Complex Systems **3** (1989) 343–348

One-Dimensional Cellular Automata: Injectivity From Unambiguity

Tom Head
Department of Mathematical Sciences, State University of New York, Binghamton, New York 13901, USA

Abstract. New algorithms for deciding the injectivity of the global update function associated with a cellular automaton CA of dimension one are presented. This is done by interpreting each ordered pair determined by the local update function as the edge of a labeled directed graph GR which has the property that each bi-infinite sequence of states of the cellular automaton is the sequence of input labels of one and only one bi-infinite path in the graph. For an appropriate conversion of GR into a finite automaton, injectivity of the global update function of CA on the set of pseudofinite sequences is equivalent to the unambiguity of this automaton. For appropriate conversions of GR into a finite set of finite automata, the injectivity of the global update function on all sequences is equivalent to the condition that every automaton in the finite set be unambiguous.

1. Introduction and significance

Cellular automata are under extensive study as models of complex systems as shown in the volumes by Wolfram [1] and Toffoli and Margolus [2]. The property of reversibility possessed by some, but not all, such automata has received special concern [3,2, Ch. 14]. A cellular automaton cannot be reversible unless its global update function is injective. Injectivity and reversibility have been studied by Amoroso and Patt [4] and by Richardson [5]. Culik has given a particularly broad treatment for the one-dimensional case of reversibility, injectivity, and their relationship in [6]. It is known from these works that there exist procedures for deciding, for any given one-dimensional cellular automaton, whether the global update function is injective and also whether the restriction of this function to the set of pseudofinite bi-infinite sequences is injective.

The purpose of this article is to present new algorithms for deciding these two forms of injectivity. *Our algorithms reduce each question to the testing of the unambiguity of finite automata.* Testing unambiguity is quite elementary: one method is to test whether the language recognized by an associated finite automaton is empty [7]. Theorems 1 and 2 of this paper are closely related to the same numbered theorems of [6]. The details of the constructions used here allow the resolution of decisions into testing unambiguity rather than single-valuedness of transducers.

The labeled graphs constructed in section 2 have a history: examples appear in [8] and also in [1, pp. 199, 203] where they are said to be analogous to de Bruijn graphs. We have adapted them from [9] where a more general construction for Lindenmayer systems with interactions was applied.

2. Constructing a labeled graph from a cellular automaton

Let $CA = (S, r, L)$ be a linear (i.e., one-dimensional) cellular automaton having S as its finite set of *states*, r as its *context radius*, and $L : S^{2r+1} \to S$ as its *local (update) function*. We assume throughout that $r > 0$ since our problems have trivial solutions for $r = 0$. *Let K be the cardinal number of S.* From CA a directed labeled graph $GR = GR(CA) = (V, E)$ will be constructed. The set V of vertices will have cardinal number K^{2r} and the set E of edges will have cardinal K^{2r+1}. $GR(CA)$ is merely an alternate formalism for CA, but it is a convenient tool for subsuming the theory of *linear* cellular automata into the theory of finite automata.

For each word $w = s(0)\ldots s(2r)$ in the domain of L we place $q_{s(0)\ldots s(2r-1)}$ and $q_{s(1)\ldots s(2r)}$ in V and the labeled edge $(q_{s(0)\ldots s(2r-1)}, s(r), L(w), q_{s(1)\ldots s(2r)})$ in E. The label of this edge is the ordered pair $(s(r), L(w))$. We may think of $s(r)$ and $L(w)$ as the *input and output symbols* associated with this edge. Later we will convert GR into a finite transducer (definitions appear in [10] and [11]) by choosing initial and terminal sets of states. A crucial fact about GR is that *the input symbol of each edge is redundant* since it is the $(r+1)$st symbol of the subscript of the left vertex (and the rth symbol of the subscript of the right vertex). In sections 3 and 4, automata will be constructed from GR by deleting these redundant input symbols. The set of edges incident at each vertex is easily described: at each vertex $q_{s(0)\ldots s(2r-1)}$ there are exactly K incoming edges. *All incoming edges have the same input symbol*, namely $s(r-1)$. These edges come from the states $q_{ts(0)\ldots s(2r-2)}$ for t in S. Likewise *all outgoing edges have the same input symbol*, namely $s(r)$. These edges go to the K states $q_{s(1)\ldots s(2r-1)t}$ for t in S. The fact that L is a function, rather than merely a relation, has certainly not made GR deterministic. It has, however, made GR unambiguous as detailed in the next proposition. A function $P : Z \to E$ is a (bi-infinite) *path* in GR if, for each i in Z, the terminal vertex of the edge $P(i)$ is the initial vertex of the edge $P(i+1)$.

Proposition 1. *For each function $s : Z \to S$, there is exactly one path $P : Z \to E$ in $GR(CA)$ for which, for each i in Z, the input symbol of the edge $P(i)$ is $s(i)$.*

Proof. One such path is $P : Z \to E$ where $P(i) = (q_{s(i-r)\ldots s(i+r-1)}, s(i), L(s(i-r)\ldots s(i+r)), q_{s(i-r+1)\ldots s(i+r)})$. This path P is the only one meeting the requirement because, for each i in Z, the stated edge $P(i)$ is the only one that occurs as the $(r+1)$st edge in a path segment consisting of $2r+1$ edges having the string of input symbols $s(i-r)\dots s(i)\dots s(i+r)$.

3. Global updating of pseudofinite functions

The linear cellular automaton $CA = (s, r, L)$ defines a function $G : S^Z \to S^Z$ called the *global (update) function* of CA. Each element of S^Z is a function from the ring Z of integers into the state set S of CA. The definition of the global function G is based on the local function L: for each $s : Z \to S$, $G(s)[i] = L(s(i-r)\dots s(i)\dots s(i+r))$. Let us consider how $G(s)$ arises from s via the labeled graph $GR = GR(CA) = (V, E)$. There is precisely one path $P : Z \to E$ for which s is the input sequence. From P we obtain $G(s) : Z \to S$ as the function for which each $G(s)[i]$ is the output symbol of the edge $P(i)$. In section 4 the injectivity of G on the whole of the set S^Z will be treated. Here we treat injectivity of G on an important proper subset of S^Z, the pseudofinite functions defined below. For the present section only, we add additional structure to CA: let $CA = (S, r, L, D)$ where D is a distinguished state in S called *the dormant (or quiescent) state.* We demand that $L(D\dots D) = D$ which has the interpretation that a *dormant cell remains dormant as long as all cells within its context radius are dormant.* A function $s : Z \to S$ is *pseudofinite* if $s(i) = D$ for all but finitely many i in Z.

From the labeled graph $GR = GR(CA) = (V, E)$, we construct the finite transducer $TR = TR(CA) = (V, \{q_{D\dots D}\}, \{q_{D\dots D}\}, E)$ having state set V, edge set E, and the singleton $\{q_{D\dots D}\}$ as both the set of initial states and the set of terminal states. The alphabet of TR is S for both input and output. By deleting the input symbols from the edges of TR we construct the automaton $AU = AU(CA) = (V, \{q_{D\dots D}\}, \{q_{D\dots D}\}, E')$. Formally, $E' = \{(u, b, v)$ in $VXSXV$: there is a symbol a in S for which (u, a, b, v) is in $E\}$. Recall that a finite automaton (resp., transducer) is *unambiguous* if, for each string accepted, there is only one acceptance path. We consider that *in every automaton the null string is accepted by at most one path.*

Theorem 1. *Let CA be a linear cellular automaton. Let $AU = AU(CA)$ be the finite automaton constructed above from CA. The restriction of the global update function G of CA to the set of pseudofinite functions in S^Z is injective if and only if AU is unambiguous.*

Proof. Let AU be ambiguous. Then there is a word $w = s(1)\dots s(k)$ of length $k > 0$ for which there are two distinct paths from $q_{D\dots D}$ to $q_{D\dots D}$ having output label w. Let $P1$ and $P2 : Z \to E$ be the extensions of these two finite paths to bi-infinite paths in $GR(CA)$ extended as follows: $P_1(i) = P_2(i) = (q_{D\dots D}, D, D, q_{D\dots D})$ when either $i < 1$ or $i > k$. Then P_1 and P_2 are distinct paths in GR having the same output function, namely $g(i) = s(i)$ for $1 \le i \le k$ and D otherwise. Since they are distinct paths they have, by the proposition, distinct input functions f_1 and $f_2 : Z \to S$. Note that f_1, f_2, and g are all pseudofinite functions. The restriction of G to the set of pseudofinite functions is not injective since $G(f_1) = G(f_2) = g$.

Let AU be unambiguous. Let f_1, f_2, and g be pseudofinite functions for which $G(f_1) = G(f_2) = g$. By the proposition there is a unique path $P1 : Z \to E$ in $GR(CA)$ having f_1 as input function and a unique path $P2 : Z \to E$ having f_2 as input function. The output function of both P_1 and P_2 is g. Since f_1 and f_2 are pseudofinite there are integers j and k for which $P_1(i) = P_2(i) = (q_{D...D}, D, D, q_{D...D})$ for all $i < j$ and all $i > k$. Let R_1 and R_2 be the restrictions of P_1 and P_2 to the interval $[j,k] = \{i \text{ in } Z : j \leq i \leq k\}$. Then R_1 and R_2 are paths from the initial state to the terminal state of AU that have the same output label: for each i in $[j,k], R_1(i) = P_1(i)$; $R_2(i) = R_2(i)$; and both $P_1(i)$ and $P_2(i)$ have $g(i)$ as output symbol. From the unambiguity of AU it follows that $R_1 = R_2$. Consequently $P_1 = P_2$ and $f_1 = f_2$.

Remark 1. In theorem 1 of [6] the method of testing for injectivity on the set of pseudofinite functions consists of constructing a finite transducer T simulating G and then testing for single-valuedness of T^{-1}. It is known that testing for single-valuedness can be done in polynomial time [12], but we know of no easily implementable algorithm for this purpose. Our transducer TR can play the role of T. The specifics of the construction of our TR make TR unambiguous, from which it follows that TR^{-1} will be single-valued iff TR^{-1} is unambiguous. The unambiguity of TR^{-1} is equivalent to the unambiguity of our AU.

4. Injectivity of the global update function

Let $GR(CA)$ be the graph constructed from $CA = (S, r, L)$ in section 2. From $GR(CA)$ we construct a finite set of finite automata $AU_{(I,J)}(CA) = (V, I, J, E')$ each of which has alphabet S, state set V, and edge set $E' = \{(u, b, v) \text{ in } VXSXV$: there is an a in S for which (u, a, b, v) is in $E\}$. Specifying the set CHOICES from which the sets I and J of initial and terminal states are to be chosen requires careful preparation.

Recall that the cardinal of V is K^{2r}. Initialize the set CHOICES to be the empty set. For each of the $K^{2r}(K^{2r} - 1)/2$ unordered pairs $\{p, q\}$ of distinct vertices in V, place the doubleton $\{p, q\}$ in CHOICES if and only if the two automata $(V, \{p\}, \{p\}, E')$ and $(V, \{q\}, \{q\}, E')$ accept a common non-null word. For each vertex q that is not an element of any doubleton in CHOICES, place the singleton $\{q\}$ in CHOICES if and only if the automaton $(V, \{q\}, \{q\}, E')$ accepts a non-null word. The allowable choices for both the initial set I and terminal set J for our finite set of automata $AU_{(I,J)}$ are precisely the sets in CHOICES.

Theorem 2. *Let CA be a linear cellular automaton. Let $\{AU_{(I,J)} : I, J$ in CHOICES$\}$ be the finite family of automata constructed above from CA.*

The global update function G of CA is injective on its domain S^Z if and only if each of the automata $AU_{(I,J)}$ is unambiguous.

Proof. Let $AU = AU_{(I,J)} = (V, I, J, E')$ be ambiguous. Recall the construction of AU from $GR = (V, E)$ and especially of E' from E. Then there are states p_1 and p_2, not necessarily distinct, in I and states r_1 and r_2, not necessarily distinct, in J for which there are *distinct* paths in GR from p_1 to r_1 and from p_2 to r_2 having the same output label, $y = y(1)\dots y(k-1)$, in GR. At p_1 and p_2 there are loops that have the same output label, $x = x(m-1)\dots x(0)$, in GR. At r_1 and r_2 there are loops that have the same output label, $z = z(0)\dots z(n-1)$, in GR. It follows that there are *distinct* paths $P_1 : Z \to E$ and $P2 : Z \to E$ in GR with the *same* output function g, namely: for $i \leq 0, g(i) = x((-i)MOD\ m)$; for $1 \leq i \leq k-1$, $g(i) = y(i)$; and for $k \leq i$, $g(i) = z((i-k)MOD\ n)$. Let f_1 and f_2 be the input functions of P_1 and P_2. Since P_1 and P_2 are distinct paths in GR, it follows from the proposition that f_1 and f_2 are distinct. Then G is not injective since $G(f_1) = G(f_2) = g$.

Suppose that G is not injective. Then there are distinct functions f_1 and $f_2 : Z \to S$ for which $G(f_1) = G(f_2) = g$. Let k in Z be such that $f_1(k) \neq f_2(k)$. By the proposition there are unique paths P_1 and $P_2 : Z \to E$ in GR having input functions f_1 and f_2. The output function of both P_1 and P_2 is g. Recall that *there are only* $N = (K^{2r})^2$ *ordered pairs in* VXV. Consequently there are integers i_1 and j_1, $k - N \leq i_1 \leq j_1 \leq k$, for which the initial states of $P_1(i_1)$ and $P_1(j_1)$ coincide and the initial states of $P_2(i_1)$ and $P_2(j_1)$ coincide. Let p be the common initial state of $P_1(i_1)$ and $P_1(j_1)$. Let p' be the common initial state of $P_2(i_1)$ and $P_2(j_1)$. Since the two path segments $P_1(i_1)\dots P_1(j_1-1)$ and $P_2(i_1)\dots P_2(j_1-1)$ must have the same output string, we have $\{p, p'\}$ in CHOICES. Likewise there are integers i_2 and j_2, $k \leq i_2 \leq j_2 \leq k + N$, for which the terminal states of $P_1(i_2)$ and $P_1(j_2)$ coincide and the terminal states of $P_2(i_2)$ and $P_2(j_2)$ coincide. Let q be the common terminal state of $P_1(i_2)$ and $P_1(j_2)$. Let q' be the common terminal state of $P_2(i_2)$ and $P_2(j_2)$. Since the two path segments $P_1(i_2+1)\dots P_1(j_2)$ and $P_2(i_2+1)\dots P_2(j_2)$ must have the same output string, we have $\{q, q'\}$ in CHOICES. For $I = \{p, p'\}$ and $J = \{q, q'\}$ we observe that $AU_{(I,J)}$ is ambiguous: the path segments $P_1(j_1)\dots P_1(i_2)$ and $P_2(j_1)\dots P_2(i_2)$ share the same output string $g(j_1)\dots g(i_2)$ but they are distinct since $j_1 \leq k \leq i_2$ and the edges $P_1(k)$ and $P_2(k)$ have input symbols $f_1(k) \neq f_2(k)$.

A review of theorem 1 will confirm that it is virtually a special case of theorem 2: the conditions in theorem 1 allow the singleton $\{q_{D\dots D}\}$ to play the role CHOICES plays in theorem 2.

Remark 2. For a cellular automaton CA presented in the form of a labeled graph $Gr(CA)$, only three common elementary procedures suffice as the building blocks for all decisions used in this article: (1) decide whether a finite automaton accepts at least one word; (2) decide whether two finite automata accept a common word — which at worst can be done by constructing their product and applying (1); and (3) decide whether a finite automaton A is unambiguous — which at worst can be done by constructing a modified union of A and AXA as in [7] and applying (1). It may therefore be a simple matter to extend existing symbolic manipulation systems to include the injectivity decision procedures given here.

References

[1] S. Wolfram, *Theory and Applications of Cellular Automata* (World Scientific, Singapore, 1986).

[2] T. Toffoli and N. Margolus, *Cellular Automata Machines* (MIT Press, Cambridge, 1987).

[3] T. Toffoli, "Reversible computing," in *Automata, Languages and Programming*, J.W. Bakker and J. van Leeuwen, eds., 7th ICALP, *Lecture Notes in Computer Science*, **85**, (Springer-Verlag, 1980).

[4] S. Amoroso and Y.N. Patt, "Decision procedures for surjectivity and injectivity of parallel maps for tessellation structures," *Journal of Computer and System Sciences*, **6** (1972) 448–464.

[5] D. Richardson, "Tessellations with local transformations," *Journal of Computer and System Sciences*, **6** (1972) 373–388.

[6] K. Culik II, "On invertible cellular automata," *Complex Systems*, **1** (1987) 1035–1044.

[7] M. Blattner and T. Head, "Automata that recognize intersections of free monoids," *Information and Control*, **35** (1977) 173–176.

[8] I.J. Good, "Normal recurring decimals," *Journal of the London Mathematical Society*, **21** (1946) 167–169.

[9] T. Head, "Codeterministic Lindenmayer schemes and systems," *Journal of Computer and System Sciences*, **19** (1979) 203–210.

[10] J. Berstel, *Transductions and Context-Free Languages* (Teubner, Stuttgart, 1979).

[11] M. Harrison, *Introduction to Formal Language Theory* (Addison-Wesley, Reading, MA, 1978).

[12] E. Gurari and O. Ibarra, "A note on finite-valued and finitely ambiguous transducers," *Mathematical Systems Theory*, **16** (1983) 61–66.