetric directed graphs. We are then interested in several construction mechanisms leading to such graphs for both time-stamp and interpolation systems. We prove that Zielonka's solution also provides an optimal solution for interpolation systems in the signed case. For this reason, we will essentially consider in the following the unsigned case.

This paper is organized as follows : we introduce in section 2 the main definitions and properties we will use in the sequel. The following sections are devoted to several construction mechanisms leading to time-stamp and interpolation systems. All complete proofs can be found in [1].

2 Definitions and basic properties

A directed graph G is given as a finite set of *vertices* $V(G)$ and a set of *arcs* $A(G) \subset V(G) \times V(G)$. We will only consider loopless and antisymmetric directed graphs (that is such that $(x,y) \in A(G) \Longrightarrow (y,x) \notin A(G)$), simply called *graphs* later on. If (x,y) is an arc, we say that x is a *predecessor* of y and that y is a *successor* of x. We denote by $\Gamma_G^+(x)$ (resp. $\Gamma_G^-(x)$) the set of successors (resp. predecessors) of a vertex x in G. The subgraph of G induced by the successors (resp. predecessors) of x is denoted by G_x^+ (resp. G_x^-).

A sequence of vertices $(y_1, y_2, \ldots, y_p)$ is called an *ordered sequence* if for any i, j with $1 \leq i < j \leq p$, y_j is a successor of y_i. We will say that a vertex x is a successor (resp. predecessor) of an ordered sequence $(y_1, y_2, \ldots, y_p)$ if for any i, $1 \leq i \leq p$, x is a successor (resp. predecessor) of y_i. By convention, we will consider that any vertex is a successor and a predecessor of the empty sequence.

Definition 2.1 Let k be a strictly positive integer ; a graph G is a *time-stamp system of order k* if the following condition holds :

(E_k) any ordered sequence in G having at most $k-1$ elements has a successor.

Such a graph gives a solution to the above-stated mailbox problem (for k users) as follows : we use the vertices of G as time-stamps and the precedence relation is given by the set of arcs. When a user wants to put a new message in the mailbox he chooses as time-stamp a vertex which is a successor of the vertices currently in the mailbox. It is not difficult to check that by using this algorithm, the set of time-stamps which are currently in the mailbox is always an ordered sequence in G. Property (E_k) ensures that a successor for such a sequence will always exist.

Definition 2.2 Let k be a strictly positive integer ; a graph G is an *interpolation system of order k* if the following condition holds :

(I_k) any ordered sequence $(s_1, s_2, \ldots, s_p)$ in G having at most $k-1$ elements is such that :

$$\forall\ 1 \leq i \leq p+1,\ \exists\ y_i \in V(G)\ /\ \begin{cases} 1 \leq j < i \Longrightarrow (s_j, y_i) \in A(G) \\ i \leq j \leq p \Longrightarrow (y_i, s_j) \in A(G). \end{cases}$$

We will say that such a sequence can be *interpolated* in G. The value i is the *interpolation position* and y_i is said to be *inserted* in position i.

Note that property (I_k) generalizes property (E_k). Hence, any interpolation system is also a time-stamp system of the same order.

Example 2.3 A graph satisfies the property (E_2) (resp. (I_2)) if all its vertices have a successor (resp. a successor and a predecessor). The smallest time-stamp system (resp. interpolation system) of order 2 is thus the directed cycle C_3 on three vertices.

The following properties will be useful in the next sections.

Proposition 2.4 [5] *A graph G is a time-stamp system of order k if and only if for any vertex x in $V(G)$ the subgraph G_x^+ is a time-stamp system of order $k-1$.*

This property allows us to establish a lower bound on the number of vertices in a time-stamp system of order k :

Corollary 2.5 [2, 5] *If G is a time-stamp system of order k, then $|V(G)| \geq 2^k - 1$.*

This lower bound can be reached for $k = 2$ (the directed cycle C_3) and $k = 3$ (see the graph QR_7 in section 5). For $k = 4$, it can be proved that there is no time-stamp system with 15 vertices. The optimal graph in this case has 16 vertices and is due to Tromp [12] (see section 6).

A similar property can be derived for interpolation systems :

Proposition 2.6 *A graph G is an interpolation system of order k if and only if for any vertex x in $V(G)$ the subgraphs G_x^+ and G_x^- are both interpolation systems of order $k-1$.*

Proof. Let us first show that if G is an interpolation system of order k then for any $x \in V(G)$, G_x^+ is an interpolation system of order $k-1$. Let $(y_1, y_2, \ldots, y_p)$ be an ordered sequence in G_x^+ having at most $k-2$ elements ; then $(x, y_1, y_2, \ldots, y_p)$ is an ordered sequence in G having at most $k-1$ elements which can be interpolated. For any interpolation position i, $2 \leq i \leq p+1$, the inserted vertex belongs to G_x^+. Hence, the sequence $(y_1, y_2, \ldots, y_p)$ can be interpolated in G_x^+. In a similar way, it can be shown that G_x^- is also an interpolation system of order $k-1$. Conversely, let $(x_1, x_2, \ldots, x_q)$ be any ordered sequence in G having at most $k-1$ elements. Since $G_{x_1}^+$ (resp. $G_{x_q}^-$) is an interpolation system of order $k-1$, this sequence can be interpolated in any position i, $2 \leq i \leq q+1$ (resp. $1 \leq i \leq q$). □

Note that for interpolation systems, we do not have up to now a better lower bound than for time-stamp systems ($2^k - 1$ vertices).

In the following sections we introduce different construction mechanisms for time-stamp systems and interpolation systems.

3 The lexicographic product

This method is well-known in graph theory and leads to solutions for time-stamp systems.

Definition 3.1 Let G and H be two graphs : the *lexicographic product* of G and H, denoted by $G \otimes H$, is the graph whose set of vertices is $V(G) \times V(H)$ and whose set of arcs is given by :

$$((x, y), (x', y')) \in A(G \otimes H) \iff (x, x') \in A(G) \text{ or } (x = x' \text{ and } (y, y') \in A(H)).$$

Proposition 3.2 [2, 5] *If G and H are two time-stamp systems of respective orders k and ℓ then $G \otimes H$ is a time-stamp system of order $k + \ell - 1$.*

Example 3.3 Figure 1 depicts the graph $C_3 \otimes C_3$. The double-arrows linking two consecutives copies of C_3 stand for an arc from any vertex of one copy towards any vertex of the following copy. The graph C_3 satisfies the property (E_2), the graph $C_3 \otimes C_3$ thus satisfies the property (E_3) : one can easily check that every ordered sequence having at most 2 elements (that is every vertex and every arc) has a successor.

This construction allows us to obtain time-stamp systems of any order : by applying $k-2$ times this construction to the graph C_3, we obtain a time-stamp system of order k having 3^{k-1} vertices. However, this construction cannot be used for interpolation systems as shown by the following remark.

Remark 3.4 The graph C_3 satisfies the property (I_2) but the graph $C_3 \otimes C_3$ does not satisfy the property (I_3). One can see for instance that the ordered sequence $(0, 1)$ cannot be interpolated in position 2 : 0 does not have any successor which is a predecessor of 1.

4 Zielonka's construction

This construction has been proposed by Zielonka [13] as a generalization of a construction initially introduced by Lamport [6], and gives an optimal solution for time-stamp systems in the signed case. We prove that the graphs thus obtained also give solutions for interpolation systems. Since every interpolation system is a time-stamp system, those solutions are also optimal in the signed case.

Definition 4.1 Let k be a strictly positive integer : the *Zielonka graph* of order k, denoted by Z_k, is given by :

(*i*) $V(Z_k) = \{(\alpha, x_1, \ldots, x_k) \in \{1, 2, \ldots, k\} \times \{0,1\}^k / x_\alpha = 0\}$

(*ii*) $$((\alpha, x_1, \ldots, x_k), (\beta, y_1, \ldots, y_k)) \in A(Z_k) \iff \begin{cases} (\alpha < \beta \text{ and } x_\beta = y_\alpha) \\ \text{or} \\ (\alpha > \beta \text{ and } x_\beta \neq y_\alpha). \end{cases}$$

Note that the graph Z_k has exactly $k \times 2^{k-1}$ vertices.

Proposition 4.2 *The graph Z_k is an interpolation system of order k.*

Proof. Note first that there is no arc between any two vertices having the same first component. Thus, any ordered sequence $S = (s_1, s_2, \ldots, s_p)$ in Z_k is such that the first components of its vertices are pairwise distinct. If S has at most $k-1$ elements then there exists a first component, say α, which does not appear in S. In order to interpolate S in position i, $1 \le i \le p+1$, it suffices to insert the vertex $x = (\alpha, x_1, x_2, \ldots, x_k)$ given by :

(*i*) $x_\alpha = 0$,

(*ii*) if no element in S has β as first component then $x_\beta = 0$,

(*iii*) if $s_j = (\beta, y_1, y_2, \ldots, y_k) \in S$ then :

- if $j < i$, $x_\beta = y_\alpha$ if $\beta < \alpha$, $x_\beta = 1 - y_\alpha$ otherwise,
- if $j \ge i$, $x_\beta = 1 - y_\alpha$ if $\beta < \alpha$, $x_\beta = y_\alpha$ otherwise. □

5 Rotational tournaments

The notion of time-stamp system is related to a property of tournaments, stronger than (E_k), initially considered by Schütte and Erdős [3, 8] : a tournament T satisfies the property (S_k) is every set having at most $k-1$ vertices in T has a successor (in case of time-stamp systems, only ordered sequences are considered). Szekeres and Szekeres [10] have proved that such a tournament must have at least $2^{k-2}(k+1)-1$ vertices ; they also gave in their paper two sample tournaments having 7 and 19 vertices which respectively satisfy the properties (S_3) and (S_4). By using some results from group theory, Graham et Spencer [4] gave a construction which, for any value of k, leads to a tournament satisfying the property (S_k).

Every tournament satisfying the property (S_k) is obviously a time-stamp system of order k. However, the converse is not necessarily true : the graph $C_3 \otimes C_3 \otimes C_3$ is a tournament which does not satisfy the property (S_4) although it satisfies the property (E_4) by construction.

We show in this section that the tournaments introduced by Graham and Spencer are also interpolation systems.

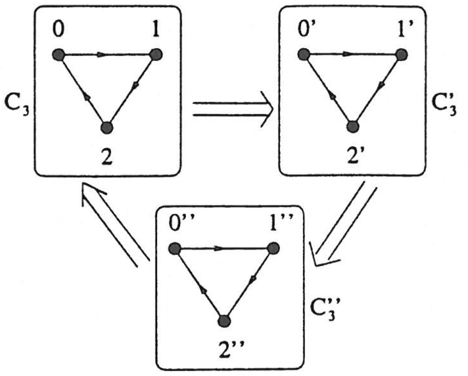

Figure 1: The graph $C_3 \otimes C_3$

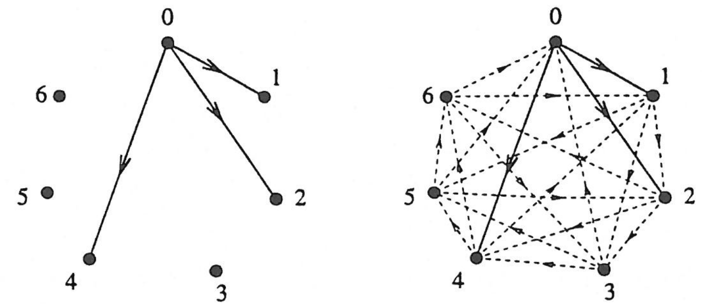

Figure 2: The tournament obtained from the quadratic residues of 7

Definition 5.1 Let p be a prime, congruent to 3 modulo 4. The rotational tournament QR_p, obtained from the quadratic residues of p [7] is defined by :

(i) $V(QR_p) = \{0, 1, \ldots, p-1\}$,

(ii) $A(QR_p) = \{(i, j) \ / \ j - i$ is a non-zero quadratic residue of $p\}$.

One can check that this construction leads to an antisymmetric directed graph : since p is congruent to 3 modulo 4, if $j - i$ is a non-zero quadratic residue of p then $i - j$ is not [7].

Example 5.2 The tournament depicted in Figure 2 is QR_7. The non-zero quadratic residues of 7 are $1, 2, 4$. These are the successors of 0. The successors of any vertex are deduced from the successors of 0 by applying a rotation. It is not difficult to check that the graph QR_7 satisfies the property (S_3) (and thus (E_3)) as well as the property (I_3).

In [4] Graham and Spencer proved that for any k, there exists an integer N_k such that every tournament of QR_p type, with $p > N_k$, satisfies the property (S_k). The proof of this result can be extended to interpolation systems [1] :

Theorem 5.3 *If p is a prime congruent to 3 modulo 4, $p > (k-1)^2 2^{2k-4}$, then the tournament QR_p is an interpolation system of order k.*

Smaller values of p may lead to tournaments which are also interpolation systems : we have seen that $QR_3 = C_3$ satisfies the property (I_2) and that QR_7 satisfies the property (I_3). One can also check for instance that QR_{19} satisfies the property (I_4), that QR_{47} satisfies the property (I_5) and that QR_{271} satisfies the property (I_6).

6 Tromp's construction

This method was initially proposed by Tromp [12] for building a time-stamp system of order 4. This construction is based on the graph QR_7 and leads to a solution with 16 vertices. This construction can in fact be applied to any graph. We show in this section that this construction can be used to obtain time-stamp systems and interpolation systems of any order.

Definition 6.1 Let G be a graph, and $\overline{G}$ an isomorphic copy of G. The "Tromp's construction" applied to G, denoted by $Tr(G)$, is the graph obtained as follows :

(i) $V(Tr(G)) = V(G) \cup V(\overline{G}) \cup \{w, \overline{w}\}$

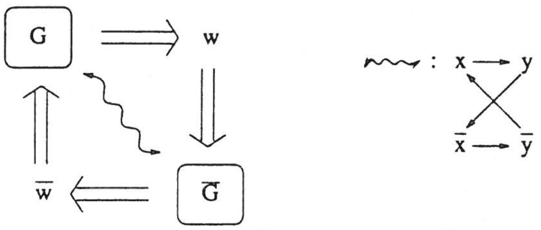

Figure 3: Tromp's construction

(ii) $\forall\, x \in V(G),\ (x,w),(\overline{w},x),(w,\overline{x}),(\overline{x},\overline{w}) \in A(Tr(G)),$

(iii) $\forall\, x,y \in V(G),\ (x,y) \in A(G) \implies (x,y),(\overline{x},\overline{y}),(y,\overline{x}),(\overline{y},x) \in A(Tr(G)).$

By construction, the graph $Tr(G)$ satisfies in particular the following property :

$$\forall\, x \in V(G) \cup \{w\},\quad Tr(G)^+_x = Tr(G)^-_{\overline{x}} \text{ and } Tr(G)^-_x = Tr(G)^+_{\overline{x}}.$$

If the starting graph G is an interpolation system of order k and if for every vertex x in $Tr(G)$ the subgraph $Tr(G)^+_x$ is isomorphic to G itself, denoted by $Tr(G)^+_x \sim G$, then the graph $Tr(G)$ thus obtained is an interpolation system of order $k+1$ (proposition 2.6). It is then interesting to characterize those graphs G which satisfy such a property. This is namely the case for the rotational tournaments introduced in the previous section :

Proposition 6.2 *Let p be a prime congruent to 3 modulo 4. The graph $Tr(QR_p)$ is such that :*

$$\forall\, x \in V(Tr(QR_p)),\ Tr(QR_p)^+_x \sim QR_p.$$

Proof. (sketch of) We first prove that the graph $Tr(QR_p)$ is vertex-transitive, that is for any vertices x and y there exists an automorphism of $Tr(QR_p)$ which maps x onto y. It is then sufficient to prove the desired result for one vertex in $Tr(QR_p)$, which is immediate for the vertex $\overline{w}$. □

We then obtain :

Corollary 6.3 *Let p be a prime congruent to 3 modulo 4. If QR_p is an interpolation system of order $k-1$, then $Tr(QR_p)$ is an interpolation system of order k.*

This result gives in particular time-stamp systems (and interpolation systems) of orders 5 and 6 which improve the previously known constructions. By using the lexicographic product this also improves the time-stamp systems upper bounds for the orders from 8 to 11.

7 Concluding remarks

The table in Figure 4 gives the best known results for time-stamp systems and interpolation systems (in the unsigned case). A minus sign in a given entry indicates that we do not know in this case any construction achieving a better bound than Zielonka's one. In particular, note that for $k \geq 12$ we do not know any (unsigned) time-stamp system having less vertices than the corresponding Zielonka's graph. For interpolation systems this is true as soon as $k > 6$ (the lexicographic product is not a valid construction for interpolation systems) : if we apply Tromp's construction to the graph QR_{271} (which satisfies the property (I_6)), we

Order k	Time-stamp systems #vertices	Time-stamp systems Graph	Interpolation systems #vertices	Interpolation systems Graph	Z_k
2	3	C_3	3	C_3	4
3	7	QR_7	7	QR_7	12
4	16	$Tr(QR_7)$	16	$Tr(QR_7)$	32
5	40	$Tr(QR_{19})$	40	$Tr(QR_{19})$	80
6	96	$Tr(QR_{47})$	96	$Tr(QR_{47})$	192
7	256	$Tr(QR_7) \otimes Tr(QR_7)$	-		448
8	640	$Tr(QR_{19}) \otimes Tr(QR_7)$	-		1024
9	1536	$Tr(QR_{47}) \otimes Tr(QR_7)$	-		2304
10	3840	$Tr(QR_{47}) \otimes Tr(QR_{19})$	-		5120
11	9216	$Tr(QR_{47}) \otimes Tr(QR_{47})$	-		11264
12	24576	$Tr(QR_{47}) \otimes Tr(QR_7) \otimes Tr(QR_7)$	-		24576
13	-		-		53248
$\vdots$					

Figure 4: Table of best known solutions

obtain a graph having 544 vertices.
Note that up to $k = 6$, the best known solutions for time-stamp systems are also solutions for interpolation systems, although property (I_k) is stronger than property (E_k).

References

[1] V. Albiero, E. Sopena, *Systèmes d'estampillage et systèmes d'interpolation*, Université Bordeaux I, Technical Report LaBRI n° 991-94 (1994).

[2] R. Cori, E. Sopena, *Some combinatorial aspects of time-stamp systems*, Europ. J. Combinatorics **14** (1993), 95-102.

[3] P. Erdös, *On a problem in graph theory*, Math. Gaz. **47** (1963), 220-223.

[4] R. L. Graham, J. H. Spencer, *A constructive solution to a tournament problem*, Canad. Math. Bull. **14** (1971), 45-48.

[5] A. Israëli, M. Li, *Bounded time-stamps*, Distributed Computing **6** (1993), 205-209.

[6] L. Lamport, *The mutual exclusion problem (II) : statement and solutions*, J. ACM **33-2** (1986), 327-348.

[7] W. J. Leveque, *Topics in number theory*, Vol. I, Addison-Wesley, Reading, Mass., (1954).

[8] J. W. Moon, *Topics on tournaments*, Holt, New York (1968).

[9] M. Saks, F. Zaharoglou, *Optimal space distributed move-to-front lists*, Comm. ACM **439** n° 2 (1991), 65-73.

[10] E. Szekeres, G. Szekeres, *On a problem of Schütte and Erdős*, Math. Gaz. **49** (1965), 290-293.

[11] J. Tromp, *On update-last schemes*, Parallel Processing Letters **3** n° 1 (1993), 25-28.

[12] J. Tromp, *Unpublished manuscript.*

[13] W. Zielonka, *Time-stamp systems for a fixed set of agents*, Université Bordeaux I, Technical Report LaBRI (1990).