Constructing Counters through Evolution

Moshe Sipper

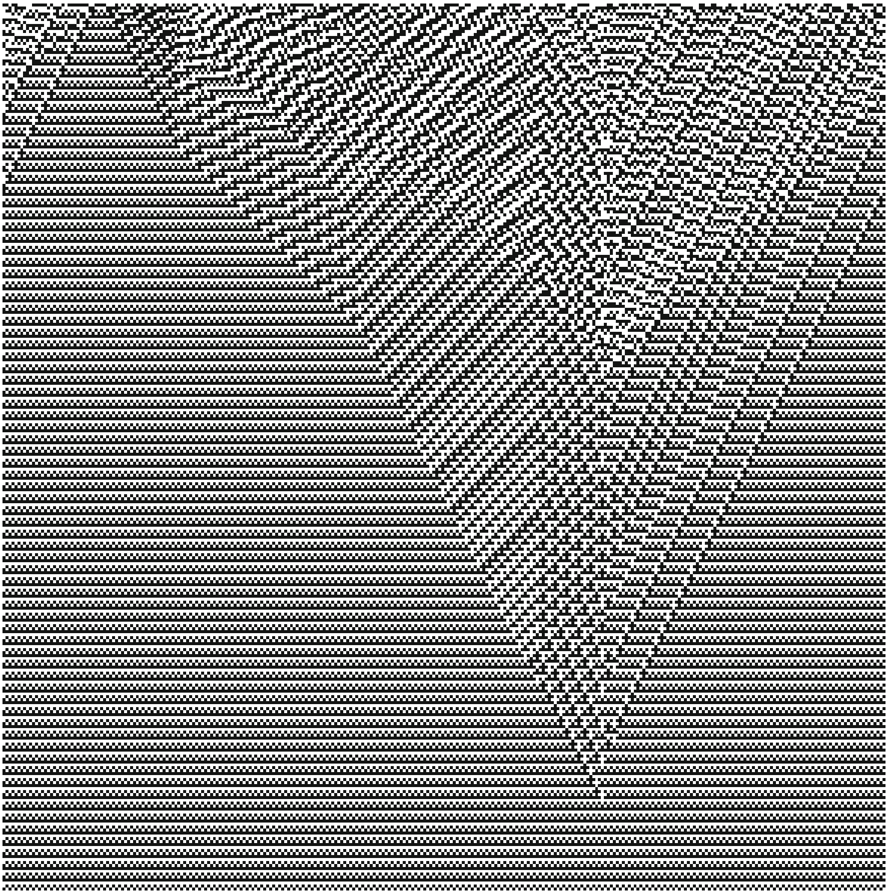
In the one-dimensional synchronisation task, discussed in [147], the final pattern consists of an oscillation between all 0s and all 1s. From an engineering point of view, this period-2 cycle may be considered a 1-bit counter. Building upon such an evolved CA, using a small number of different cellular clock rates, 2- and 3-bit counters can be constructed.

Constructing a 2-bit counter from a non-uniform, radius $r = 1$ CA, evolved to solve the synchronisation task, is carried out by "interlacing" two $r = 1$ CAs, in the following manner: each cell in the evolved $r = 1$ CA is transformed into an $r = 2$ cell, two duplicates of which are juxtaposed (the resulting grid's size is thus doubled). This transformation is carried out by "blowing up" the $r = 1$ rule table into an $r = 2$ one, creating from each of the (eight) $r = 1$ table entries four $r = 2$ table entries, resulting in the 32-bit $r = 2$ rule table. For example, entry $110 \rightarrow 1$ specifies a next-state bit of 1 for an $r = 1$ neighbourhood of 110 (left cell is in state 1, central cell is in state 1, right cell is in state 0). Transforming it into an $r = 2$ table entry is carried out by "moving" the adjacent, distance-1 cells to a distance of 2, i.e., $110 \rightarrow 1$ becomes $1X1Y0 \rightarrow 1$; filling in the four permutations of (X,Y), namely, $(0,0)$, $(0,1)$, $(1,0)$, and $(1,1)$, results in the four $r = 2$ table entries. The clocks of the odd-numbered cells function twice as fast as those of the even-numbered cells, meaning that the latter update their states every second time step with respect to the former. The resulting CA converges to a period-4 cycle upon presentation of a random initial configuration, a behaviour that may be considered a 2-bit counter.

Constructing a 3-bit counter from a non-uniform, $r = 1$ CA is carried out in a similar manner, by "interlacing" three radius $r = 1$ CAs (the resulting grid's size is thus tripled). The clocks of cells $0,3,6,\ldots$ function normally, those of cells $1,4,7,\ldots$ are divided by two (i.e., these cells change state every second time step with respect to the "fast" cells), and the clocks of cells $2,5,8,\ldots$ are divided by four (i.e., these cells change state every fourth time step with respect to the fast cells). The resulting CA converges to a period-8 cycle upon presentation of a random initial configuration, a behaviour that may be considered a 3-bit counter. We have thus demonstrated how one can build upon an evolved behaviour in order to construct devices of interest.

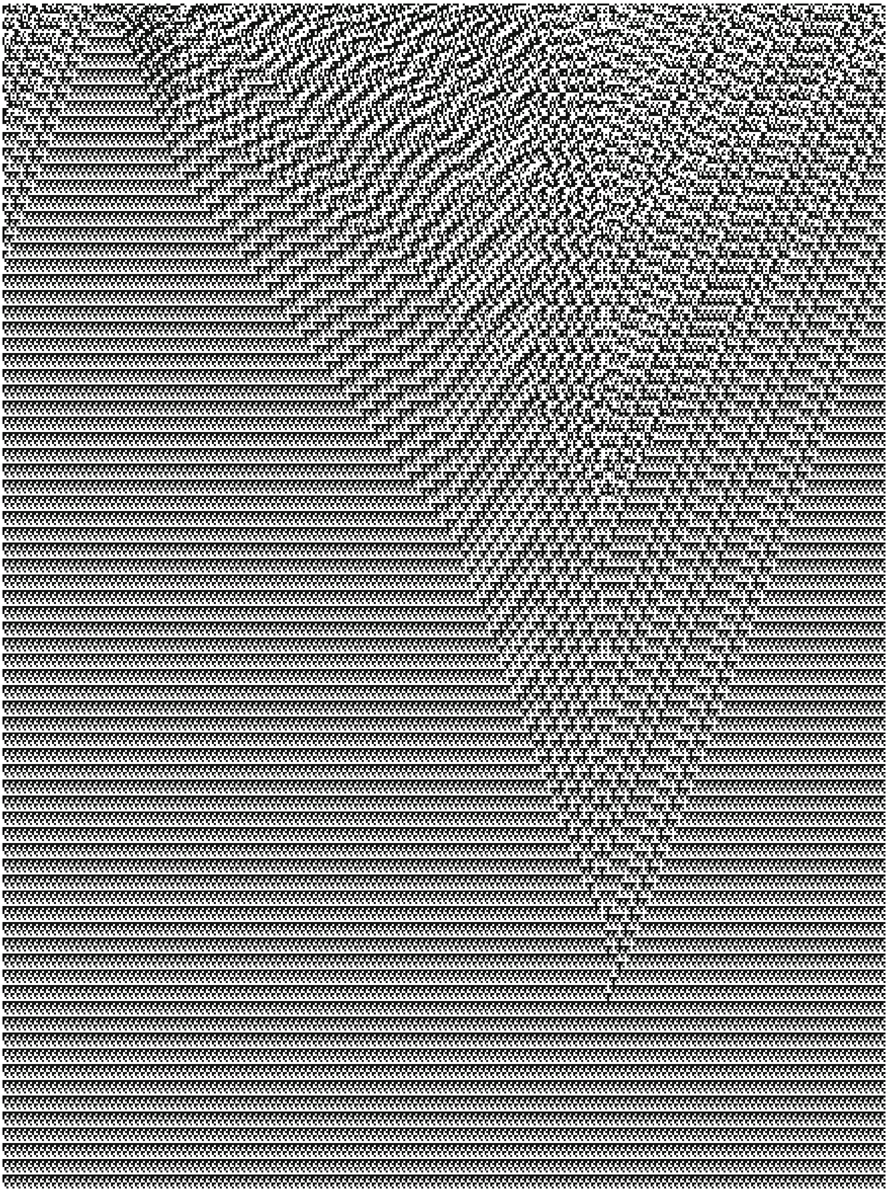
M. Sipper
Ben-Gurion University, Beer Sheva, Israel
e-mail: sipper@cs.bgu.ac.il

A. Adamatzky and G.J. Martínez (eds.), *Art of Cellular Automata*,
Emergence, Complexity and Computation 20,
DOI: 10.1007/978-3-319-27270-2_10



2-bit counter

The one-dimensional synchronisation task: A 2-bit counter. Operation of a non-uniform, 2-state CA, with connectivity radius $r = 2$. Grid size is $N = 298$. The CA converges to a period-4 cycle upon presentation of a random initial configuration, a behaviour that may be considered a 2-bit counter. [147].



3-bit counter ©2015 Moshe Sipper.
The one-dimensional synchronisation task: A 3-bit counter. Operation of a non-uniform, 2-state CA, with connectivity radius $r = 3$. Grid size is $N = 447$. The CA converges to a period-8 cycle upon presentation of a random initial configuration, a behaviour that may be considered a 3-bit counter [147].

sipper@cs.bgu.ac.il