Minimum Equivalence in Random Boolean Networks, Elementary Cellular Automata, and Beyond

Tom Eivind Glover[1,*], Ruben Jahren[1], Ola Huse Ramstad[3] and Stefano Nichele[1,2]

[1] Department of Computer Science, OsloMet - Oslo Metropolitan University, Oslo, Norway
[2] Department of Computer Science and Communication, Østfold University College, Halden, Norway
[3] Department of Neuromedicine and Movement Science,
Norwegian University of Science and Technology, Trondheim, Norway
[*] tomglove@oslomet.no

Abstract

Random Boolean networks (RBN) and Cellular Automata (CA) operate in a very similar way. They update their state with simple deterministic functions called Boolean function or Transition Table (TT), both being essentially the same mechanism under different names. This paper applies a concept most known from CA called Minimum Equivalence (ME). ME is applied to RBN and shows how to calculate the number of unique computations for a given number of neighbours. Crucially, it is shown how RBN rules are *even more equivalent* than in CA, how the set can be reduced into even fewer unique rules, and how the concept becomes more relevant with larger neighbourhoods. For example, switching transformation alone reduces the number of unique rules in RBN with 4 neighbours from 65 536 to only 3 984 (6.1%) rules. Additionally, this paper examines the ME and transformations in substrates beyond Elementary CA (ECA), such as CA with additional spatial dimensions and number of states.

Introduction

CA, RBN and everything in between has long been used as models to understand computation in biology. von Neumann (1966) demonstrated how a CA can be self-replicating. Kauffman (1969) suggested using RBN to model the Gene Regulatory network. Walker (1971, 1965); Walker and Ashby (1966) studied an intermediate substrate between CA and RBN to understand an organism's behaviour. Wuensche et al. (1992); Wuensche (1992, 1994, 1997) investigated the attractor basin of CA, RBN and intermediate substrates, and it even goes as far as to suggest it is the *Ghost in the Machine*, referring to it as a cognitive substrate.

As early as 1965, the possible trivial transformations that can be done to the rule-space to find equivalent rules were discussed (Walker, 1965, p. 51, p. 176). While they were not presented in ECA, they were introduced in a substrate close enough that the ME would be the same in such a substrate. In (Wolfram, 1986, 2018; Li and Packard, 1990; Wuensche et al., 1992), the 88 ME rules of ECA were demonstrated and made accessible. In (Li, 1992), Li briefly explained what changes for the ME when applied in a substrate closer to RBN. In (Wuensche, 1997, p. 4), ME was also very briefly mentioned in a 2D CA context. The few studies that propose the concept outside of ECA, explain the concept only briefly. Therefore, this paper suggests it is of value to examine the ME principle and how it changes in relevant substrates and illustrate how to calculate it.

This paper builds on the ME principle and applies it to RBN, demonstrates how it can be calculated for any given number of neighbours, and propose an algorithm for calculating it. It demonstrates how the ME in RBN is even more equivalent than in CA. Furthermore, it demonstrates how the reduction in unique computation grows with the number of neighbours. In addition, ME in CA beyond ECA is examined, and what happens when adding more spatial dimensions and possible states is demonstrated. This paper's findings suggest there is untapped potential in many substrates for efficiency, particularly in exhaustive search.

Background

Cellular Automata

CA is a simple substrate consisting of cells in limited distinct states. The cells are connected uniformly and change state synchronously depending on the configuration of states of their immediate neighbours in the grid. How they change is determined by a deterministic lookup table called the Transition Table (TT). ECA is a subset of CA in 1-dimension, binary states ($S = 2$) and 3 neighbours ($K = 3$) (left, right and centre). Therefore, ECA only has $S^{S^K} = 2^{2^3} = 256$ possible rules, and the whole set of these is often named the rule-space. It is a convention to name individual rules in a rule-space after the output states of the TT $\text{Binary}(01011010) = \text{Decimal}(90)$. CA is deterministic, and the rule, together with the initial condition, leads the

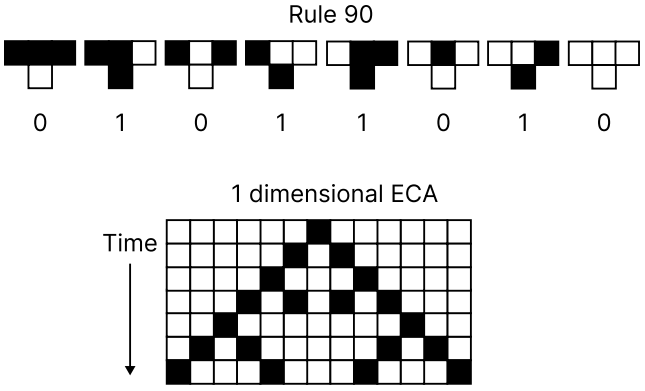

Figure 1: Example of 1 dimensional CA with rule 90 with TT, starting from a central cell on, executing 7 time-steps.

CA into a set of subsequent states called the trajectory. An example of rule 90 can be seen in Figure 1.

Beyond ECA are many other types of CA, such as 2-dimensional CA seen in Figure 2. In this substrate, the most typical neighbourhood scheme is one of two configurations in Figure 3.

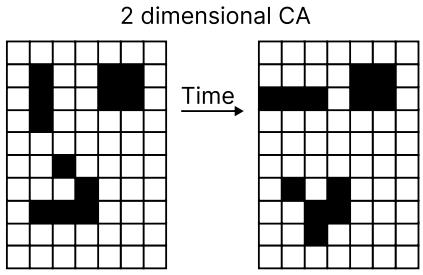

Figure 2: Single time-step of a 2-dimensional CA with Conway's Game of Life rules. (Gardner, 1970)

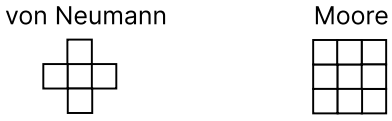

Figure 3: Common 2-dimensional neighbourhood schemes.

Many additional forms of CA exist, e.g. Stochastic CA (Pontes-Filho et al., 2022), Asynchronous CA (Fates, 2013), Continuous CA (Chan, 2020), a combination (Mordvintsev et al., 2020; Variengien et al., 2021) or beyond simple classification (Nichele et al., 2016, 2017). Equivalences can be made in these substrates as well. However, this work limits itself to deterministic, discrete and synchronous substrates.

Random Boolean Networks

The RBN is similar to a CA yet has two key differences. Firstly, in the RBN, the grid neighbour connections are not regular but randomly set up. Secondly, every node (cell) typically has a random TT, often called an Activation function or Boolean function. This type of RBN is also sometimes called Classical RBN (CRBN) (Gershenson, 2002). The number of direct neighbours can be random, semi-random or constant. The latter is called homogeneous RBN (Gershenson, 2002), an example is given in Figure 4.

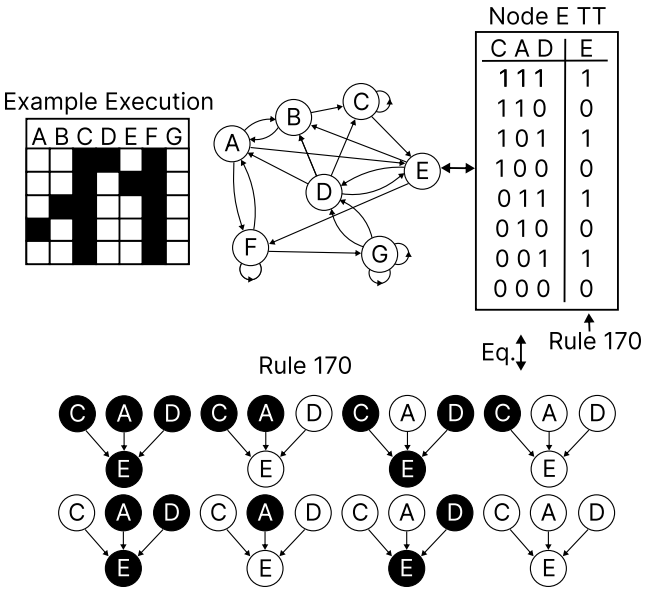

Figure 4: Example of an RBN with 7 Nodes and 3 neighbours, with a transition table in two forms and a short execution example.

As with CAs, several extensions exist beyond the original RBN, such as Continuous RBN (Vohradsky, 2001) or stochastic RBN (Ribeiro et al., 2006; Elowitz et al., 2002). While Kauffman first developed the CRBN to model gene regulatory networks, these more modern extensions to the RBN model better emulate the biological activity of development (Vohradsky, 2001; Elowitz et al., 2002). However, the CRBN were discovered early on (Kauffman, 1971) to contain a limited number of stable states, or attractors, from which the system would settle down to following a random initialisation. The basin of attraction reduces numerous initial states to a few stable cycles or fixed points.

Intermediate Substrates: Homogeneous Homogeneous RBN (HHRBN), Non-local CA, Disordered CA

A system can be in a range of possible states that would be somewhere between CA and RBN. This paper will discuss

a substrate with homogeneous rules but random neighbour wiring. Such a substrate will be called Homogeneous Homogenous RBN (HHRBN). HHRBN to distinguish it from what is in Gershenson (2002) called HRBN. It is called HHRBN rather than non-local CA because the substrate seems to behave more like RBN than CA, and the equivalence in this substrate is more applicable in RBN than CA.

Li (1992) worked with systems where all cells had the same activation function (TT), but the neighbour connections were in various configurations. Classifying the different connection schemes as between non-local (random) and partially-local (central self-reference) as well as non-distinct and distinct input/output (uniform number of outputs). Li then classifies the rule-space for these substrates using mean field approximation and shows they are very neatly classified, particularly non-local CA (HHRBN).

Much earlier Walker (1965, 1971); Walker and Ashby (1966) studied a system that Li would classify as partially-local CA.

HHRBN has additional commonly used names beyond non-local CA, such as Graph CA (Marr and Hütt, 2009; Grattarola et al., 2021) or (Cellular) Automata Networks (Bhattacharjee et al., 2020)

Wuensche (1992) examined substrates between CA and RBN, including non-local CA but also other disordered CA. Wuensche defines disordered CA as a super-set of CA which includes non-local CA and mixed rule CA. Furthermore, Wuensche calculates these networks' basin of attraction fields and demonstrates how rewiring the network can train or modify the basin of attraction.

Mixed rule CA is also known as Non-uniform CA (Cattaneo et al., 2009; Bhattacharjee et al., 2020) or hybrid CA (Bhattacharjee et al., 2020)

Mathematical definition of RBN

HHRBN can be defined as the following. A set of N nodes connected randomly to K number of other nodes, the specific connections for a given node can be denoted by K_N. The nodes can be in one of the two binary states, and every N has the same activation function f_a (TT), out of 2^{2^K} possible rule setups. If the requirements are relaxed to HRBN, every node has a random activation function. If relaxed to RBN, then K varies between nodes.

RBN Classification

ECA is often partitioned and classified into several different categories or traits. In (Martinez, 2013), a good overview of many common or well-known ones can be found.

Similarly, RBN can be classified by their behaviour, i.e. ordered, complex or chaotic (Gershenson, 2002; Kauffman et al., 1993). Depending on the value of N and K, the behaviour might differ, and one alternative name for RBN is NK model. Kauffman (1990) added another parameter P which can organise the rule-space. The rule has a given P parameter value based on the number of neighbourhood combinations resulting in a 1 or a 0. In later work (Kauffman et al., 1993), the larger distribution dominates, meaning $P \geq 0.5$. Figure 4 has $P = 0.5$. One can use this parameter to control the behaviour. P close to 1 would likely result in ordered behaviour, and P close to 0.5 would likely result in chaotic behaviour. In between these, a critical (complex) P_c behaviour might be found in the phase transition between order and chaos. This point or border is often also called the edge of chaos. The work is reminiscent of CA work in (Langton, 1990).

Another way to categorise RBN and CA is to look at the basin of attraction. Wuensche et al. (1992); Wuensche (1992, 1994, 1997) did extensive work in both RBN and CA and their basin of attractions. What opened up this possibility was a method that could calculate backwards from a state. Take a cell in a state and consider what possible local neighbourhood configurations would result in this state. These are the possible previous states (preimage) for the neighbourhood. Finally, apply this for all the cells and limit the possibilities between cells by constraint satisfaction. The possible preimages often collapse to very few, making it possible to quickly calculate the basin of attraction.

Rule	Equivalent	Rule	Equivalent	Rule	Equivalent
0	255	35	49,59,115	108	201
1	127	36	219	110	124,137,193
2	16,191,247	37	91	122	161
3	17,63,119	38	52,155,211	126	129
4	223	40	96,235,249	128	254
5	95	41	97,107,121	130	144,190,246
6	20,159,215	42	112,171,241	132	222
7	21,31,87	43	113	134	148,158,214
8	64,239,253	44	100,203,217	136	192,238,252
9	65,111,125	45	75,89,101	138	174,208,244
10	80,175,245	46	116,139,209	140	196,206,220
11	47,81,117	50	179	142	212
12	68,207,221	51		146	182
13	69,79,93	54	147	150	
14	84,143,213	56	98,185,227	152	188,194,230
15	85	57	99	154	166,180,210
18	183	58	114,163,177	156	198
19	55	60	102,153,195	160	250
22	151	62	118,131,145	162	176,186,242
23		72	237	164	218
24	66,189,231	73	109	168	224,234,248
25	61,67,103	74	88,173,229	170	240
26	82,167,181	76	205	172	202,216,228
27	39,53,83	77		178	
28	70,157,199	78	92,141,197	184	226
29	71	90	165	200	236
30	86,135,149	94	133	204	
32	251	104	233	232	
33	123	105			
34	48,187,243	106	120,169,225		

Table 1: The group of equivalent rules for ECA.

Minimum Equivalence (ME)

ECA consists of $2^{2^3} = 256$ rules, but due to symmetries and other properties, there are only 88 rules that are considered unique. The reason is that all excluded rules can be

transformed into one of the 88 unique rules by one of the following trivial methods.

- reflection: switching left and right
- complement: switching 0 and 1
- reflection and complement: the combination of both transformations

An overview of the 88 rules can be found in Table 1. examples of the transformed rules can be found in Figure 5 and 6.

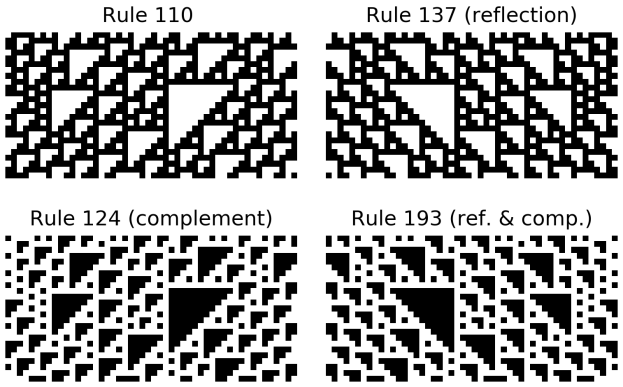

Figure 5: Reflection, complement and reflection complement transformation of rule 110 and equivalent with random initialisation. Reflection is initialised with a mirrored state and complement with a flipped value state.

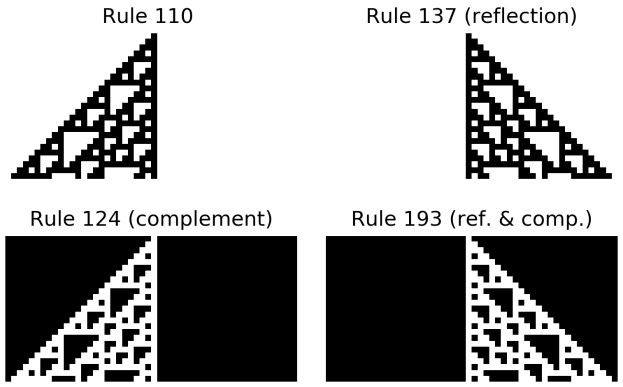

Figure 6: Reflection, complement and reflection complement transformation of rule 110 and equivalent with centroid initialisation. Reflection is initialised with a mirrored state and complement with a flipped value state.

The concept of reflection and complement seems to originate already in (Walker, 1965, p. 51, p. 176), and more densely explained by the same author in Walker (1971). In the previous source, the concept originated in an intermediate substrate between ECA and RBN, with two random neighbours and itself. The ME concept works the same in such a substrate, but the concept is perhaps best known when applied to ECA in Wolfram (1986, 2018); Li and Packard (1990); Wuensche et al. (1992).

Equivalence in RBN

This section will explain how to apply the concept of ME to RBN and provide enough insight that the reader might also apply the concept beyond RBN. Therefore, this and the following sections should be considered the main contributions of this paper.

Switching neighbours

In RBN, the order of the K_N is arbitrary and randomly set up with the topology of the RBN. It is not featured in most illustrations, see Figure 4 and 7.

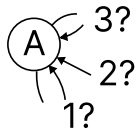

Figure 7: The topology map of an RBN does not normally specify the order of the neighbours

This order is arbitrary, it is equally likely to get one order over any of the other 6 orders possible for $K = 3$ illustrated in Figure 8

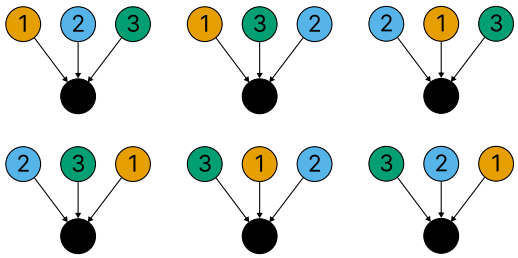

Figure 8: Possible input orders for RBN with $K = 3$, image and future coloured images uses colourblind friendly colour palette from Wong (2011)

One can trivially transform the rule to another by shifting the corresponding TT entries as illustrated for rule 170 in Figure 9 This transformation can be done for every topology. Therefore, the rule transformed must also produce the same space of trajectories and distribution for that space. For every trajectory, it is just as likely to get the corresponding equivalent when creating the RBN. Therefore, they are just as likely to appear. Any deviation in the result of any experiment from equivalent rules should only be indistinguishable

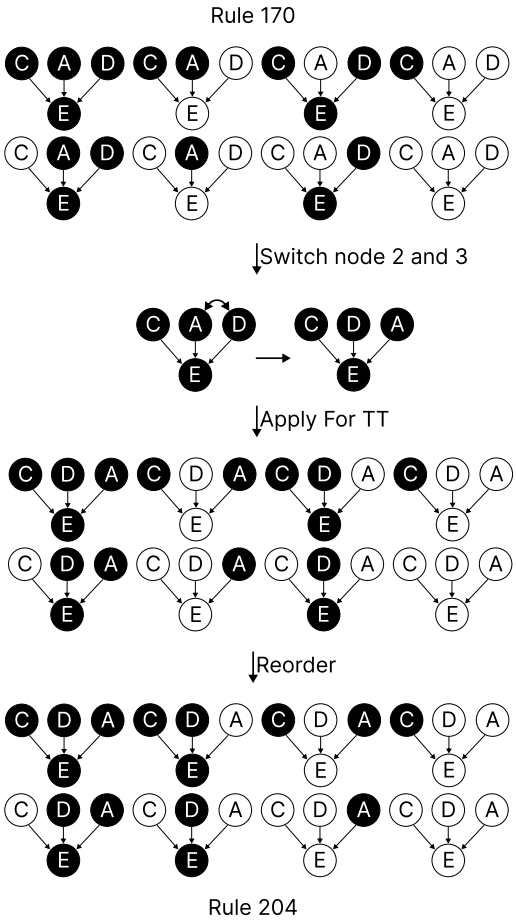

Figure 9: Taking the TT from Figure 4, this demonstrates switching the central and left neighbour turns rule 170 into rule 204.

from pure luck, granted that the generation of the RBN is random.

In CA, the reflection equivalence is only equivalent under trivial transformation. In RBN, this switching transformation is also equivalent under trivial transformation but also additionally equivalent from having the same probability of hitting the same trajectory in the space of trajectories. Therefore, they are *more equivalent* than the same concept in CA.

Algorithm

The switching transformation can be generalised into an algorithm. First, find the unique possible orders for a particular K, then find switching pathways to get to that condition.

Number every sub-value in the permutation $0...n-1$. If the subset values are not in order, they must be switched. However, as multiple values can be out of order, it is also necessary to find the path or combination of the switching as illustrated in Figure 10. This correct path can be found by brute force or shuffling the order until it is correct. As the

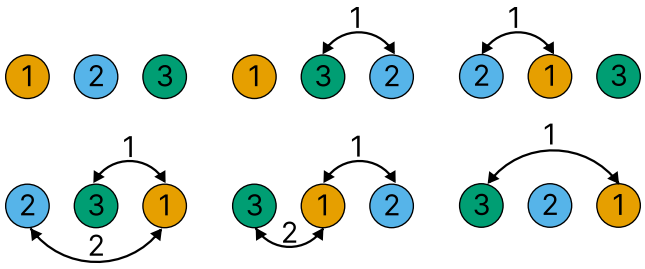

Figure 10: Transformation for 6 possible equivalences. The figure includes possible switching orders

next portion of the full algorithm is on a different scale, the wasted computation by shuffling here is negligible. Pseudocode for how this part can be done is presented in Algorithm 1.

Algorithm 1 How to find correct switching path

```
PE ← permutations for K
for each P in PE do
    S ← []
    for each in order subset of size 2 P_s in P do
        if P_{s_0} > P_{s_1} then S append P_s
        end if
    end for
    while (Apply S on P_0) ≠ P do shuffle S
    end while
end for
```

Once this is prepared, it is simple to go through every rule and apply the transformation on the rule table, then reorder it as displayed in Figure 9 for the full TT. There are also many ways to improve performance, e.g. parallel processing or using a hashmap of all rules found and skipping them in the loop.

An implementation of the algorithm in python code can be found Online [1].

Computational Rule-space of $K = 3$

If the switching transformation is applied for $K = 3$, the following rule-set of only 80 rules represents the computational space, as seen in Table 2. Rules that are totalistic or invariant to direction do not form equivalent groups, rules invariant to a switch form sets of 3, and rules that are variant on all switches form sets of 6.

Complementary transformation and when it can be applied

Compared to CA, the complementary transformation has mostly stayed the same in HHRBN and RBN. Considering

[1] https://github.com/DeepCANFR/RBN_Equivalence/blob/main/rbn_eq.py

Rule	Equivalent	Rule	Equivalent
0		128	
1		129	
2	4, 16	130	132, 144
3	5, 17	131	133, 145
6	18, 20	134	146, 148
7	19, 21	135	147, 149
8	32, 64	136	160, 192
9	33, 65	137	161, 193
10	12, 34, 48, 68, 80	138	140, 162, 176, 196, 208
11	13, 35, 49, 69, 81	139	141, 163, 177, 197, 209
14	50, 84	142	178, 212
15	51, 85	143	179, 213
22		150	
23		151	
24	36, 66	152	164, 194
25	37, 67	153	165, 195
26	28, 38, 52, 70, 82	154	156, 166, 180, 198, 210
27	29, 39, 53, 71, 83	155	157, 167, 181, 199, 211
30	54, 86	158	182, 214
31	55, 87	159	183, 215
40	72, 96	168	200, 224
41	73, 97	169	201, 225
42	76, 112	170	204, 240
43	77, 113	171	205, 241
44	56, 74, 88, 98, 100	172	184, 202, 216, 226, 228
45	57, 75, 89, 99, 101	173	185, 203, 217, 227, 229
46	58, 78, 92, 114, 116	174	186, 206, 220, 242, 244
47	59, 79, 93, 115, 117	175	187, 207, 221, 243, 245
60	90, 102	188	218, 230
61	91, 103	189	219, 231
62	94, 118	190	222, 246
63	95, 119	191	223, 247
104		232	
105		233	
106	108, 120	234	236, 248
107	109, 121	235	237, 249
110	122, 124	238	250, 252
111	123, 125	239	251, 253
126		254	
127		255	

Table 2: The ME set for $K = 3$ without counting complement

Rule	Equivalent
0	255
1	127
2	4, 16, 191, 223, 247
3	5, 17, 63, 95, 119
6	18, 20, 159, 183, 215
7	19, 21, 31, 55, 87
8	32, 64, 239, 251, 253
9	33, 65, 111, 123, 125
10	12, 34, 48, 68, 80, 175, 187, 207, 221, 243, 245
11	13, 35, 47, 49, 59, 69, 79, 81, 93, 115, 117
14	50, 84, 143, 179, 213
15	51, 85
22	151
23	
24	36, 66, 189, 219, 231
25	37, 61, 67, 91, 103
26	28, 38, 52, 70, 82, 155, 157, 167, 181, 199, 211
27	29, 39, 53, 71, 83
30	54, 86, 135, 147, 149
40	72, 96, 235, 237, 249
41	73, 97, 107, 109, 121
42	76, 112, 171, 205, 241
43	77, 113
44	56, 74, 88, 98, 100, 173, 185, 203, 217, 227, 229
45	57, 75, 89, 99, 101
46	58, 78, 92, 114, 116, 139, 141, 163, 177, 197, 209
60	90, 102, 153, 165, 195
62	94, 118, 131, 133, 145
104	233
105	
106	108, 120, 169, 201, 225
110	122, 124, 137, 161, 193
126	129
128	254
130	132, 144, 190, 222, 246
134	146, 148, 158, 182, 214
136	160, 192, 238, 250, 252
138	140, 162, 174, 176, 186, 196, 206, 208, 220, 242, 244
142	178, 212
150	
152	164, 188, 194, 218, 230
154	156, 166, 180, 198, 210
168	200, 224, 234, 236, 248
170	204, 240
172	184, 202, 216, 226, 228
232	

Table 3: The ME set for $K = 3$ with complement

that the switching transformation is more equivalent than its reflection counterpart in CA, the difference between when it is valid to apply switching and complement would seem to have expanded at first. The distinction of which state is quiescent and which is active is arbitrary as long as they are treated the same way in the system, but in many contexts, it is not. It is common in CA to initiate the CA with a majority of one state. e.g. initiate with a central cell as active as in Figure 1. Therefore, the complement rule would only strictly be equivalent if initiated with the opposite states. This requirement also holds in HHRBN, but it becomes more valid again for some special initial conditions, such as the common random initialisation. In a similar manner as for the switching transformation, the random initialisation makes the equivalence valid. It is as likely to get the setup leading to the same trajectory in its normal or complement rule setup. For RBN, this would be more complicated, but still, there are trivial equivalent cases, e.g. if one went through every node and switched it out with the complement rule and switched the initial condition, this RBN would be equivalent.

If applying the complement, switching and combinations is valid, then the rule-set would be represented with only 46 rules (18%), as seen in Table 3.

Discussion

Where it can be applied and beyond HHRBNs

The main use case for the RBN equivalence is for an exhaustive search of the RBN rule space. This is most feasible when working with a workable search space like HHRBN. It can reduce the search space quite effectively. For $K = 3$, it reduces 256 rules to just 46 (a reduction to only 18% of its original size)

This concept also has potential applications outside of HHRBN. For example, many nodes will have the same TT if one simulates a very large RBN (Kauffman (1990) already used RBNs with $N = 10000$ around 30 years ago). In order to save space and lookup time, one could reuse the same TT between nodes with the same rules. This list of lookup tables would be shorter and save some memory space if reduced further to just the ME rules. A smaller memory foot-

print can, on its own, have some optimising effects, e.g. the whole set of tables fitting within a lower layer of memory, cache, or enabling the complete list to exist within a GPUs memory.

It is important to note that when generating RBN's some of the rules have a different number of equivalent rules. This means there is a difference in how likely a rule is picked depending on whether the RBN is generated from the ME rules or the full set. If this is important, it is easy to get past. When generating the RBN use the full set and exchange the rule for the ME afterwards or generate the RBN with a different probability of a rule being picked based on the number of equivalent rules in the equivalence set.

Additionally, for a larger K value, the rule space contains every rule from the smaller K values. The rule-space contains the rules that are invariant to a neighbour, these rules would be equivalent to a rule in a smaller K value. Therefore, any rule can be mapped into larger K, forming equivalences across K values. This can be relevant if working with RBN (distribution of K).

Additionally, ME is helpful without being an optimisation tool. For example, consider Kauffman (1990) and the work with the P parameter. A switching equivalence set would all have the same P parameter value. Thus, it reduces the number of unique computations within a given P parameter set by quite a considerable amount. This shows that the computational variants in a given P parameter value are lower than they would seem without knowing the equivalence.

In comparison to totalistic RBN

In totalistic RBN, rules depend on the number of on and off neighbours instead of distinguishing between cell neighbourhoods. Therefore, it is invariant to the order of K_N. For totalistic HRBN with $K = 3$, only $2^4 = 16$ possible rules would be in the rule-space. These rules would also be contained within a common RBN rule space. The switching equivalence concept can be viewed as unifying many non-totalistic RBNs. As previously mentioned, totalistic RBN rules will have no equivalent rules in the switching equivalence. This is because switching the neighbour of a totalistic rule would turn it into itself again. Considering the engineer's point of view, a fair question might be, "What is the computational difference between the totalistic RBN and the non-totalistic RBN rules, and how could it be useful?".

The equivalence set clarifies that for an RBN, the distinction between left and right holds little meaning, but it does not follow that the rule distinguishes behaviour from nodes 1,2,3 being different. These differences give a potential degree of freedom where some rules behave more or less chaotically, allowing for a finer search for the edge of chaos. If "chaos to order is the only useful variance in RBN" is a correct axiom, and if there is another way to control for chaos to order, the utility of the non-totalistic RBN is replaced by that function instead. There could be a better control parameter. Considering how the behaviour landscape changes when moving to heterogeneity (Sánchez-Puig et al., 2022), perhaps regulating the substrate with the level of heterogeneity is a better control parameter. If such a control parameter exists, then one can show that there is no utility from the rules that cannot be replaced with the control parameter, then the logical conclusion is that there is no point in going beyond totalistic RBN as an engineering tool, for this purpose at least. If this holds, then one can reduce the computational space of RBN to just totalistic rules.

Beyond just $K = 3$

This paper supplied a list of the unique rules for $K = 3$, but RBN equivalence applies to any K.

The number of possible equivalent rules for the switching is $K!$. Therefore, we hypothesise that the larger the K, the more significant the reduction. In Table 4, even at $K = 4$, the search space is already reduced to just 3.1% of its original size.

Minimum Equivalence in non-binary systems

So far, only binary systems have been considered, but the complement transformation could also be relevant in discrete non-binary systems. This transformation would work in a very similar way to switching transformations in RBN. Also, in this substrate, switching paths between possible permutations of the states is relevant, as in Figure 10.

A system with 3 states and 3 neighbours has many possible rule configurations $3^{3^3} = 7\ 625\ 597\ 484\ 987$. With the current level of computational power, it is not impossible, but it is not entirely feasible to calculate the ME set for this rule-space, let alone exhaustively search. Nevertheless, some configurations still make sense. For example, in a totalistic system, the switching or reflection transformation would turn into itself and is not helpful to apply, but this does not hold for the compliment transformation. Therefore, the complement equivalence is still relevant in totalistic rule-space. Furthermore, the size of a totalistic rule-set does not grow at the same level, as seen in Table 5. In addition, a system with $K = 2$ would also be feasible for both transformations.

Minimum Equivalence in 2D CA

In a 2D CA substrate, the reflection transformation changes too. This concept of equivalence in 2D CA was briefly mentioned in (Wuensche, 1997, p. 4), but never explained or fleshed out. In 2D, instead of just reflection, there is now also every neighbourhood rotation. The rotation is quite clearly equivalent, but we hypothesise that the switching between right and left also are equivalent, as seen in Figure 11. Up and down would also be equivalent, but it is possible to get to the same states by combining the left to right and rotation.

K	2	3	4
rules	16	256	65 536
switches	2	6	24
ME switching	12(75%)	80(31%)	3 984(6.1%)
ME switching + compliment	7(44%)	46(18%)	2 036(3.1%)

Table 4: The number of unique rules for different K. It shows a trend for smaller $\%$ of unique computational rules the larger the K.

K	2	3	4
rules	19 683	7 625 597 484 987	$4.4 * 10^{38}$
totalistic rules	729	50 549	14 348 907

Table 5: Rule-space size in a 3-state system

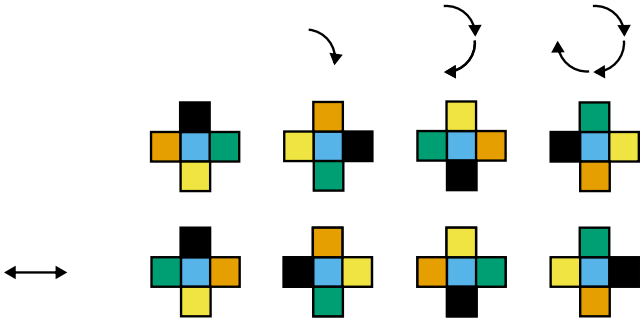

Figure 11: Equivalent symmetries in a 2D von Neumann neighbourhood. Top row: rotation. Bottom row: reflection and combinations

The same would apply to the Moore neighbourhood, but the rule-space is much larger $2^{2^9} = 1.3 * 10^{154}$, so no exhaustive searches are expected anytime soon.

Up until 2D, the transformation would result in an equivalence is apparent, but for this 2D ME, it seems less clear why left-to-right transformation leads to an equivalence or not. Essentially, could there be interference from the new dimension? Some trivial transformations would not lead to equivalent rules because they do not satisfy the same property. The equivalent transformations must uphold that the transformation and any development of that CA to any deterministic subsequent states can be freely transformed back with perfect accuracy. The same could be stated in first-order predicate logic as seen in Equation 1.

$$R_1 \equiv R_2 \Leftarrow \forall x \exists y \exists f \text{ where } f_{x_i} \rightarrow y_i \wedge f^{-1}_{y_{i+z}} \rightarrow x_{i+z} \quad (1)$$

Meaning that one can transform a CA to the equivalent rule and transform the states likewise (e.g. for rotation, also rotate the starting state). If it can be run for an arbitrary number of steps and through this state, know exactly what state the original rule should be. If this is possible for all state, then the rules are equivalent under trivial transformation.

This 2D CA left-to-right equivalence was tested by running 1000 random rules with random initialisation for 10 steps, all successful. Source code can be found online [2]

Conclusion and Future Work

This paper has shown how to apply trivial transformations beyond ECA; into RBN, multi-state, 2D CA or totalistic rules. All the different substrates have their own ME sets, and hopefully, our work has made them more accessible.

It has been demonstrated how these ME sets in specific substrates shrink the rule space significantly. Combined with the increase in computational power now available, we are crossing into the viability to do explorations on the computational space like what was done in ECA, but on substrates beyond ECA.

This work opens up interesting possibilities. Work that was previously limited to ECA can now be extended beyond. For example, the reduction in exhaustive search space enables avenues to explore how the distribution in behaviour maps between substrates.

Acknowledgements

This work was partially financed by the Research Council of Norway's DeepCA project, grant agreement 286558. This work was performed at the OsloMet Artificial Intelligence lab and NordSTAR - Nordic Center for Sustainable and Trustworthy AI Research. Special thanks to Trym Lindell, Barbora Hudcvoá, Michael Tarlton and Akriti Sharma for valuable discussions on various topics in this paper.

References

Bhattacharjee, K., Naskar, N., Roy, S., and Das, S. (2020). A survey of cellular automata: types, dynamics, non-uniformity and applications. *Natural Computing*, 19:433–461.

Cattaneo, G., Dennunzio, A., Formenti, E., and Provillard, J. (2009). Non-uniform cellular automata. In *Language and Automata Theory and Applications: Third International Conference, LATA 2009, Tarragona, Spain, April 2-8, 2009. Proceedings 3*, pages 302–313. Springer.

Chan, B. W.-C. (2020). Lenia and expanded universe. In *ALIFE 2020: The 2020 Conference on Artificial Life*, pages 221–229. MIT Press.

Elowitz, M. B., Levine, A. J., Siggia, E. D., and Swain, P. S. (2002). Stochastic gene expression in a single cell. *Science*, 297(5584):1183–1186.

[2] https://github.com/DeepCANFR/RBN_Equivalence/blob/main/2d_CA_eq.py

Fates, N. (2013). A guided tour of asynchronous cellular automata. In *Cellular Automata and Discrete Complex Systems: 19th International Workshop, AUTOMATA 2013, Gießen, Germany, September 17-19, 2013. Proceedings 19*, pages 15–30. Springer.

Gardner, M. (1970). The fantastic combinations of john conway's new solitaire game "life". *Scientific American*, 223:120–123.

Gershenson, C. (2002). Classification of random boolean networks. *arXiv preprint cs/0208001*.

Grattarola, D., Livi, L., and Alippi, C. (2021). Learning graph cellular automata. *Advances in Neural Information Processing Systems*, 34:20983–20994.

Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22(3):437–467.

Kauffman, S. A. (1971). Cellular homeostasis, epigenesis and replication in randomly aggregated macromolecular systems. *Journal of Cybernetics*, 1(1):71–96.

Kauffman, S. A. (1990). Requirements for evolvability in complex systems: orderly dynamics and frozen components. *Physica D: Nonlinear Phenomena*, 42(1-3):135–152.

Kauffman, S. A. et al. (1993). *The origins of order: Self-organization and selection in evolution*. Oxford University Press, USA.

Langton, C. G. (1990). Computation at the edge of chaos: Phase transitions and emergent computation. *Physica D: Nonlinear Phenomena*, 42(1-3):12–37. Publisher: Elsevier.

Li, W. (1992). Phenomenology of nonlocal cellular automata. *Journal of Statistical Physics*, 68(5):829–882.

Li, W. and Packard, N. (1990). The structure of the elementary cellular automata rule space. *Complex Systems*.

Marr, C. and Hütt, M.-T. (2009). Outer-totalistic cellular automata on graphs. *Physics Letters A*, 373(5):546–549.

Martinez, G. J. (2013). A note on elementary cellular automata classification. *arXiv preprint arXiv:1306.5577*.

Mordvintsev, A., Randazzo, E., Niklasson, E., and Levin, M. (2020). Growing neural cellular automata. *Distill*, 5(2):e23.

Nichele, S., Glover, T. E., and Tufte, G. (2016). Genotype regulation by self-modifying instruction-based development on cellular automata. In *Parallel Problem Solving from Nature–PPSN XIV: 14th International Conference, Edinburgh, UK, September 17-21, 2016, Proceedings 14*, pages 14–25. Springer.

Nichele, S., Ose, M. B., Risi, S., and Tufte, G. (2017). Ca-neat: evolved compositional pattern producing networks for cellular automata morphogenesis and replication. *IEEE Transactions on Cognitive and Developmental Systems*, 10(3):687–700.

Pontes-Filho, S., Lind, P. G., and Nichele, S. (2022). Assessing the robustness of critical behavior in stochastic cellular automata. *Physica D: Nonlinear Phenomena*, 441:133507.

Ribeiro, A., Zhu, R., and Kauffman, S. A. (2006). A general modeling strategy for gene regulatory networks with stochastic dynamics. *Journal of computational Biology*, 13(9):1630–1639.

Sánchez-Puig, F., Zapata, O., Pineda, O. K., Iñiguez, G., and Gershenson, C. (2022). Heterogeneity extends criticality. *arXiv preprint arXiv:2208.06439*.

Variengien, A., Nichele, S., Glover, T., and Pontes-Filho, S. (2021). Towards self-organized control: Using neural cellular automata to robustly control a cart-pole agent. *arXiv preprint arXiv:2106.15240*.

Vohradsky, J. (2001). Neural network model of gene expression. *the FASEB journal*, 15(3):846–854.

von Neumann, J. (1966). Theory of self-reproducing automata. *Mathematics of Computation*, 21:745.

Walker, C. C. (1965). *A Study of a Family of Complex Systems–an Approach to the Investigation of Organisms' Behavior*. Ph.D., University of Illinois at Urbana-Champaign, United States – Illinois. ISBN: 9781084587113.

Walker, C. C. (1971). Behavior of a class of complex systems: the effect of system size on properties of terminal cycles. *Journal of Cybernetics*, 1(4):55–67.

Walker, C. C. and Ashby, W. R. (1966). On temporal characteristics of behavior in certain complex systems. *Kybernetik*, 3(2):100–108.

Wolfram, S. (1986). Theory and applications of cellular automata. *World Scientific*.

Wolfram, S. (2018). Tables of cellular automaton properties: 1986. In *Cellular Automata and Complexity*, pages 513–584. CRC Press.

Wong, B. (2011). Color blindness. *nature methods*, 8(6):441.

Wuensche, A. (1992). Basin of attraction fields of disordered cellular automata networks. `http://www.ddlab.com/publications.html`.

Wuensche, A. (1994). The ghost in the machine: Basins of attraction of random boolean networks. *Artificial Life III: SFI Studies in the Sciences of Complexity, vol. ëVII. Addison-Wesley*.

Wuensche, A. (1997). Attractor basins of discrete networks. *Cognitive Science Research Paper*, 461.

Wuensche, A., Lesser, M., and Lesser, M. J. (1992). *Global Dynamics of Cellular Automata: An Atlas of Basin of Attraction Fields of One-Dimensional Cellular Automata*, volume 1. Andrew Wuensche.