Constrained Eden

Steve Seif

Mathematics Department
University of Louisville
Louisville, KY 40292, USA
swseif01@louisville.edu

Let CA be a one-dimensional cellular automaton. A computational complexity problem, the Constrained Eden problem, denoted *C-EDEN*(CA), is introduced. *C-EDEN*(CA) is a finitary variant of the Garden of Eden problem. Even for certain elementary cellular automata, *C-EDEN*(CA) is NP-complete, providing the first examples of NP-complete problems associated with cellular automata over a two-element alphabet.

1. Introduction

In "Twenty Problems in the Theory of Cellular Automata" [1], Stephen Wolfram asks in Problem 19: "How common are computationally intractable problems about cellular automata?"

The goal here is to respond to Problem 19 by providing a short paper, accessible to anyone familiar with the basics of computational complexity at the level of the Garey and Johnson classic [2] that introduced Constrained Eden as a computational complexity problem associated with one-dimensional cellular automata. Constrained Eden is a finitary variant of the classic Garden of Eden problem and it provides the first example of NP-complete behavior for an elementary cellular automaton. In a sequel to this paper, Constrained Eden problems and variants will be shown to be log-space equivalent to certain constraint satisfaction problems.

Unlike the Garden of Eden problem (and existing finitary variants of it), variables are involved in instances of Constrained Eden. Constrained Eden is a problem that involves an aspect of equation-solving, of algebra, over a cellular automaton, which may help explain how there can exist class 2 elementary cellular automata (automata whose behavior stabilizes after a small number of iterations) with a Constrained Eden problem that is NP-complete, as is shown later in Section 2.1.

1.1 One-Dimensional Cellular Automata

Let $A = \{a_1, \ldots, a_k\}$ be a k-element alphabet. A *configuration* C is a bi-infinite string, that is, an element in ${}^{\omega}(A \bigcup \{b\})^{\omega}$ where b denotes

https://doi.org/10.25088/ComplexSystems.18.3.379

"blank". The use of a blank symbol as an element of the alphabet is nonstandard in cellular automata theory, but is essential in the definition of the Constrained Eden problem. The *A-support* of a configuration is the set of cells with entries in A. A *finite configuration* over A is a configuration with finite A-support. A finite configuration is generally specified as a finite sequence in $A \bigcup \{b\}$.

A one-dimensional cellular automaton CA of *radius r* consists of a finite alphabet $A = \{a_1, \ldots, a_k\}$ and *local rule* $\rho : A^{2r+1} \to A$. A given configuration evolves in time by synchronously updating the value of each cell via the local rule ρ. It is understood that if a blank b is encountered as part of an input involved in determining a cell, then the output (the new value of the cell) is also a b. So, in effect, the domain and range of ρ is extended to include b, and b acts as an absorbing element. Using the given convention, a one-dimensional cellular automaton CA induces, via its local rule ρ, a function from the set of configurations to itself, a function that will also be denoted ρ.

Consider the so-called elementary cellular automaton 36 over the alphabet $A = \{0, 1\}$, with local rule $\rho : \{0, 1\}^3 \to \{0, 1\}$ given by $\rho^{-1}(1) = \{(1, 0, 1), (0, 1, 0)\}$. Here is the evolution of the configuration 01 011 010 after three iterations of elementary cellular automaton 36:

$$
\begin{array}{ccccccc}
0 & 1 & 0 & 1 & 0 & 0 & 1 \\
 & 1 & 1 & 1 & 0 & 0 & \\
 & & 0 & 0 & 0 & & \\
 & & & 0 & & &
\end{array}
$$

For a one-dimensional cellular automaton over $A = \{a_1, \ldots, a_k\}$ with radius r and local rule ρ, let $\rho^{-1}(a_i) = \{(a_{2r}, \ldots, a_0) : \rho(a_{2r}, \ldots, a_0) = a_i\}$. There are 256 elementary cellular automata, cellular automata with radius 1 over the alphabet $\{0, 1\}$, each designated by a positive integer n, where $0 \le n \le 255$, as follows: $n = \Sigma_{(a_2, a_1, a_0) \in \rho^{-1}(1)} 2^{a_2 (2^2) + a_1 (2^1) + a_0}$.

The one-dimensional cellular automata here are assumed to have a *quiescent* state $0 \in A$: $\rho\left(0^{2r+1}\right) = 0$. The *support* of a configuration is the set of cells that are not in the quiescent state 0 or blank. A configuration is *0-finite* if its support is finite.

An alphabet A, a local rule ρ, and a fixed positive integer k determine a *finite cellular automaton*: For such a cellular automaton, the action of ρ is restricted to A^k (i.e., blanks are not used). To evaluate

https://doi.org/10.25088/ComplexSystems.18.3.379

the left and right boundary of an element of A^k the k-tuple is typically padded with entries consisting of some fixed quiescent element of A.

1.1.1 Background: The Garden of Eden Problems and Complexity Problems Associated with Cellular Automata

Let CA be a cellular automaton over an alphabet A. The Garden of Eden for CA has as an instance a bi-infinite configuration over A with no blanks, the *target* T, and asks: "Does there exist a configuration S such that $\rho(S) = T$?". In [3] a finitary version of the Garden of Eden problem, called the Predecessor Existence Problem (PEP) is introduced. As defined by K. Sutner in [3], the scope of PEP is **finite** cellular automata, but in the paragraph before Proposition 3.1, Sutner broadens the scope of PEP to include 0-finite cellular automata. In [3], (see the paragraph before the introduction to Proposition 3.1, via the semiautomaton determined by the so-called deBruijn labeled graph determined by the local rule of the one-dimensional cellular automaton CA. Also in [3] are NP-completeness results requiring quite nontrivial constructions for the so-called "pattern reachability" problem for finite cellular automata.

The first known examples of NP-complete problems associated with a one-dimensional cellular automaton were given by Frederic Green in [4]. Let CA be a one-dimensional cellular automaton over the alphabet A. Consider the problem CA-PREIMAGE, with instances T that are finite configurations over A, with size $|T|$ the length of T, and with the question: Does there exist a finite subconfiguration that maps to T in $|T|$ steps? Green shows that one-dimensional cellular automata do exist that have an NP-complete CA-PREIMAGE problem. On the other hand, for various trivial one-dimensional automata, it is clear that CA-PREIMAGE is in P. Thus, CA-PREIMAGE properly separates the one-dimensional cellular automata (assuming P $\neq$ NP). Green's nontrivial examples of NP-complete behavior for CA-PREIMAGE involve cellular automata over a fairly large alphabet.

2. Constrained Eden

Let CA be a one-dimensional cellular automaton of radius r over a finite alphabet $A = \{a_1, \ldots, a_k\}$ with the local transition function ρ. It is assumed that A contains a quiescent state 0. Let $V = \{x_i : i \in \mathbf{N}\}$ be a countable variable set.

1. An instance of the Constrained Eden problem for a cellular automaton CA (denoted *C-EDEN*(CA)) consists of

    - a *list*, a finite sequence of entries $Y_1, \ldots, Y_m$ with entries from $V \bigcup A$, and

https://doi.org/10.25088/ComplexSystems.18.3.379

- a *target*, a finite sequence $Z_{r+1}, \ldots, Z_{m-r}$ with entries consisting of elements from $A \bigcup \{b\}$. The list is placed over the target, $Y_r + 1$ over Z_{r+1}, and so on.

2. The question is: Does there exist an assignment $e : \{x_1, \ldots, x_m\} \to A$, so that $e(Y_1), e(Y_2), \ldots, e(Y_m)$ is mapped by the local rule ρ to T, with the understanding that cells of the target containing blanks are ignored?

Of course the variables in an instance of Constrained Eden place constraints on potential solutions. An instance of *C-EDEN*(CA) follows, where CA could be any one-dimensional cellular automaton with radius $r = 1$ over any alphabet A containing $\{0, 1\}$:

$$\begin{array}{cccccc} x_2 & 0 & x_1 & x_2 & x_1 & 0 \\ & 1 & b & 0 & 1 & \end{array}$$

It is often convenient to assume that the length of the target is the same as the length of the list. This can be accomplished by adding a sufficient number of bs at each end of the target. Thus, an instance of Constrained Eden can be viewed as a finite sequence of 2-tuples, with entries from $(V \bigcup A) \times (A \bigcup \{b\})$. It is clear that *C-EDEN*(CA) is in NP.

Lemma 1 is not intrinsic to the paper; the proof is a slight expansion of an observation in Sutner's paper involving the labeled directed de Bruijn graph associated with a one-dimensional automaton, and the interested reader is referred to [3].

Lemma 1. Let CA be a one-dimensional cellular automaton with $r \geq 1$ over an alphabet A with quiescent state 0. Then the 0-finite version of PEP(CA) is log-space equivalent to a subproblem of *C-EDEN*(CA).

Proof. Let $I = 0^{\omega} c_1 \ldots c_m 0^{\omega}$ be an instance (of size m) of PEP(CA). In [3], Sutner indicates that I is satisfiable if and only if $c_1 \ldots c_m$ is recognized by the A-labeled de Bruijn directed graph (interpreted as a semiautomaton) determined by the one-dimensional CA, with

- an initial state $x \in A^{2r}$ such that there exists a directed path in the de Bruijn graph from 0^{2r} to x, a path labeled entirely by 0s, and
- a final state $y \in A^{2r}$ such that there is a directed path from y to 0^{2r} such that the entire path is labeled by 0s.

Since the diameter of the de Bruijn graph in question has a length of no more than $2r$, it follows that there exists a satisfying 0-finite pre-image for the instance $I = c_1 \ldots c_m$ if and only if there exists a satisfying 0-finite pre-image whose support is contained in a subconfiguration of a length no more than $m + 2r$.

For an instance $I = c_1 \ldots c_m$ of PEP(CA), with size m, let $\alpha(I)$ be a list of $\alpha(I) = 0^{2r} x_1 \ldots x_{m+2r} 0^2 r$, where $x_1, \ldots, x_{m+2r}$ are distinct variables, and with target $0^{2r} c_1 \ldots c_m 0^{2r}$, situated so that x_{r+1} is

https://doi.org/10.25088/ComplexSystems.18.3.379

over c_1 in the target. From the preceding paragraph, it follows that I is satisfiable for the 0-finite version of PEP(CA) if and only if $\alpha(I)$ is satisfiable for *C-EDEN*(CA). Meaning that the map is log-space is clear. □

Let *C-EDEN*(n) refer to the Constrained Eden problem for elementary cellular automaton n, and let ρ_n refer to the local rule for elementary cellular automaton n. Note that if $\rho\left(x_1, x_2, x_3\right) = x_2$, the local rule for elementary cellular automaton 206, then *C-EDEN*(206) is certainly in P. The introduction of variables and blanks results in a natural problem that, assuming P ≠ NP, divides one-dimensional cellular automata into at least two complexity classes, as shown next.

2.1 *C-EDEN*(36) is NP-Complete

Recall that NOT-ALL-EQUAL 3-SAT ("NAE 3-SAT") is the computational complexity problem with instances consisting of a finite set of clauses of length 3, and question: Does there exist an assignment of the variables involved in the instance such that each clause is satisfied, but at least one of the literals in each clause is not satisfied? NOT-ALL-EQUAL 3-SAT, which is well known to be NP-complete, is reduced to *C-EDEN*(36).

Theorem 1. *C-EDEN*(36) is NP-complete.

Proof. Let $I = \left\{C_1, \ldots, C_k\right\}$ be an instance of NOT-ALL-EQUAL 3-SAT, and let var_I be the variables occurring in I. By reindexing if necessary, it can be assumed that $\text{var}_I = \left\{x_j : j = 1, \ldots, m\right\}$ for some non-negative integer m. (A particular instance, E, of NOT-ALL-EQUAL 3-SAT is presented to make it easier to follow the constructions: Let $E = \left\{\left(x_1, x_2, x_3\right), \left(\neg x_1, \neg x_2, \neg x_3\right), \left(x_2, x_3, \neg x_4\right), \left(\neg x_1, x_2, x_4\right)\right\}$. Note that E is satisfiable; for example, it is satisfied with $f : \left\{x_1, x_2, x_3, x_4\right\} \to \{0, 1\}$ with $f^{-1}(1) = \{x_1, x_2, x_4\}$.)

The first step involves the construction of a set of Boolean equations, denoted $\overline{I}$. Then, in the second and last step, the set of equations $\overline{I}$ is transformed into an instance, *newI*, of *C-EDEN*(36).

Consider the Boolean equation $\rho_{36}(x, y, x) = 1$. An evaluation $e : \{x, y\} \to \{0, 1\}$ satisfies the equation if and only if $e(x) \neq e(y)$. That is, the equation $\rho_{36}(x, y, x) = 1$ defines $\neg x$. The variable x_{k+m} will represent $\neg x_k$. The equations $\left\{\rho_{36}\left(x_i, x_{i+m}, x_i\right) = 1 : i = 1, \ldots, m\right\}$ are put into $\overline{I}$. Note that these equations depend only on the variables mentioned in the instance of I. (The equations that arise in this manner from the instance E are $\left\{\rho\left(x_i, x_{i+4}, x_i\right) = 1 : i = 1, 2, 3, 4\right\}$, a set of Boolean equations that will be expanded to $\overline{E}$ in the following.)

A second group of equations is added to $\overline{I}$, completing its construction. Consider the Boolean equation $\rho_{36}(X, \neg Y, Z) = 0$, where

https://doi.org/10.25088/ComplexSystems.18.3.379

X, Y, Z are literals. An evaluation e of the variables involved in X, Y, Z satisfies $\rho_{36}(X, \neg Y, Z) = 0$ if and only if $\{0, 1\} = \{e(X), e(Y), e(Z)\}$. That is, e satisfies $\rho_{36}(X, \neg Y, Z) = 0$ if and only if e (extended to literals in the usual way) maps at least one of the literals X, Y, Z to 0, and at least one of the literals X, Y, Z to 1. For each clause $C_j = \{X_j, Y_j, Z_j\}$ of I, add $\rho_{36}(X_j, \neg Y_j, Z_j) = 0$ to $\overline{I}$. This completes the construction of $\overline{I}$, a set with $m + k$ equations. For the instance E, the equations $\overline{E}$ are:

$$\begin{aligned}&\{\rho(x_i, x_{i+4}, x_i) = 1 : i = 1, 2, 3, 4\} \cup\\ &\quad \{\rho(x_1, x_5, x_3) = 0, \rho(x_5, x_2, x_7) = 0,\\ &\qquad \rho(x_2, x_7, x_8) = 0, \rho(x_5, x_6, x_4) = 0\}.\end{aligned}$$

The instance I of NOT-ALL-EQUAL 3-SAT is satisfiable, under an assignment e of the variables involved in I, if and only if the $m + k$ equations $\overline{I}$ are simultaneously satisfiable under $\overline{e}$, where $\overline{e}$ extends e as follows:

$$\overline{e}(x_{i+m}) = 1 + e(x_i) \bmod 2, \text{ where } i = 1, \ldots, m.$$

Next, $\overline{I}$ is mapped to an instance *newI* of *C-EDEN*(36). For an equation $\rho(x, y, z) = c \in \overline{I}$, denote (x, y, z) its "variable-tuple" and let $c \in \{0, 1\}$ be its "target". Suppose that $\overline{I}$ consists of j equations, given in some order. Concatenate the j variable-tuples to form a sequence of variables and constants of length $3j$. This is the "list" part of the instance *newI* of *C-EDEN*(36). Consider the sequence of length j consisting of the j targets of the equations $\overline{I}$. Form another sequence, the target T, consisting of constants and blanks, with $T = (b, a_1, b, b, a_2, b, b, a_3, \ldots, b, b, a_j, b)$. Complete the construction of the instance *newI* by placing the list over the target.

It is not difficult to see that I is satisfiable if and only if the set of equations $\overline{I}$ is satisfiable if and only if *newI* is satisfiable. The mapping from I to *newI* is clearly polynomial-time; thus, NOT-ALL-EQUAL 3-SAT reduces to *C-EDEN*(36).

In the case of instance E, *newE* is given below. To fit *newE* on one line, variables (in the first row) "x_i" are shortened to "i":

$$\begin{array}{cccccccccccccccccccccccc} 1 & 5 & 1 & 2 & 6 & 2 & 3 & 7 & 3 & 4 & 8 & 4 & 1 & 5 & 3 & 5 & 2 & 7 & 2 & 7 & 4 & 5 & 6 & 4 \\ & 1 & b & b & 1 & b & b & 1 & b & b & 1 & b & b & 0 & b & b & 0 & b & b & 0 & b & b & 0 & \end{array}.$$

Observe that *newE* is satisfiable with assignment $\overline{f} : \{x_1, \ldots, x_8\} \to \{0, 1\}$, where $\overline{f}$ extends the map f given at the end of the first paragraph of this proof, via $f(x_{4+i}) = 1 + x_i \pmod 2$, for $i = 1, 2, 3, 4$. □

https://doi.org/10.25088/ComplexSystems.18.3.379

3. Conclusion

Assuming that P ≠ NP, it is shown here that Constrained Eden separates the elementary cellular automata into at least two complexity classes. It is apparent from the proof of Theorem 1 that Constrained Eden(CA) can be viewed as an equation-solving problem over the cellular automaton CA. That is, an instance of Constrained Eden asks whether a certain finite equation, in an algebraic system with the single operation ρ, has a solution. The complexity of equation-solving (i.e., algebra) over cellular automata is pursued further in a sequel by the author.

Also in the sequel, Constrained Eden problems will be shown to be log-space equivalent to certain constraint satisfaction problems, thereby allowing recently developed universal algebra methods for classifying constraint satisfaction problems to be applied to Constrained Eden problems. For example, these methods allow the Constrained Eden problem for elementary cellular automata to be separated into exactly two classes: those with a Constrained Eden problem in P, and those with an NP-complete Constrained Eden problem. Further refinements of the classification of Constrained Eden for one-dimensional cellular automata involving subclasses of NC are also discussed in the sequel.

Finally, it is not difficult to see that a higher-dimensional cellular automaton has a natural enough Constrained Eden problem, one that is polynomial-time equivalent to the Constrained Eden problem of an appropriately defined one-dimensional cellular automaton over the same alphabet.

Acknowledgments

The author thanks Matthew Szudzik for interesting and useful discussions concerning the topics and results of the paper at the 2008 NKS Summer School.

References

[1] S. Wolfram, "Twenty Problems in the Theory of Cellular Automata," *Physica Scripta*, **T9**, 1985 pp. 170-183.

[2] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, San Francisco: W. H. Freeman, 1979.

[3] K. Sutner, "On the Computational Complexity of Finite Cellular Automata," *Journal of Computer and System Sciences*, **50**(1), 1995 pp. 87-97.

[4] F. Green, "NP-Complete Problems in Cellular Automata," *Complex Systems*, **1**(3), 1987 pp. 453-474.

https://doi.org/10.25088/ComplexSystems.18.3.379