A Note on Synchronization Steps in Firing Squad Synchronization Problem

Akira Nomura*, Jean-Baptiste Yunès†, Hiroshi Umeo‡

*Univ. of Osaka Electro-Communication, Neyagawa-shi, Hatsu-cho, 18-8, 572-8530, Osaka, Japan
nomura@cyt.osakac.ac.jp

†LIAFA - Université Paris 7 Denis Diderot, 175, rue du chevaleret, 75013 Paris, France
Jean-Baptiste.Yunes@liafa.jussieu.fr

‡Univ. of Osaka Electro-Communication, Neyagawa-shi, Hatsu-cho, 18-8, 572-8530, Osaka, Japan
umeo@cyt.osakac.ac.jp

Abstract—The firing squad synchronization problem on cellular automata has been studied extensively for more than forty years, and a rich variety of synchronization algorithms have been proposed. In this paper, we propose two synchronization algorithms and their implementations, each having been implemented on a cellular automaton, including a self-reproducing cellular automata. See Figure 1. The array operates in lock-step mode such that the next state of each cell (except the end cells) is determined by both its own present state and the present states of its right and left neighbors. All cells (soldiers), except one general cell, are initially in the quiescent state at time $t = 0$ and have the property whereby the next state of a quiescent cell having quiescent neighbors is the quiescent state. At time $t = 0$ the general cell is in the fire-when-ready state, which is an initiation signal to the array. The FSSP is stated as follows: Given an array of $n$ identical cellular automata, including a general on the left end which is activated at time $t = 0$, we want to give the description (state set and next-state transition function) of the automata so that, at some future time, all of the cells will simultaneously and, for the first time, enter a special firing state. The initial general is on the left end of the array.

Figure 1. A one-dimensional cellular automaton.

B. A Seven-state Implementation of $O(n^2)$-step Synchronization Algorithm

In this section, we propose a seven-state cellular automaton that can synchronize for any array of length $n$ in $O(n^2)$ steps.

The algorithm is to find a center cell (cells) of the cellular space of length $n$ in $3/4n^2 + O(1)$ steps. Figure 2 shows a space-time diagram for finding the center cell. We have implemented the algorithm on a cellular automaton $M$ with seven internal states $Q = \{Q, G, F, A, B, C, D\}$, where $Q$ is the quiescent, $G$ is the general, and $F$ is the firing state,
respectively. The other states are used for auxiliary states. The state transition table is given in Fig. 3. In the table a symbol "*" means a boundary state of right and left ends of arrays. Some snapshots for the synchronization processes on 11 cells can be found in Fig. 4.

Figure 2. A space-time diagram for the synchronization algorithm.

Figure 3. Transition table for the seven-state implementation

III. CONCLUSION

We have proposed a synchronization algorithm and given its 7-state implementation having \( O(n^2) \) synchronization steps. By the similar way above, we can develop an algorithm with \( O(2^n) \) synchronization steps. Its implementation can be also given on a 15-state cellular automaton.

ACKNOWLEDGMENT

A part of this work of the last author is supported by Grant-in-Aid for Scientific Research (C) 21500023.

REFERENCES


