1992

J.T. Tromp

On update-last schemes

Computer Science/Department of Algorithmics and Architecture   Report CS-R9247 November

CWI is the research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a non-profit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands organization for scientific research (NWO).

On Update-Last Schemes

John Tromp
CWI
P.O. Box 4079, 1009 AB
Amsterdam, The Netherlands

Abstract

This paper defines the notion of an Update-Last scheme as a method of distributing the identity of a 'last' among n objects over their labels. A precise characterization of the possibility of such a scheme is given in terms of the number of values each label can assume.

1991 Mathematics Subject Classification: 05A05
1991 CR Categories: D.1.3, D.2.7, D.4.2, G.2.1
Keywords and Phrases: combinatorial problems, data management, distributed systems
Note: This work was supported in part by NFI project Aladdin while the author was visiting the Technion.
This paper is submitted for publication in Parallel Processing Letters (PPL).

1 Introduction

Let $X = X_1 \times \cdots \times X_n$ be the product of n domains, one for each $i \in I = \{1, \ldots, n\}$. Elements of the X_i are called *labels*, and those of X *label vectors*. We say that two label vectors l and l' are i-equivalent, $l \equiv_i l'$, whenever $\forall_{j \neq i} : l_j = l'_j$.

Definition 1 *An Update-Last scheme is a partial function last* : $X \rightarrow I$ *with non-empty domain* $U \subseteq X$ *such that*

$$\forall\, l \in U, i \in I\, \exists\, l' \in U : l \equiv_i l' \wedge last(l') = i.$$

One can think of an Update-Last scheme as providing a method whereby each of a set of n objects can be made a 'leader' by choosing its label appropriately. The objects can be either active entities that carry out the label-inspection and choice-making themselves, or passive objects in some system that wants to keep track of which object is special. The vector of labels can be seen as a way of storing an index in a distributed fashion.

The above formalization is intended to capture the essence of such methods, which may appear in different forms and shapes. The possibility of f not being total accomodates methods where not all possible label combinations can arise.

This paper derives exact bounds on the space complexity of an Update-Last scheme, in terms of the domain sizes $|X_i|$.

Report CS-R9247
ISSN 0169-118X
CWI
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

The main motivation for this work comes from the implementation of wait-free multi-writer registers from single-writer ones. Update-Last schemes immediately provide for *serial* implementations that work correctly as long as no two operations overlap: a writer tags each new value with a label that shows this value to be the last-written or current one, while a reader simply collects all value-label pairs and returns the value of the last writer. The exact bounds proved here on label domain sizes thus yield a lower bound on the space complexity of real concurrent implementations.

2 Related Work

Li and Vitanyi [4] present an Update-Last scheme where each X_i is of size n, and argue that for the purpose of implementing multi-writer schemes, one does not need the full functionality of so-called *time-stamp schemes* [3], where any two labels can be compared to find the last. Such schemes are known to require much larger label domains, of size at least 2^n.

The unlabeled (non-indexed) case, where *last* gets as input a set of n different labels from a single domain, and maps to one of them, was considered by Cori and Sopena [2]. They proved a tight bound of $2n-1$ on the size of the label domain. The proof of our main Theorem was in part inspired by their proof.

Another surprising application is in the simulation of a DFA by a (fully-connected) asynchronous cellular automaton [1]. Such an ACA has one node for each letter in the alphabet, which is activated when that letter appears in the input. Upon activation, it changes its state according to the states of all its neighbours.

In the simulation, part of a node's state is the state of the simulated DFA, and an Update-Last scheme allows an activated node to identify the node holding the current DFA state.

3 Characterizing Update-Last schemes

Theorem 1 *There exists an Update-Last scheme with domain X iff $\sum_{i\in I} 1/|X_i| \leq 1$.*

Proof.

$\Rightarrow$ Let U_i be $U/\equiv_i$ and U' be the disjoint union of all U_i. Each element in U_i is an equivalence class of label vectors differing only in component i. Such a class can also be thought of as a label vector in which component i has been blanked.

For a fixed i, consider the number of pairs $(l,x) \in U \times U_i$ with $l \in x$. There is one such pair for each $l \in U$ and hence $|U|$ in total. On the other hand, for an $x \in U_i$, there are at most $x_i \stackrel{\text{def}}{=} |X_i|$ such pairs. With $e_i \stackrel{\text{def}}{=} |U_i|$, this gives $|U| \leq e_i x_i$, or $1/x_i \leq e_i/|U|$. By definition of U', we also have that $|U'| = \sum_{i\in I} e_i$. Thus,

$$\sum_{i\in I} 1/x_i \leq \sum_{i\in I} e_i/|U| = |U'|/|U|.$$

Given that $last : X \to I$ is an Update-Last scheme, there exists a function $new : U' \to U$ satisfying

$$\forall x \in U_i : new(x) \in x \wedge last(new(x)) = i.$$

Note that *new* is invertible (one-one), since $new^{-1}(l)$ is just the equivalence class of l w.r.t. $\equiv_{last(l)}$. Hence $|U'| \leq |U|$ and the result follows.

$\Leftarrow$ W.l.o.g. assume that $X_i = \{0,1,\ldots,x_i-1\}$. Given that $\sum_{i\in I} 1/|X_i| \leq 1$, we can partition the $[0,1)$ interval into n disjoint half-open intervals such that i's interval has length at least $1/x_i$. Now define the total function *last*(l) to be the index whose interval contains $\sum_{i\in I} l_i/x_i \bmod 1$ ($x \bmod 1$ denotes the fractional part of x). Since, for $r = \sum_{j\neq i} l_j/x_j$, the set of numbers $\{(r + l_i/x_i) \bmod 1\}_{l_i\in X_i}$ has a non-empty intersection with i's interval, it is clear that *last* is an Update-Last scheme. ■

Putting $y_i = 1/|X_i|$, the condition of the Theorem becomes $\sum_{i\in I} y_i \leq 1$. By standard convexity arguments, $\prod_{i\in I} y_i$ is maximal when all y_i equal $1/n$. This proves the following lower bound on the number of label configurations:

COROLLARY 2 *An Update-Last scheme with domain X satisfies* $\prod_{i\in I} |X_i| \geq n^n$.

This proves the optimality of the construction presented in [4], which is essentially an Update-Last scheme with $|X_i| = n, i = 1, \ldots, n$.

4 Further Work

There are several directions in which Update-Last schemes can be generalized. One would be a scheme where i can choose its label in a way that makes j become last, for all $i, j \in I$. Theorem 1 shows a way to do this when all $|X_i|$ equal n, but in general the condition $\sum_{i\in I} 1/|X_i| \leq 1$ will not be sufficient any more.

More interesting, perhaps, is the study of general schemes where i can choose its label so as to satisfy some constraint on a function of all the labels, where this function might represent the state of a shared object.

5 Acknowledgements

Discussions with Amos Israeli and Ted Herman were most helpful.

References

[1] R. Cori, Y. Metivier, W. Zielonka, " Asynchronous mappings and asynchronous cellular automata", technical report 89-97, LaBRI, December 1989.

[2] R. Cori, E. Sopena, "Some Combinatorial aspects of Time Stamp Systems", submitted to Journal of Algorithms.

[3] A. Israeli, M. Li, "Bounded Time-Stamps", Proceedings of the 28th IEEE Symposium on Foundations of Computer Science, pp. 371–382, 1987.

[4] M. Li, P.M.B. Vitányi, "Optimality of Wait-Free Atomic Multi-Writer Variables", technical report CS-R9128, CWI, June 1991.