On the Equivalence Classes of One-Dimensional Cellular Automata

November 23, 2024

Abstract

Using a group-theoretical approach, a method is presented to determine the equivalence classes of one-dimensional cellular automata for an arbitrary but finite order of the state set and an arbitrary but finite order of the neighborhood. Results for state sets of order two and three are included.

1 Introduction

A one-dimensional cellular automaton (CA) operates on a bi-infinite lattice of cells where each cell is in one state from a finite set of possible states. A computational step of the automaton comprises the following operations. For each cell the automaton reads the states of a small set of neighboring cells including the cell itself. The values of the states read are used as input of a lookup table, called the local rule, that determines the new state of the cell. Then all cells are updated synchronously. The net effect of one computational step is the calculation of a new bi-infinite sequence of states.

Multiple iterative computational steps of the CA leads to a sequence of configurations, termed the evolution of the CA. If each configuration of a CA's evolution is shifted the same number of cells to the left or to the right, the CA's evolution is still governed by the same local rule. This fundamental property of CA is called shift invariance.

Other symmetry operations transform the local rule. If the CA's evolution is reflected (or mirrored), the resulting evolution is governed by the reflected rule, which, in general, is different, to the unmirrored one. Reflection is thus a symmetry operation that transforms rules.

Similarly, since the states of a CA are merely labels, permutating the labels does not change the dynamic behaviour of the CA, but will in general result in a different local rule. Rules that can be transformed into each other under reflection or permutation or their product are considered equivalent. Consequently, the space of all CA rules splits up into classes of equivalent rules.

Wolfram [10] designated the family of one-dimensional CA with two states and three neighbors as elementary. In [9], pp. 485–557, he gave a table that divided the 256 rules of the elementary CA into 88 equivalence classes. A mathematical derivation of this result was carried out by Li and Packard [5]. The result for the next larger family of one-dimensional CA with two states and four neighbors can also be found in literature: the 65 536 rules break up into

16 704 equivalence classes, see, e.g., [7]. Cattaneo et al. [1] studied a variety of transformations of the rule space, in particular, also the symmetry operation to be discussed in this work. They gave, inter alias, the general result for the equivalence classes of two-state CA with $2r + 1$ neighbors, where r is a nonnegative integer.

Determining the equivalence classes is an elementary classification and serves both to understand the rule space of CA in terms of symmetry operations and to reduce the number of non-equivalent rules. There are a variety of other classification schemes. For instance, Wolfram's classification [11] is based on the phenomenological behavior of the dynamic evolution, the Culik-Yu classification [2] captures the computational complexity of the limit sets, see [8] for an overview. The classification by symmetry operations precedes these higher-level classifications as rules in an equivalence class are all in the same class of other classification schemes (at least they should be).

Section 2 provides the formal definitions of CA and symmetry operations. Section 3 presents two different types of group actions: one that acts on the set of local rules and another that acts on the set of words forming the domain of a local rule. Additionally, this section describes the method to determine the number of equivalence classes. The actual calculation of the number of equivalence classes can be found in Section 4 for a state set of order 2 and in Section 5 for a state set of order 3. Following this, Section 6 presents a brute-force algorithm for computing the number of equivalent classes. Due to the runtime complexity an actual implementation will terminate only if the number of states and the number of neighbors are sufficiently small. This section serves also to validate the results obtained in Section 4 and 5 for a small size of neighbors.

2 Definitions

2.1 One-dimensional Cellular Automata

The states of a CA are represented by symbols from a finite set, also called an alphabet. As the symbols only serve to designate the states, any finite set will do, so we choose the set $\Sigma = \{0, 1, \ldots, k-1\}$ to represent a state set of order k. The order of (an arbitrary) set A is denoted by $|A|$.

A word $w = x_0 x_1 \ldots x_{m-1}$ over an alphabet Σ is a finite sequence of symbols from Σ juxtaposed. The length of a word w, denoted $|w|$, is the length of the sequence, that is $|x_0 x_1 \ldots x_{m-1}| = m$ (the notation $|.|$ denotes both the order of a set and the lenght of a word). The set of all words of length m over the alphabet Σ is denoted by Σ^m. A configuration x of an one-dimensional CA is a bi-infinite sequence over the alphabet Σ, defined as a mapping of $\mathbb{Z}$ into Σ. The i-th element, $i \in \mathbb{Z}$, of a configuration x is denoted by x_i.

Formally, a one-dimensional CA is denoted by a triple (k, n, f), where
k is a positive integer, the order of $\Sigma = \{0, 1, \ldots, k-1\}$.
n is a positive integer, the size of the neighborhood.
f is the local rule, a function from Σ^n to Σ.

The local mapping f induces the global mapping on the set of configurations $\Phi_f : \Sigma^{\mathbb{Z}} \to \Sigma^{\mathbb{Z}}$, defined by $\Phi_f(x)_i = f(x_i x_{i+1} \ldots x_{i+n-1})$.

If the CA is initialised with the configuration x, the CA computes in one step the configuration $\Phi_f(x)$.

For any fixed k and any fixed n, $L(k,n)$ denotes the set of local rules of the family of CA with k states and n neighbors (sometimes just L if k and n are given). The order of $L(k,n)$ is k^{k^n}. If $k > 1$, $L(k,n)$ grows extremly fast as function of n: $L(k,0) = k$, and $L(k,n+1) = L(k,n)^k$.

The shift operator σ operates on the set of configurations, it shifts a configuration one cell to the left, formally defined by $\sigma(x)_i = x_{i+1}$. By the definition of the CA, the global mapping commutes with the shift operator: $\Phi_f(\sigma(c)) = \sigma(\Phi_f(c))$. A fundamental result of Hedlund [4] shows that an alternative, topological definition of one-dimensional CA based on the shift operator and continous mappings is equivalent to the one above.

2.2 Symmetry operations

The notion of the equivalence of one-dimensional CAs is based on two classes of symmetry operations: permutations of the state set and reflection of the configuration.

Let S_k be the symmetric group of degree k, that is the set of all permutations of the set $\Sigma = \{0, 1, \ldots, k-1\}$, and suppose $\alpha \in S_k$. If $a \in \Sigma$ we write the image of a under α as product αa. The extension of α to words and configurations is defined by elementwise application. If $w = a_0 \ldots a_{n-1} \in \Sigma^n$ is a word, set $\alpha w = (\alpha a_0) \ldots (\alpha a_{n-1})$. If x is a configuration, set $(\alpha x)_i = \alpha(x_i)$. Suppose f is a local rule that maps Σ^n to Σ. The permutation operator $\hat{\alpha}$ is defined by $\hat{\alpha} f(w) = \alpha f(\alpha^{-1} w)$ for all words $w \in \Sigma^n$. It represents a transformation of the rule space. Note that the "hat" on the operator is necessary, because αf and $\hat{\alpha} f$ are distinct entities. The first one is the composite function $\alpha \circ f$, whereas the second represents the composite function $\alpha \circ f \circ \alpha^{-1}$. If Φ_f is the induced global mapping of f, we define $\hat{\alpha}\Phi_f$ similarly: $\hat{\alpha}\Phi(x) = \alpha\Phi_f(\alpha^{-1}x)$ for all configurations x. From

$$\begin{aligned} \left(\hat{\alpha}\Phi_f(x)\right)_i &= \left(\alpha\Phi_f(\alpha^{-1}x)\right)_i = \alpha f\left(\alpha^{-1}(x_i \cdots x_{i+n-1})\right) \\ &= \hat{\alpha} f(x_i \cdots x_{i+n-1}) = \left(\Phi_{\hat{\alpha} f}(x)\right)_i \end{aligned}$$

follows $\hat{\alpha}\Phi_f = \Phi_{\hat{\alpha} f}$.

They second type of operator is the reflection operator. If $w = a_0 \ldots a_{n-1} \in \Sigma^n$ is a word over Σ, define $rw = a_{n-1} \ldots a_0$. Note that $ra = a$ for all $a \in \Sigma$. If x is a configuration, define $(rx)_i = x_{-i}$ for all configurations x. The reflection operator $\hat{r}$ is defined by $\hat{r} f(w) = f(rw)$ and $\hat{r}\Phi_f(x) = r\sigma^{1-n}\Phi_f(rx)$. Since r is self-inverse, that is $r^{-1} = r$, we can also write $\hat{r} f(w) = r f(r^{-1}w)$, making the notation consistent with the one of the permutation operator. From

$$\begin{aligned} \left(\hat{r}\Phi_f(x)\right)_i &= \left(r\sigma^{1-n}\Phi_f(rx)\right)_i = \left(\sigma^{1-n}\Phi_f(rx)\right)_{-i} \\ &= \left(\Phi_f(rx)\right)_{-i-n+1} = f\left((rx)_{-i-n+1} \cdots (rx)_{-i}\right) \\ &= f\left(x_{i+n-1} \cdots x_i\right) = f\left(r(x_i \cdots x_{i+n-1})\right) \\ &= \left(\hat{r} f(x_i \cdots x_{i+n-1})\right)_i = \left(\Phi_{\hat{r} f}(x)\right)_i ; \end{aligned}$$

we conclude that the relation $\hat{r}\Phi_f = \Phi_{\hat{r} f}$ holds. Note that this relation is only fulfilled, because we added the term σ^{1-n} involving the shift operator to the definition of $\hat{r}\Phi_f$.

The significance of the operators defined above, can be seen by the following observation. Suppose $\hat{\alpha}$ is either one of the permutation operators or the

reflection operator, and consider two CA with the same state set and the same neighborhood size and respectively, with local rule f and local rule $\hat{\alpha}f$. If the initial configuration of CA 1 is x and the one of CA 2 is αx, then the same 1-1 correspondence between the configurations established by α persists for all iterations: $\alpha\Phi_f^t(x) = \Phi_{\hat{\alpha}f}^t(\alpha x)$ holds for any positive integer t (Φ^t denotes the t-th iteration of Φ).

We call $R = \{1, r\}$, the reflection group. The direct product of S_k and R, written as S_kR, is the group that contains all permutations, the reflection and their products. Suppose that $\hat{\alpha}$ and $\hat{\beta}$ are two operators. Then

$$\hat{\alpha}\hat{\beta}f(w) = \alpha\hat{\beta}f(\alpha^{-1}w) = \alpha\beta f(\beta^{-1}\alpha^{-1}w) = \widehat{\alpha\beta}f(w).$$

This shows that the operators form a group that is isomorphic to S_kR. Note that the reflection operator commutes with all permutation operators

If the global mapping Φ_f satisfies $\Phi_f = \hat{\alpha}\Phi_f$, the CA is said to be invariant under the operator $\hat{\alpha}$.

In the literature, e.g. [11], an alternate definition of the global function is frequently employed, where the updated cell is in the center of the neighborhood block. If r is a positive integer, the the global mapping has the form $\Phi_f(x)_i = f(x_{i-r} \dots x_i \dots x_{i+r})$, where r is called the radius of the CA. Defining CA this way has the benefit that it eliminates the need for a shift operator in the definition of the reflection operator. However, it becomes cumbersome, when dealing with neighborhoods of even order. The previous definition updates the first cell to the left of the neighborhood block, which accommodates both even and odd neighborhood sizes.

3 Preliminaries

We assume some basic knowledge of groups as it can be found in introductory textbooks, e.g. [3], [6]. However, we briefly define group actions and related concepts and state some propositions about them, all of these to be found in more depth and more relaxed pace in the references above.

A group action of a group G on a set A is a map from $G \times A$ to A satisfying the following properties:

(i) $g_1(g_2a) = (g_1g_2)a$ for all $g_1, g_2 \in G$, $a \in A$, and

(ii) $1a = a$, for all $a \in A$.

The relation on A, defined by $a \sim b$ if and only if $a = gb$ for some $g \in G$, is an equivalence relation. The equivalence classes $[a] = \{ga : g \in G\}$ are called G-orbits. The stabilizer of $a \in A$ is the set $\mathrm{stb}(a) = \{g \in G : ga = a\}$, which forms a subgroup of G.

Let H be a subgroup of G. The index $[G : H]$ of G is defined to be the number of left cosets of H in G. Lagrange's theorem states that $|G| = [G : H]\,|H|$. The order of an equivalence class $[a]$ is given by $|[a]| = |G : \mathrm{stb}(a)|$.

3.1 Symmetry operators acting on the rule space

Fix k and n. The mapping $S_kR \times L \to L$; $(\alpha, f) \mapsto \hat{\alpha}f$ is a group action. If $f \in L$, the equivalence class of f, denoted by $[f]$, is defined by the group action. The

stabilizer of f consists of all symmetry operators under which f is invariant. It maps f to a subgroup H of S_kR. The inverse mapping $\mathrm{stb}^{-1}(H)$ creates therefore a partition of L. The direct calculation of $\mathrm{stb}^{-1}(H)$ is cumbersome, it is easier to take a detour. If H is a subgroup of S_kR, define

$$\mathrm{ivt}(H) = \{f \in L : \hat{\alpha}f = f \text{ for all } \alpha \in H\},$$

that maps H to the set of local rules that are invariant under all operators in H (short, invariant under H). This mapping does not create a partition of L: if H_1 is a proper subgroup of H_2 (H_1 is a subgroup of H_2 and $H_1 \neq H_2$), denoted by $H_1 < H_2$, then $\mathrm{ivt}(H_2) \subset \mathrm{ivt}(H_1)$. The mappings ivt and stb are related:

$$\mathrm{stb}^{-1}(H) = \mathrm{ivt}(H) \setminus \bigcup_{H<H'} \mathrm{stb}^{-1}(H'), \tag{1}$$

where H' is also assumed to be a subgroup of S_kR. Since the sets $\mathrm{stb}^{-1}(H)$ are disjunct, Equation 1 implies:

$$|\,\mathrm{stb}^{-1}(H)| = |\,\mathrm{ivt}(H)| - \sum_{H<H'} |\,\mathrm{stb}^{-1}(H')|. \tag{2}$$

To calculate $|\,\mathrm{stb}^{-1}(H)|$ for all subgroups, one has to start with S_kR, for which $\mathrm{stb}^{-1}(S_kR) = \mathrm{ivt}(S_kR)$ holds, the calculation of the subgroups can then be done successively. The sets $\mathrm{stb}^{-1}(H)$ contain all invariant rules. The following lemma is helpful in understanding how they split up into equivalence classes.

Lemma 1 *Let f be a local rule that is invariant under $\hat{\alpha}$. Then $\hat{\beta}f$ is invariant under $\widehat{\beta\alpha\beta^{-1}}$.*

We make use of $(\beta\alpha\beta^{-1})^{-1} = \beta\alpha^{-1}\beta^{-1}$. A function g is invariant under $\widehat{\beta\alpha\beta^{-1}}$ if and only if $g(\beta\alpha^{-1}\beta^{-1}v) = \beta\alpha^{-1}\beta^{-1}g(v)$ for all words v in the domain. We put $g = \hat{\beta}f$. By definition of the operator $\hat{\beta}$, $g(w) = \beta f(\beta^{-1}w)$ for all words w in the domain. Hence

$$g(\beta\alpha^{-1}\beta^{-1}v) = \beta f(\beta^{-1}\beta\alpha^{-1}\beta^{-1}v) = \beta f(\alpha^{-1}\beta^{-1}v)$$

From the invariance of f under $\hat{\alpha}$ follows $f(\alpha^{-1}w) = \alpha^{-1}f(w)$. Thus

$$\beta f(\alpha^{-1}\beta^{-1}v) = \beta\alpha^{-1}f(\beta^{-1}v) = \beta\alpha^{-1}\beta^{-1}\beta f(\beta^{-1}v) = \beta\alpha^{-1}\beta^{-1}g(v).$$

This completes the proof.

Let H be a subgroup of S_kR and f be invariant under H. If H is a normal subgroup then Lemma 1 shows that $[f]$ is a subset of $\mathrm{stb}^{-1}(H)$. If H is not normal, $[f]$ is distributed over several sets, say $[f] \subset \mathrm{stb}^{-1}(H_1) \cup \cdots \cup \mathrm{stb}^{-1}(H_q)$.

3.2 Symmetry operators acting on the rule space domain

The domain of a local rule is the set N of all words over Σ having length n: $N = \Sigma^n$. If H is a subgroup of S_kR the mapping $H \times N \to N$ defined by $(\alpha, w) \mapsto \alpha w$ satisfies the properties of a group action.

Let A and B be two H-orbits. If there is word $v \in A$ and a word $w \in B$, such that $\mathrm{stb}(v) = \mathrm{stb}(w)$, A and B are said to have the same type. If this is

the case, the mapping $\varphi : A \to B$ defined by $\varphi(v) = w$, and $\varphi(\alpha v) = \alpha\varphi(v)$ for all $\alpha \in H$ is an isomorphism.

We will now study mappings that are defined on an orbit. Suppose $A \subset N$ is an H-orbit, and g is a mapping $A \to \Sigma$. If $\alpha \in H$ then $\alpha A = \{\alpha w | w \in A\} = A$. This shows that the mapping $\hat{\alpha}g$ also has A as its domain, and that it makes sense to speak about invariant functions defined on A.

The set $\{A_1, \ldots, A_p\}$ of all H-orbits is a partition of N. If f is a local rule invariant under H, then the restriction $f|A_i$ is clearly also invariant under H. On the other hand, if $g_i : A_i \to \Sigma$; $i = 1 \ldots, p$, is a sequence of mappings, all invariant under H, then the local rule defined by $f(w) = g_i(w)$ if $w \in A_i$ is also invariant under H. This shows that invariant functions defined on the orbits are the building blocks of invariant functions defined on N.

Let A be again an H-orbit A and suppose $g : A \to \Sigma$ is invariant under H. Choose a word w in A, and consider a different word in A, say v. Since there is an $\alpha \in H$ such that $v = \alpha w$, the relation $g(v) = \hat{\alpha}g(v) = \alpha g(\alpha^{-1}v) = \alpha g(w)$ holds, and the value of $g(v)$ is determined by $g(w)$. This implies that there are at most k different mappings $g : A \to \Sigma$ that are invariant under H.

3.3 Examples

We will study the group action of $\langle(01)r\rangle$ on some of the orbits of 2-state and 3-state neighborhoods.

1. Let $\Sigma = \{0, 1\}$, and $n = 2m$ be a positive even integer. Suppose that the group $\langle(01)r\rangle$ acts on Σ^n. Consider the word $w = 0^m1^m$ (m copies of 0 followed by m copies of 1). Since $w = (01)rw$, the set $A = \{w\}$ represents a singleton orbit. Assume there is a function f from A to Σ that is invariant under $\langle(01)r\rangle$. Then f has to satisfy the relation $f((01)rw) = (01)rf(w)$. But since $f((01)rw) = f(w)$, we obtain the contraction $f(w) = (01)f(w)$. This shows that there is no local rule on Σ^n that is invariant under $\langle(01)r\rangle$.

2. Let Σ be as above, and let $n = 2m + 1$ be a positive odd integer. Then for each word w in Σ^n, $(01)rw$ is different to w, and they form both the orbit $A = \{w, (01)rw\}$. Choose a symbol a from Σ and set $f(w) = a$. If we set $f((01)rw) = (01)a$, the function f is invariant under $\langle(01)r\rangle$. Since a was arbitrary, there are 2 functions with domain A that are invariant under $\langle(01)r\rangle$.

3. Let $\Sigma = \{0, 1, 2\}$, $n = 2m$ be a positive even integer, and $w = 0^m1^m$. The singleton $A = \{w\}$ is an orbit of $\langle(01)r\rangle$. Assume that f is invariant on A. Then $f(w) = (01)f(w)$ must hold, which is satisfiable by the choice $f(w) = 2$. Hence there exists exactly one function from A to Σ that is invariant under $\langle(01)r\rangle$.

4. Let Σ be as above, but let $n = 2m+1$ be a positive odd integer. Consider a word w of Σ^n and write it in the form $w = ucv$, where u and v are words of length m and c is a symbol. The relation $w = (01)rw$ leads to the constraints $v = (01)ru$ and $c = 2$, satisfied by 3^m words. If w is one of these words, a function defined on the orbit $\{w\}$ that is invariant is constrained to the function value $f(w) = 2$. All other orbits of N have order 2 and allow for three different invariant functions. To summarize,

N splits up into 3^m orbits of order 1, which allow one invariant function per orbit, and $(3^{2m+1} - 3^m)/2$ orbits of order 2, which allow 3 invariant functions.

The examples demonstrates that the number of functions defined on an orbit and invariant under $\langle(01)\rangle$, can be 0, 1, or k. Example 4 shows that it is possible to find expressions that specify the number of orbits of a specific type as a function of n.

3.4 Free Orbits

We have seen in Subsection 3.2 that the number of invariant functions on an orbit is at most the order of the state set $k = |\Sigma|$. To single out the orbits that allow k invariant functions, we introduce the concept of free orbits.

Suppose H is a group acting on N, A is an orbit of this group action, and $k = |\Sigma|$. The orbit A is said to be free if there are exactly k functions from A to Σ which are invariant under H.

The procedure for calculating the equivalence classes of a CA which we will present shortly, requires to decide whether a certain orbit is free or not. The following two Lemmas will facilitate this task. The first lemma states that to decide whether an orbit is free or not, it is sufficient to consider only one word of this orbit. The second lemma says that an orbit is free if the order of the orbit equals the order of the group.

Lemma 2 *Let H be a group, A be an H-orbit, f be a function from A to Σ, and w be any word of A. The function f is invariant under the group H if $f(\alpha w) = \alpha f(w)$ is true for all $\alpha \in H$.*

If v is a word of A and β is an element of H, we have to show that $f(\beta v) = \beta f(v)$ holds. Since A is an orbit there is a $\gamma \in H$ such that $v = \gamma w$. Then $f(\beta v) = f(\beta\gamma w) = \beta f(\gamma w) = \beta f(v)$.

Lemma 3 *Let H be a group and A be an H-orbit. If $|A| = |H|$, then A is free.*

Let w be any word in A, and a be any symbol in Σ. Define a function f from A to Σ as follows. Set $f(w) = a$. Let v be another word of A. Since $|A| = |H|$ there is exactly one $\alpha \in H$ such that $v = \alpha w$. This shows that the value $f(v) = \alpha f(w) = \alpha a$ is uniquely defined. Lemma 2 implies that f is invariant under H.

3.5 Outline of the Complete Calculation

Now all pieces have been introduced to lay down the procedure for calculating the number of equivalence classes of 2-state and 3-state CAs.

1. Construct the group $S_k R$ that is the direct product of the permutation group S_k and the reflective group $R = \{0, r\}$. Having done that, construct the sublattice of all subgroups of $S_k R$.

2. For each subgroup H of $S_k R$ enumerate the types of orbits of the group action $H \times N \to N$. For each type, decide whether the orbits of this type are free or not. If they are free, count the number of them. The result

is an arithmetical expression with n, the number of neighbors, as a free variable. Let p denote the total count of the free H-orbits over all types.

3. For each subgroup H of S_kR calculate the number of local rules invariant under H. We distinguish two cases.

   (i) If $k = 2$ and there is an H-orbit that is not free, then $\mathrm{ivt}(H) = 0$, otherwise $\mathrm{ivt}(H) = 2^p$.

   (ii) If $k = 3$, then $\mathrm{ivt}(H) = 3^p$.

4. Beginning with S_kR calculate $\mathrm{stb}^{-1}(H)$ for each subgroup by using Equation 2.

5. Calculate the number of equivalence classes of local rules. If $f \in \mathrm{stb}^{-1}(H)$ then the order of $[f]$ is $|S_kR|/|H|$.

   (i) If H is a normal subgroup, then $[f] \subset \mathrm{stb}^{-1}(H)$. The number of equivalence classes with respect to H is $|\mathrm{stb}^{-1}(H)| \times |H|/|S_kR|$.

   (ii) If H is not normal, $[f]$ is distributed over several sets, say $\mathrm{stb}^{-1}(H_1)$, ..., $\mathrm{stb}^{-1}(H_q)$. Then $H_1 \cup \cdots \cup H_q$ divides into $q \times |\mathrm{stb}^{-1}(H)| \times |H|/|S_kR|$ equivalence classes, where we assumed that H is one of the $H_1, \ldots, H_q$. This case occurs only if $k = 3$.

   Sum up the numbers to obtain the total number of equivalence classes as function of n.

4 Two States

This section deals with the equivalence classes of one-dimensional CA with a state set of order 2, that is $\Sigma = \{0, 1\}$.

4.1 The Group S_2R

S_2, the symmmetric group of degree 2 contains the identity 1 and the transposition (01). The direct product S_2R is given by

$$S_2R = \langle(01)\rangle\langle r\rangle = \{1, (01)\}\{1, r\} = \{1, (01), r, (01)r\} = \langle(01), r\rangle.$$

The notation $\langle\alpha, \beta, \ldots\rangle$ is called a generator, and denotes the group with the property that every element of the group can be written as finite product of the elements of the generator and their inverses. S_2R is isomorphic to the Klein four-group. The lattice of subgroups is depicted in Fig. 1.

4.2 Odd Number of Neighbors

Suppose the number of neighbors is odd, $n = 2m + 1$, $m = 0, 1, 2, \ldots$. For each subgroup H of S_2R, we will calculate the number of orbits of H acting on Σ^n. For the next paragraphs, N denotes the set Σ^n and w denotes a word of N.

1. The group $\langle 1\rangle$. All orbits are of order 1, and so N partitions into 2^{2m+1} singletons.

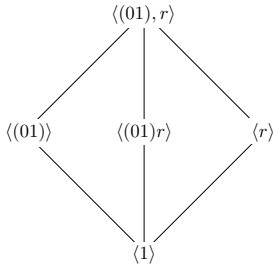

Figure 1: The lattice of S_2R.

2. The group $\langle r\rangle$. The relation $w = rw$ is satisfied if and only if the word w is of the form $ua(ru)$ where $|u| = m$ and a is a symbol. This relation is fulfilled by 2^{m+1} words. The set of these words divides into 2^{m+1} orbits of order 1. The remaining $2^{2m+1} - 2^{m+1}$ words of N are all in orbits of order 2. The total number of orbits is therefore $(2^{2m+1} - 2^{m+1})/2 + 2^{m+1} = 2^m(2^m + 1)$.

3. The group $\langle(01)\rangle$. Consider any word w of N and let a denote the symbol in the center of the word. Then $(01)a$ is the symbol in the center of the word $(01)w$. This shows that the words w and $(01)w$ are always different. Hence N partitions into 2^{2m} orbits of order 2.

4. The group $\langle(01)r\rangle$. As before, the words w and $(01)rw$ are always different. Hence the number of orbits is again 2^{2m}.

5. The group S_2R. If $w = rw$ holds, then also $r(01)w = (01)w$ holds. We have seen that there are 2^{m+1} words satisfying the relation $w = rw$, and so there are 2^m orbits of order 2 consisting of the words w and $(01)w$. The remaining $2^{2m+1} - 2^{m+1}$ words divide into orbits of order 4. Thus the total number of orbits is $(2^{2m+1} - 2^{m+1})/4 + 2^m = 2^{m-1}(2^m + 1)$.

All the orbits of the groups above are free. Put $H_1 = S_2R$, $H_2 = \langle(01)\rangle$, $H_3 = \langle(01)r\rangle$, $H_4 = \langle r\rangle$ and $H_5 = \langle 1\rangle$. Set $a_i = |\operatorname{ivt}(H_i)|$, $b_i = |\operatorname{stb}^{-1}(H_i)|$, and let c_i denote the number of equivalence classes of $\operatorname{stb}^{-1}(H_i)$, for $i = 1, \ldots, 5$. If p_i is the number of (free) orbits of the group H_i, then $a_i = 2^{p_i}$. We get

$$
\begin{aligned}
&b_1 = a_1 = c_1;\\
&b_2 = a_2 - a_1, c_2 = b_2/2;\\
&b_3 = a_3 - a_1, c_3 = b_3/2;\\
&b_4 = a_4 - a_1, c_4 = b_4/2;\\
&b_5 = a_5 - a_1 - b_2 - b_3 - b_4 = a_5 + 2a_1 - a_2 - a_3 - a_4, c_5 = b_5/4.
\end{aligned}
$$

Substituting the c_i, we obtain for the total number of equivalence classes

$$\sum_{i=1}^{5} c_i = \frac{1}{4}\sum_{i=2}^{5} a_i = \frac{1}{4}\sum_{i=2}^{5} 2^{p_i} = \frac{1}{4}\left(2 \times 2^{2^{2m}} + 2^{2^m(2^m+1)} + 2^{2^{2m+1}}\right).$$

Notice that a_1 canceled out in the equation.

Proposition 4 *Let m be a nonnegative integer. The rule space $L(2, 2m+1)$ partitions into*

$$\frac{1}{4}\left(2 \times 2^{2^{2m}} + 2^{2^m(2^m+1)} + 2^{2^{2m+1}}\right)$$

equivalence classes.

This result can be found in [1] as Proposition 21.

4.3 Even Number of Neighbors

Suppose the number of neighbors is even, $n = 2m$, $m = 1, 2, \ldots$. For the next paragraphs, N denotes the set Σ^n and w denotes a word of N. Example 1 has shown that there is no local rule invariant under $\langle (01)r \rangle$, and by implication no local rule invariant under S_2R. Thus, in calculating the number of orbits, we therefore only have to consider the remaining subgroups.

1. The group $\langle 1 \rangle$. N partitions into 2^{2m} orbits of order 1.
2. The group $\langle (01) \rangle$. N partitions into 2^{2m-1} orbits of order 2.
3. The group $\langle r \rangle$. N partitions into $2^{m-1}(2^m + 1)$ orbits.

All the orbits of the three groups above are free. Put $H_1 = \langle r \rangle$, $H_2 = \langle (01) \rangle$, and $H_3 = \langle 1 \rangle$. Set $a_i = |\operatorname{ivt}(H_i)|$, $b_i = |\operatorname{stb}^{-1}(H_i)|$, and let c_i denote the number of equivalence classes of $\operatorname{stb}^{-1}(H_i)$, for $i = 1, \ldots, 3$. If p_i is the number of (free) orbits of the group H_i, then $a_i = 2^{p_i}$. We get

$$\begin{aligned} b_1 &= a_1, c_1 = b_1/2; \\ b_2 &= a_2, c_2 = b_2/2; \\ b_3 &= a_3 - b_1 - b_2, c_3 = b_3/4. \end{aligned}$$

Substituting the c_i, we obtain for the total number of equivalence classes

$$\sum_{i=1}^{3} c_i = \frac{1}{4}\sum_{i=1}^{3} a_i = \frac{1}{4}\sum_{i=1}^{3} 2^{p_i} = \frac{1}{4}\left(2^{2^{m-1}(2^m+1)} + 2^{2^{2m-1}} + 2^{2^{2m}}\right).$$

Proposition 5 *Let m be a positive integer. The rule space $L(2, 2m)$ partitions into*

$$\frac{1}{4}\left(2^{2^{m-1}(2^m+1)} + 2^{2^{2m-1}} + 2^{2^{2m}}\right)$$

equivalence classes.

Table 1 depicts the number of equivalence classes of two-state CA for a neighborhood size $n = 1, \ldots, 5$.

5 Three States

After calculating the equivalence classes of one-dimensional 2-state CA in Section 4 this section deals with one-dimensional CA with a state set of order 3, that is $\Sigma = \{0, 1, 2\}$.

Table 1: Equivalent classes for $k = 2$, $n = 1, \ldots, 5$.

H	1	2	3	4	5
$\langle (01), r\rangle$	2	0	8	0	1 024
$\langle (01)\rangle$	0	2	4	128	32 256
$\langle r\rangle$	1	4	28	512	523 776
$\langle (01)r\rangle$	0	0	4	0	32 256
$\langle 1\rangle$	0	1	44	16 064	1 073 447 424
	3	7	88	16 704	1 074 036 736

5.1 The Group S_3R

The group S_3R, which is the direct product of the symmetric group S_3 and the reflection group R, contains all the symmetry operators of three-state CA.

Some remarks:

1. The depiction of the lattice arranges groups of equal order in the same row. From bottom to top the orders are 1, 2, 3, 4, 6, and 12.
2. Groups are specified by generators, e.g. $S_3 = \langle (01), (12)\rangle$.
3. From $[(012)r]^3 = r$ follows $\langle (012)r\rangle = \langle (012), r\rangle$.
4. The group $\langle (01)r, (012)\rangle$ consists of the elements $1, (01)r, (12)r, (20)r, (012)$, and (021).

5.2 The Orbits of the Subgroups of S_3R acting on Σ^n

Let $N = \Sigma^n$. The following Lemma facilitates the calculation of orbits of subgroups that contain the permutation (012).

Lemma 6 *Suppose H is a subgroup of S_3R that contains the permutation (012), and suppose that A is an orbit of H acting on N. Then the number 3 divides the order of A.*

There is no word in N that is invariant under (012). This is equivalent to saying that for all words w in N, the permutation (012) is not an element of the subgroup $\mathrm{stb}(w)$, and hence that the group $\langle (012)\rangle$ is not a subgroup of $\mathrm{stb}(w)$. Since (012) and (210) are the only elements of order 3 in H, the subgroup $\mathrm{stb}(w)$ is not divisible by 3. From $|\mathrm{stb}(w)|\ |A| = |H|$ follows the proposition.

The next simple Lemma will help us in classifying orbits as non-free.

Lemma 7 *Suppose H is a subgroup of S_3R and A is an orbit of H acting on N. The orbit A is not free*

(i) if $(01) \in H$ and there is a $w \in A$ such that $w = (01)w$; or

(ii) if $(01)r \in H$ and there is a $w \in A$ such that $w = (01)rw$.

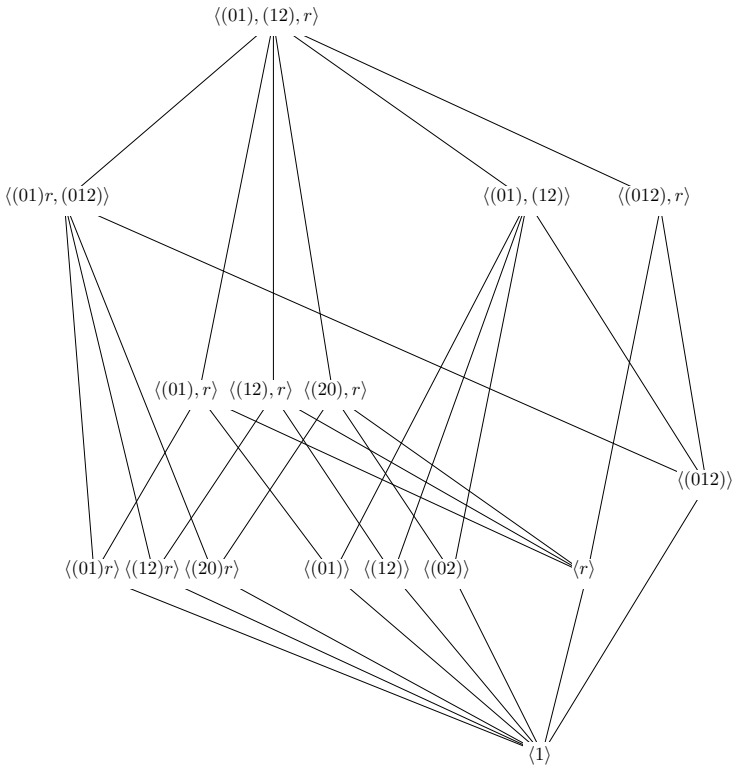

Figure 2: The lattice of S_3R.

Let α be (01) or $(01)r$ and suppose that f is an invariant function from A to Σ. Then the relation $f(w) = \alpha^{-1}f(\alpha w) = \alpha f(w)$ implies $f(w) = 2$. This completes the proof.

All words below are understood to be words over Σ. The word w always denotes a word of N. Sometimes we will write w as a concatenation of two words, that is $w = uv$, if $n = 2m$ is even, and as a concatenation of a word, a symbol, and a further word, that is $w = uav$, if $n = 2m+1$ is odd. If we do so, we assume that $|u| = |v| = m$. For each subgroup H of S_3R the number of free orbits of H acting on N is calculated as follows. In the calculations themselves we will make frequent use of Lemma 3 and Lemma 7, without explicity referencing them.

1. The group $\langle 1\rangle$. N splits up into 3^n free orbits of order 1.

2. The group $\langle r\rangle$. A calculation similar to the state set of order 2 yields $(3^{2m} + 3^m)/2$ free orbits if $n = 2m$ is even, and $(3^{2m+1} + 3^{m+1})/2$ free orbits if $n = 2m + 1$ is odd.

3. The groups $\langle(01)\rangle$, $\langle(12)\rangle$, $\langle(20)\rangle$. We study $\langle(01)\rangle$. Only the word $w = 2^n$ (n copies of 2) satisfies the equation $(01)w = w$. Hence the orbit $\{2^n\}$ is not free. The remaining words split up into $(3^n - 1)/2$ free orbits of order 2.

4. The groups $\langle(01)r\rangle$, $\langle(12)r\rangle$, $\langle(20)r\rangle$. We study $\langle(01)r\rangle$.

   Suppose $n = 2m$ is even and $w = uv$. The relation $uv = (01)r(uv) = ((01)rv)((01)ru)$ implies $v = (01)ru$ and so is satisfied by 3^m words which form 3^m non-free orbits of order 1. The remaining words split up into $(3^{2m} - 3^m)/2$ free orbits.

   Suppose $n = 2m+1$ is odd and $w = uav$. The relation $uav = (01)r(uav) = ((01)rv)((01)a)((01)ru)$ implies $v = (01)ru$ and $a = 2$ is satisfied by 3^m words which form 3^m non-free orbits of order 1. The remaining words split up into $(3^{2m+1} - 3^m)/2$ free orbits.

5. The group $\langle(012)\rangle$. All orbits are of the form $\{w, (012)w, (210)w\}$ with pairwise different elements. This shows that N partitions into 3^{n-1} free orbits.

6. The group $S_3 = \langle(01), (12)\rangle$. The orbit $\{0^n, 1^n, 2^n\}$ is not free because $(01)2^n = 2^n$. The remaining words split up into $(3^{n-1} - 1)/2$ free orbits of order 6.

7. The groups $\langle(01), r\rangle$, $\langle(12), r\rangle$, $\langle(20), r\rangle$. We study $\langle(01), r\rangle$. Only the word 2^n satisfies the relation $w = rw = (01)w = (01)rw$. The corresponding orbit $\{2^n\}$ is not free.

   Suppose $n = 2m$ is even. There are two ways that the orbit $\{w, rw, (01)w, (01)rw\}$ can fold up into orbits of order 2.

   First, if $w = rw$ and $w \neq (01)w$ holds. From the set of 3^m words that satisfy $w = rw$ we remove 2^n. The remaining words in this set split up into $(3^m - 1)/2$ free orbits of order 2.

   Second, if $rw = (01)w$ and $rw \neq w$ holds. If $w = uv$ the relation $rw = (01)w$ implies $v = (01)ru$. As above, we remove from the set of 3^m words

that satisfy this relation the word 2^n to obtain $(3^m-1)/2$ non-free orbits of order 2.

The remaining $3^{2m}-2(3^m-1)-1$ words in N split up into free orbits of order 4. Summing up the free orbits, we obtain for their number $(3^{2m}-2(3^m-1)-1)/4+(3^m-1)/2=(3^{2m}-1)/4$.

Suppose $n=2m+1$ is odd. We consider again the two different types of orbits of order 2.

The first occurs, if $w=rw$ and $w\neq(01)w$ holds. A similar calculation as above obtains $(3^{m+1}-1)/2$ free orbits of order 2.

The second occurs, if $rw=(01)w$ and $rw\neq w$. If w is written as uav, the relation becomes $(rv)a(ru)=((01)u)((01)a)((01)v)$, yielding the constraints $v=(01)ru$ and $a=2$ which are satisfied by 3^m words. Removing the word 2^n from this set results in $(3^m-1)/2$ non-free orbits of order 2.

The remaining $3^{2m+1}-(3^{m+1}-1)-(3^m-1)-1=(3^{m+1}-1)(3^m-1)$ words in N split up into free orbits of order 4. Hence the total number of free orbits is $(3^{m+1}-1)(3^m-1)/4+(3^{m+1}-1)/2=(3^{m+1}-1)(3^m+1)/4$.

8. The group $\langle(012),r\rangle$. If $w=rw$, then $\{w,(012)w,(210)w\}$ is an orbit of order 3. If $w\neq rw$, the orbit containing w is of order 6. The calculation is similar to the one of the group $\langle r\rangle$. If $n=2m$ is even, N partitions into $(3^{2m-1}+3^{m-1})/2$ free orbits, if $n=2m+1$ is odd, N partitions into $(3^{2m}+3^m)/2$ free orbits.

9. The group $\langle(01)r,(012)\rangle$. An orbit is of order 3 if and only if the orbit contains a word w that satisfies the relation $w=(01)rw$.

   Suppose $n=2m$ is even and write $w=uv$. Then $w=(01)rw$ holds if and only if $v=(01)ru$ holds. Each of these 3^m words is in a different orbit. This shows that there are 3^m orbits of order 3. None of them is free. The remaining words split up into $(3^{2m-1}-3^m)/2$ free orbits of order 6.

   Suppose $n=2m+1$ is odd and write $w=uav$. Then $w=(01)rw$ holds if and only if $v=(01)ru$ holds and $a=2$. Again, this shows that there are 3^m non-free orbits of order 3. The remaining words split up into $(3^{2m}-3^m)/2$ free orbits of order 6.

10. Subgroup S_3R. The set $\{0^n,1^n,2^n\}$ is the only orbit of order 3. This orbit is not free.

    Suppose $n=2m$ is even. By Lemma 6 the next possible order of an orbit is 6. If w is a word in an orbit of order 6, either $w=rw$ or $w=\alpha r$, where α denotes a transposition. There are 3^m words for which $w=rw$. If we subtract the 3 words of the orbit of order 3, we obtain $3(3^{m-1}-1)$ words that split up into $(3^{m-1}-1)/2$ free orbits. There are also 3^m words for which $w=(01)rw$, or 3×3^m words for which $w=\alpha rw$, where α denotes any of the three transpositions $(01),(12),(20)$. If we subtract the 3 words of the orbit of order 3, we obtain $3(3^m-1)$ words that split up into $(3^m-1)/2$ non-free orbits.

    The remaining words of N partition into orbits of order 12. The number of remaining words is $3^{2m}-3(3^{m-1}-1)-3(3^m-1)-3$, splitting up into

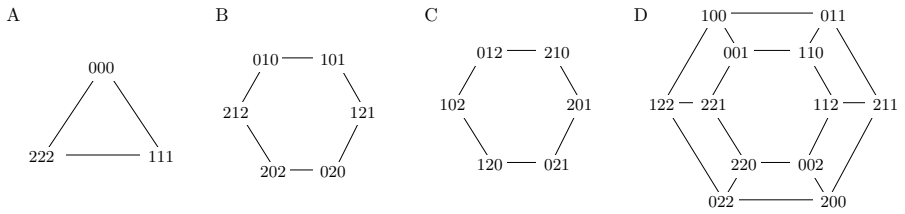

Figure 3: The orbits of S_3R acting on $\{0,1,2\}^3$.

$(3^m-1)(3^{m-1}-1)/4$ free orbits of order 12. The total number of free orbits then is $(3^m-1)(3^{m-1}-1)/4+(3^{m-1}-1)/2 = (3^m+1)(3^{m-1}-1)/4$.

Suppose $n = 2m+1$ is odd. There are 3^{m+1} words that satisfy $w = r$. Similar to the calculation above, we obtain $3^{m+1}-3$ words that split up into $(3^m-1)/2$ free orbits. The relation $ucv = (01)r(ucv)$ implies $v = (01)r$ and $c = 2$. As above, we see that $3^{m+1}-3$ words satisfying $w = \alpha rw$ split up into $(3^m-1)/2$ non-free orbits.

The remaining words of N partition into orbits of order 12. The number of remaining words is $3^{2m+1}-6(3^m-1)-3$, splitting up into $(3^m-1)^2/4$ free orbits of order 12. The total number of free orbits then is $(3^m-1)^2/4+(3^m-1)/2 = (3^{2m}-1)/4$. Fig. 3 depicts the orbits of Σ^3. The graphs are informal, highlighting certain symmetries of the orbits.

5.3 Calculation of the Equivalence Classes

Too shorten the notation, we enumerate the different types of subgroups: $H_1 = S_3R$, $H_2 = \langle (01)r, (012)\rangle$, $H_3 = \langle (01), (12)\rangle$, $H_4 = \langle (012), r\rangle$, $H_5 = \langle (01), r\rangle$, $H_6 = \langle (012)\rangle$, $H_7 = \langle (01)r\rangle$, $H_8 = \langle (01)\rangle$, $H_9 = \langle r\rangle$, and $H_{10} = \langle 1\rangle$. We set $a_i = |\operatorname{ivt}(H_i)|$, $b_i = |\operatorname{stb}^{-1}(H_i)|$, and we let c_i denote the number of equivalence classes of $\operatorname{stb}^{-1}(H_i)$. In the previous subsection we determined the number of free orbits for each of the H_i. Let k_i the number of free orbits of H_i. Then $a_i = 3^{k_i}$. We express now b_i and c_i in terms of a_i. Then

$$
\begin{aligned}
&b_1 = a_1 = c_1;\\
&b_2 = a_2 - a_1, c_2 = b_2/2;\\
&b_3 = a_3 - a_1, c_3 = b_3/2;\\
&b_4 = a_4 - a_1, c_4 = b_4/2;\\
&b_5 = a_5 - a_1, c_5 = 3b_5/3 = b_5;\\
&b_6 = a_6 - a_1 - b_2 - b_3 - b_4 = a_6 + 2a_1 - a_2 - a_3 - a_4, c_6 = b_6/4;\\
&b_7 = a_7 - a_1 - b_2 - b_5 = a_7 + a_1 - a_2 - a_5, c_7 = 3b_7/6 = b_7/2;\\
&b_8 = a_8 - a_1 - b_3 - b_5 = a_8 + a_1 - a_3 - a_5, c_8 = 3b_8/6 = b_8/2;\\
&b_9 = a_9 - a_1 - b_4 - 3b_5 = a_9 + 3a_1 - a_4 - 3a_5, c_9 = b_9/6\\
&b_{10} = a_{10} - a_1 - b_2 - b_3 - b_4 - 3b_5 - b_6 - 3b_7 - 3b_8 - b_9\\
&\quad = a_{10} - 6a_1 + 3a_2 + 3a_3 + a_4 + 6a_5 - a_6 - 3a_7 - 3a_8 - a_9, c_{10} = b_{10}/12.
\end{aligned}
$$

The equations for c_5, c_7, and c_8 contain an additional factor 3. This results from the fact that the corresponding equivalence classes are not contained in

only one set of the partition of the rule space, but are distributed over 3 sets. For instance, if $f \in \mathrm{stb}^{-1}((01))$, then the equivalence class $[f]$ intersects $\mathrm{stb}^{-1}((01))$, $\mathrm{stb}^{-1}((12))$, and $\mathrm{stb}^{-1}((20))$.

The total number of classes is

$$c = \sum_{i=1}^{10} c_i = \frac{1}{12}\left(a_9 + a_{10}\right) + \frac{1}{6}\left(a_4 + a_6\right) + \frac{1}{4}\left(a_7 + a_8\right).$$

Hence we obtain the following proposition.

Proposition 8 *Let m be a nonnegative integer. The rule space $L(3, 2m+1)$ partitions into*

$$\begin{aligned} c &= \frac{1}{12}\left(3^{(3^{2m+1}+3^{m+1})/2} + 3^{3^{2m+1}}\right) + \frac{1}{6}\left(3^{(3^{2m}+3^{m})/2} + 3^{3^{2m}}\right) \\ &+ \frac{1}{4}\left(3^{(3^{2m+1}-3^{m})/2} + 3^{(3^{2m+1}-1)/2}\right) \end{aligned}$$

equivalence classes. Let m be a positive integer. The rule space $L(3, 2m)$ partitions into

$$\begin{aligned} c &= \frac{1}{12}\left(3^{(3^{2m}+3^{m})/2} + 3^{3^{2m}}\right) + \frac{1}{6}\left(3^{(3^{2m-1}+3^{m-1})/2} + 3^{3^{2m-1}}\right) \\ &+ \frac{1}{4}\left(3^{(3^{2m}-3^{m})/2} + 3^{(3^{2m}-1)/2}\right). \end{aligned}$$

equivalence classes.

Table 2 lists the number of of free orbits and equivalance classes for a neighborhood of 2 and 3 cells. The last row gives the total number of equivalence classes.

6 Validation

In the previous sections the exact number of equivalence classes for one-dimensional 2-state and 3-state CA was calculated. This section takes another approach and describes an algorithmic brute-force approach to determine these numbers for small k and n. The algorithm is implemented in Python 3 and depicted in Table 3.

We start with some general considerations applicable to any programming language that supports arrays (referred to as sequences in Python). The state set $\Sigma = \{0, 1, \ldots, k-1\}$ is ordered, and so is the set Σ^n if we adopt the lexicographical order. We write the set Σ^n as an increasing sequence (w_i); $0 \le i < k^n$. This arrangement allows for the representation of a local rule f by the sequence (b_i), $0 \le i < k^n$, where $b_i = f(w_i)$. We define an encoding function, denoted by enc, which maps a word to an integer. The function returns the index i of the word w within the sequence (w_i) such that $w = w_i$, or equivalently, the numerical value when w is read as a number in base k: if $w = a_{n-1} \ldots a_0$, then $\mathrm{enc}(w) = a_{n-1}k^{n-1} + \cdots + a_0$. Given a local rule f represented by the sequence $b = (b_i)$ and a word w, to find $f(w)$ compute $j = \mathrm{enc}(w)$, and then access the j-th element in the sequence b: $f(w) = b_j$.

Table 2: Number of Free Orbits and Equivalence Classes $k = 3, n = 2, 3$.

	n=2		n=3	
H	#free orbits	#classes	#free orbits	#classes
$\langle(01),(12),r\rangle$	0	1	2	9
$\langle(01),(12)\rangle$	1	2	4	36
$\langle(012),r\rangle$	2	4	6	360
$\langle(01)r,(012)\rangle$	0	0	3	9
$\langle(01),r\rangle$	2		8	
$\langle(12),r\rangle$	2	8	8	6 552
$\langle(20),r\rangle$	2		8	
$\langle(012)\rangle$	3	4	9	4 716
$\langle(01)\rangle$	4		13	
$\langle(12)\rangle$	4	81	13	793 872
$\langle(20)\rangle$	4		13	
$\langle(01r)\rangle$	3		12	
$\langle(12r)\rangle$	3	9	12	262 404
$\langle(20r)\rangle$	3		12	
$\langle r\rangle$	6	116	18	64 566 684
$\langle 1\rangle$	9	1 556	27	635 433 642 324
		1 734		635 499 276 966

The program listed in Table 3 implements local rules and words as sequences. The symmetry operators are implemented in a manner closely aligned with their theoretical definitions.

We first discuss the reflection operator. The variable `rfl_pairs` refers to a sequence of integer pairs (i, j) satisfying the relations $w_j = rw_i$ and $w_i \neq rw_i$ (`w[::-1]` is a Python idom used to reverse a sequence). The function `reflectRule` takes a sequence `f` that represents a rule, and returns its reflected version `g`. Initally, the rule provided as argument is copied into the variable `g`. Then, a `for` loop iterates over `refl_pairs`, modifying the values of `g` accordingly to the pairs of `reflectRule`.

We now shift focus to the implementation of the permuation operator. Permutations of the state set are represented as sequences of lenght `k`. The function `permutateRule` accepts a local rule `f` and a permutation, and returns the permutated rule `g`. It begins with initializing the variable `g` with a sequence of length k^n. A `for` loop then iterates all words in the domain, and for each word `w`, `g` is changed, according to the equation $g(\sigma(a_0 \dots a_{n-1})) = g((\sigma a_0) \dots (\sigma a_{n-1})) = \sigma f(a_0 \dots a_n)$.

The function `equiClass` determines the equivalence class of the input function `f`. It iterates all permutations of Σ, and adds the permutated rule and the permutated and reflected rule to the equivalence class. If f is invariant under a certain operation then f will not be changed. But since the underlying data structure of the equivalence class is a `set`, subsequent additions of the same element have no effect.

Lastly `countEquiClasses` iterates the rule space: `it.product(s,repeat=k**n)` creates the cartesian product of Σ with itself, k^n times. If a rule belongs to an equivalence class of an already processed rule, the body of the `for` loop is

Table 3: Python 3 program that calculates the number of equivalence classes.

```
mport itertools as it
k = 2; n = 4                    # number of states; neighborhood size
s = tuple(range(k))             # state set (0,1,..,k-1)
def enc(w) :                    # encodes a word
    v = 0
    for a in w : v = k*v+a
    return v                    # returns  w[0]*k^(n-1) +..+w[n-1]
rfl_pairs = [(enc(w),enc(w[::-1])) for w in it.product(s, repeat=n)
    if w != w[::-1]]            # pairs (i,j), w_j = rw_i != w_i
def reflectRule(f) :            # returns reflected rule
    g = list(f)                 # copy f
    for (i,j) in rfl_pairs : g[i] = f[j]
    return tuple(g)
def permutateRule(f, perm) :    # returns permutated rule
    g = [0] * k**n
    for w in it.product(s, repeat=n) :
        g[enc([perm[a] for a in w])] = perm[f[enc(w)]]
    return tuple(g)
def equiClass(f) :              # returns the equivalence class
    c = set()
    for perm in it.permutations(s) :
        pf = permutateRule(f,perm)
        c.update({pf, reflectRule(pf)})
    return tuple(c)
def countEquiClasses() :        # counts all equivalence classes
    processed = set()           # keep track of processed rules
    count = 0
    for f in it.product(s,repeat=k**n) :
        if f not in processed :
            count += 1
            processed.update(equiClass(f))
    return count
print(countEquiClasses())       # prints number of classes
```

skipped. Otherwise a new equivalence class is created, the counter is incremented, and the members of the class are stored in a set referenced by the variable `processed`.

On a typical PC, the program prints the result within a few seconds for the input parameters $k = 2$ and $n \leq 4$ as well as for $k = 3$ and $n \leq 2$. With an optimized implementation and improved hardware, it might be possible to achieve results for a few additional combinations, such as $k = 2$, and $n = 5$. However, the algorithm's runtime complexity prevents calculations for larger input parameters.

The presented implementation is minimalistic. We briefly explore two kinds of improvements.

1. By incorporating minor changes, more detailed insights about the equivalence classes can be obtained. As an example, adding a hashtable to the function body of `countEquiClasses` allows tracking the number of equivalence classes based on their order.

2. Although traversing the entire domain might be unfeasible, exploring only parts of it might still be instructive. For instance, if the main loop is adjusted to iterate only the set $\mathrm{ivt}(\langle(012)\rangle)$, it is feasible to obtain the results for the upper lattice of S_3R, which consists of groups that encompass the group $\langle(012)\rangle$.

References

[1] G. Cattaneo, E. Formenti, L. Margara, and G. Mauri. Transformations of the one-dimensional cellular automata rule space. *Parallel Computing*, 23(11):1593–1611, 1997. Cellular automata.

[2] Karel Culik and Sheng Yu. Undecidability of ca classification schemes. *Complex Syst.*, 2(2):177–190, April 1988.

[3] D.S. Dummit and R.M. Foote. *Abstract Algebra*. Wiley, 2003.

[4] G. A. Hedlund. Endomorphisms and automorphisms of the shift dynamical systems. *Math. Syst. Theory*, 3(4):320–375, 1969.

[5] Wentian Li and Norman H. Packard. The structure of the elementary cellular automata rule space. *Complex Syst.*, 4, 1990.

[6] James S. Milne. Group theory (v4.00), 2021. Available at www.jmilne.org/math/.

[7] Eurico L.P. Ruivo, Pedro P.B. de Oliveira, Fabiola Lobos, and Eric Goles. Shift-equivalence of k-ary, one-dimensional cellular automata rules. *Communications in Nonlinear Science and Numerical Simulation*, 63:280–291, October 2018. Publisher Copyright: © 2018 Elsevier B.V.

[8] Vispoel, Milan and Daly, Aisling and Baetens, Jan. Progress, gaps and obstacles in the classification of cellular automata. *PHYSICA D-NONLINEAR PHENOMENA*, 432:30, 2022.

[9] S. Wolfram. *Theory and Applications of Cellular Automata: Including Selected Papers, 1983-1986*. Advanced series on complex systems. World Scientific, 1986.

[10] Stephen Wolfram. Statistical mechanics of cellular automata. *Reviews of Modern Physics*, 55(3):601–644, July 1983.

[11] Stephen Wolfram. Universality and complexity in cellular automata. *Physica D: Nonlinear Phenomena*, 10(1):1–35, 1984.