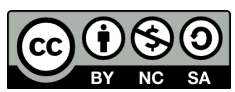
CDM

Mealy Machines

Klaus Sutner

Carnegie Mellon University

Spring 2025



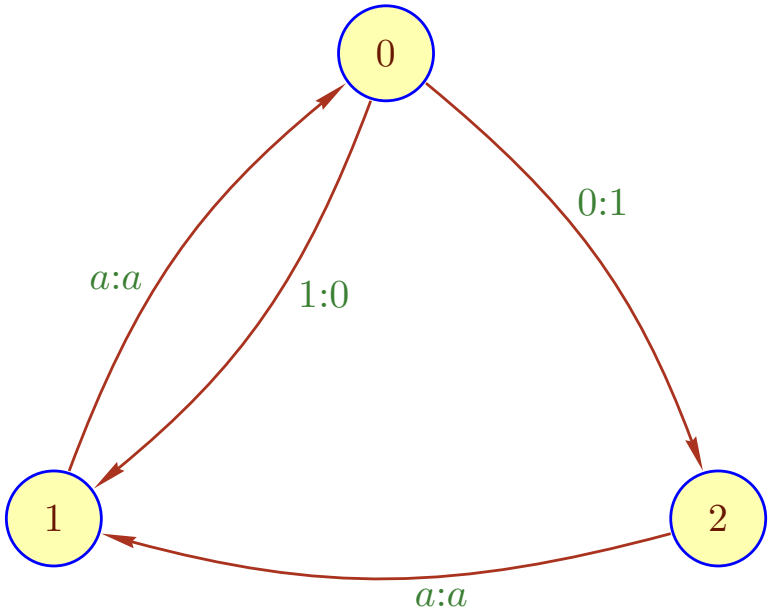
1 **A Semigroup**

2 **Wreath Products**

3 **Knuth Normal Form**

4 **Group Theory**

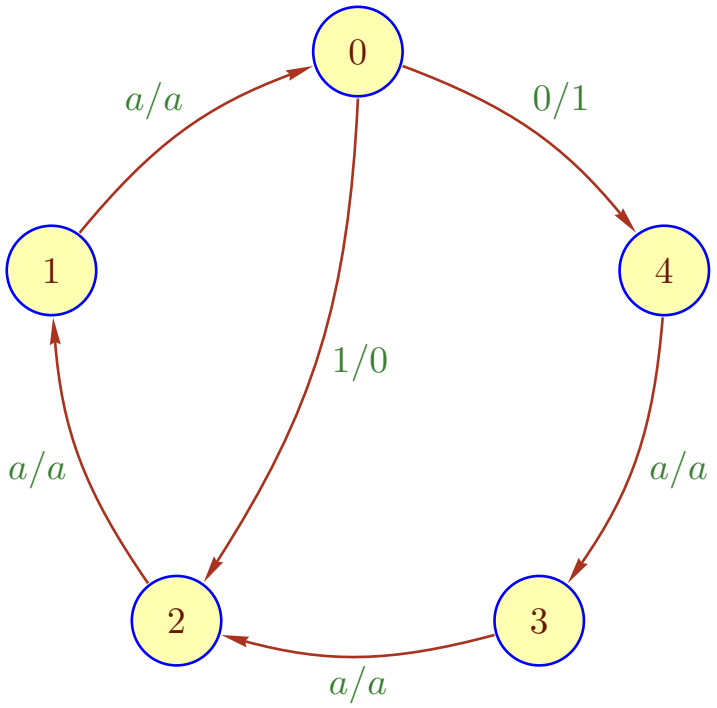
We have an invertible Mealy automaton on 3 states:



We would like to understand the map $\underline{0} : \mathbf{2}^\star \to \mathbf{2}^\star$.

Claim: It is best to study the semigroup $\mathcal{S}$ generated by the maps $\underline{0}$, $\underline{1}$, $\underline{2}$.

We will refer to the last machine as $\mathcal{A}_2^3$.



More generally, $\mathcal{A}_k^n$ is a Mealy machine that is based on a cycle of length n, plus one chord corresponding to a stride of length k.

Since we are dealing with word maps one can use induction to establish some of their properties. Of course, this requires a conjecture first.

Claim: $\mathcal{S}$ is commutative.

$$\underline{0}\,(\underline{1}\,(0x)) = \underline{0}\,(0\,\underline{0}\,(x)) = 1\,\underline{2}\,(\underline{0}\,(x))$$
$$\underline{1}\,(\underline{0}\,(0x)) = \underline{1}\,(1\,\underline{2}\,(x)) = 1\,\underline{0}\,(\underline{2}\,(x))$$

$$\underline{0}\,(\underline{1}\,(1x)) = \underline{0}\,(1\,\underline{0}\,(x)) = 0\,\underline{1}\,(\underline{0}\,(x))$$
$$\underline{1}\,(\underline{0}\,(1x)) = \underline{1}\,(0\,\underline{1}\,(x)) = 0\,\underline{0}\,(\underline{1}\,(x))$$

Simultaneous induction on all pairs of maps.

Since we are dealing with FSMs, we have a whole arsenal of algorithms lying around—and can use them to generate conjectures.

Lemma

Given two Mealy machines $\mathcal{A}$ and $\mathcal{B}$, one can construct a new Mealy machine $\mathcal{A} \times \mathcal{B}$ that determines the composition of maps defined by the two machines.

$$\mathcal{A} : p \xrightarrow{a/b} p' \qquad\qquad \mathcal{B} : q \xrightarrow{b/c} q'$$

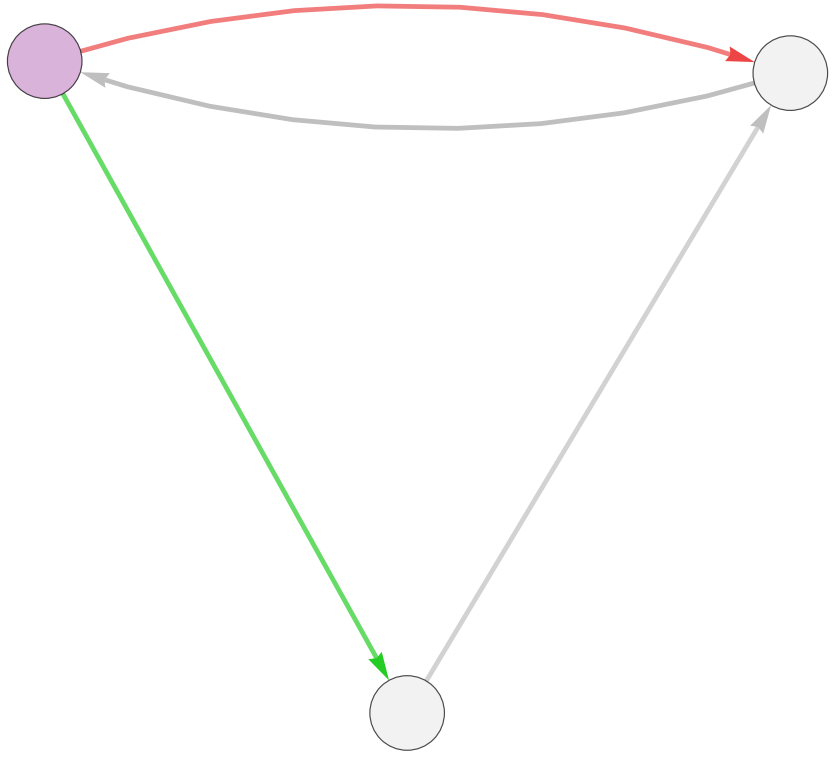
produces a new transition in the product machine

$$\mathcal{A} \times \mathcal{B} : (p, q) \xrightarrow{a/c} (p', q')$$

To keep the pictures manageable, we start with a state (p, q) and only construct the part of the full product automaton that is accessible from there.

So we focus on the composition $\underline{p} \circ \underline{q}$, which seems like the right thing to do to understand $\mathcal{S}$.

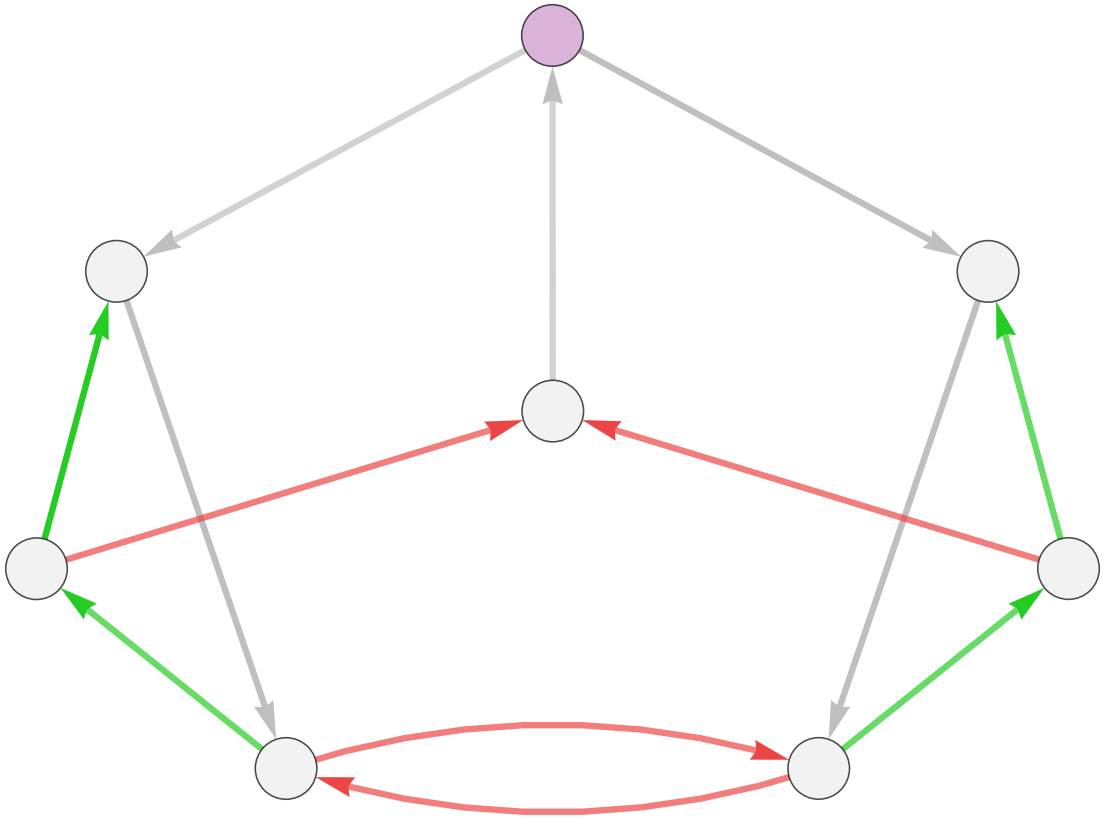
In the following examples, we build Mealy machines for $\underline{0}^{\,k}$.



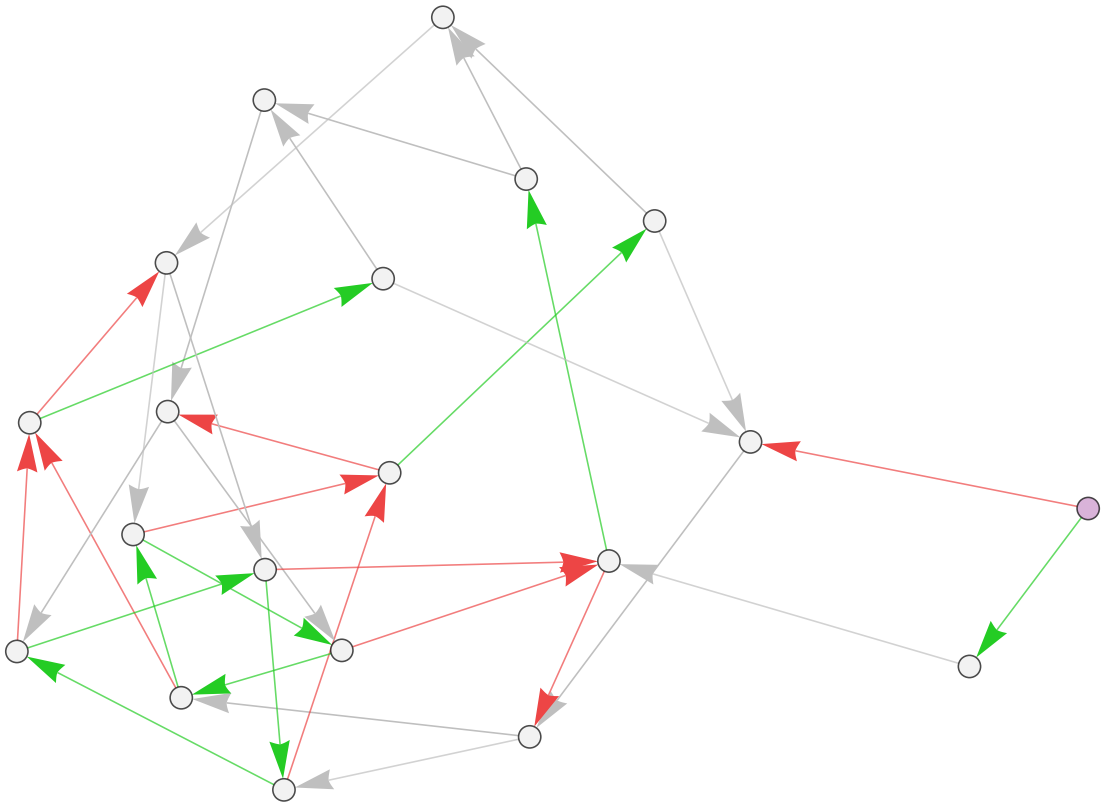
The Mealy machine $\mathcal{A}_2^3$.

All copy states have transitions to one target, indicated by gray edges.

Toggle state transitions are green and red.



The product transducer $\mathcal{A}_2^3 \times \mathcal{A}_2^3$.



The product transducer $\mathcal{A}_2^3 \times \mathcal{A}_2^3 \times \mathcal{A}_2^3$.

The machines are annoyingly complicated. One might wonder whether there is a way to minimize them analogous to ordinary DFAs.

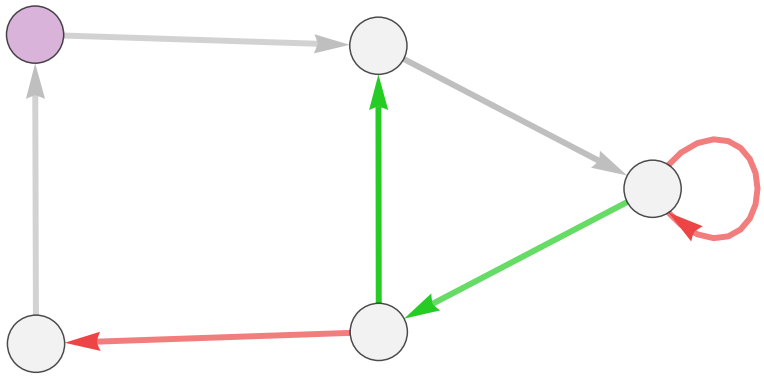
The answer is Yes, and it's actually quite cheap: think of a Mealy machine over alphabet $\mathbf{2}$ as a PDFA over $\mathbf{2} \times \mathbf{2}$.

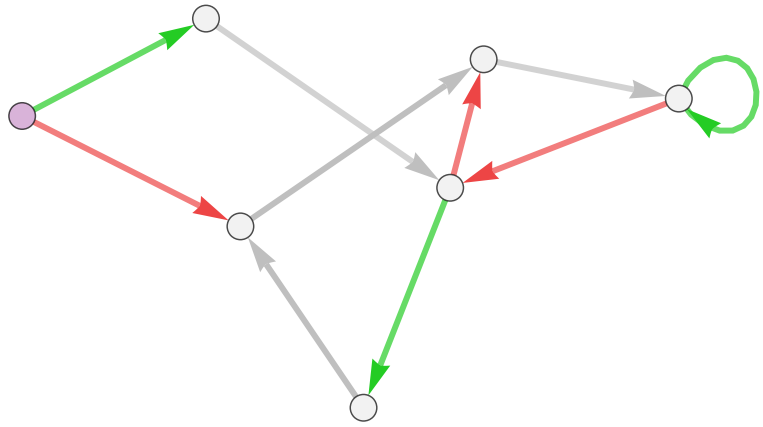
One problem, though: we have no final states to initialize the partition.

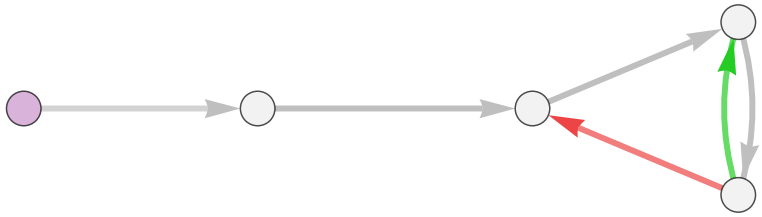
How do we get started?

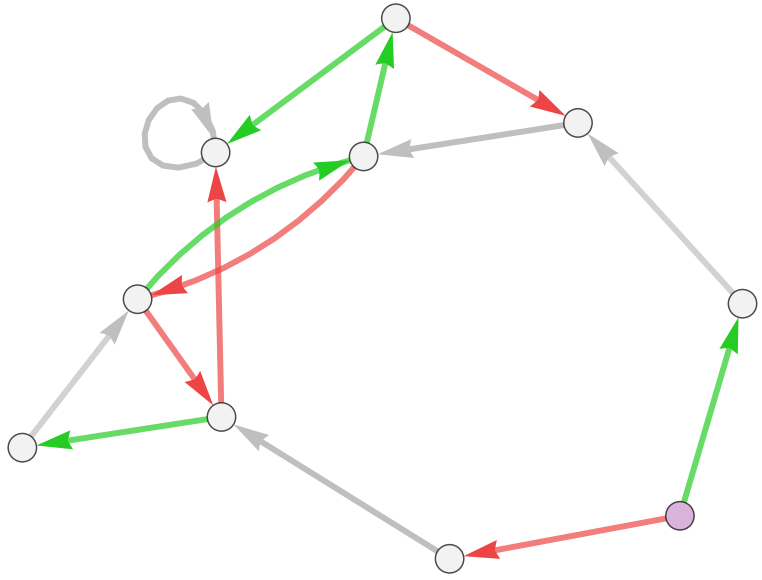
We distinguish between copy and toggle states.

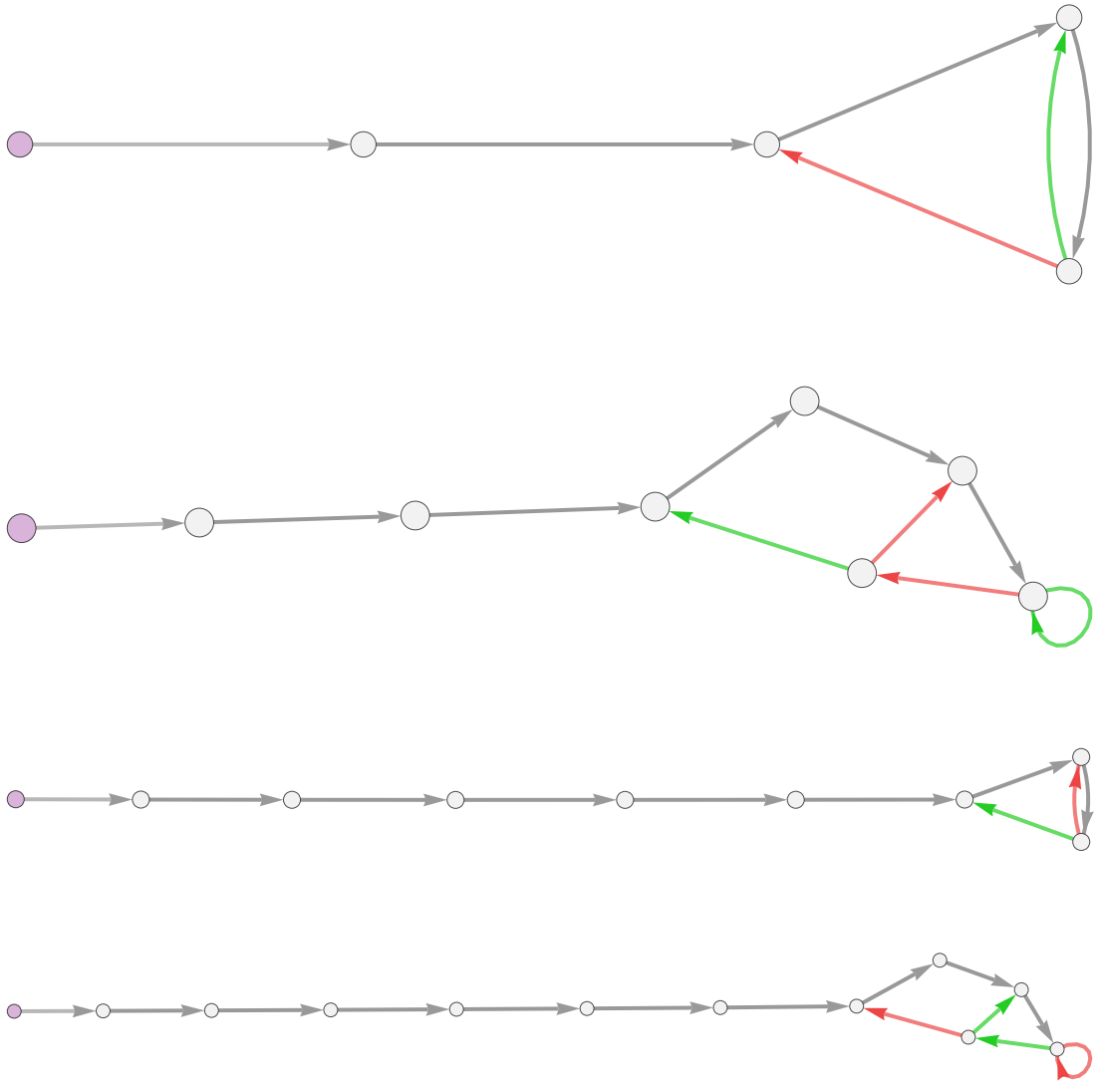
The parity of states is the only thing that matters here.











We can extract some conjectures from staring at the pictures.

Conjecture: The map $\underline{0}^{2^k}$ copies the first $2k$ bits.

More precisely, for k even/odd respectively, we have

$$\underline{0}^{2^k}(uv) = u\,\underline{0}\,(v) \qquad |u| = 2k$$
$$\underline{0}^{2^k}(uv) = u\,\underline{0}^{2}(v) \qquad |u| = 2k-2$$

Note that this explains the apparent periodicity in our first orbit pictures and our observation on cycle lengths (at least partially).

And Proofs?

One can often prove these observations by induction on words.

But we can also use our machine algorithms directly, assuming that the implementation is correct (ideally the algorithms should be formally verified).

E.g., the machine obtained from composing the automata for $\underline{1} \circ \underline{0}$ is the same as for $\underline{0} \circ \underline{1}$ (even without minimization).

Induction is arguably cleaner, but the second approach is attractive since it is easily automated; at least for small identities we can simply crank out the machines.

More Identities?

From our commutativity result we get a monoid epimorphism

$$\Phi : \mathbb{N}^3 \to \mathcal{S} \qquad (a, b, c) \mapsto \underline{0}^{\,a}\,\underline{1}^{\,b}\,\underline{2}^{\,c}$$

Question: Are there any other interesting identities?

To search for identities we can systematically generate minimal machines for all small products (say, $a + b + c \leq 10$) and check if some of them are isomorphic.

Claim: $\mathcal{S}$ is a group.

This follows directly from the identity

$$\underline{0}^{\,2}\underline{1}^{\,2}\underline{2} = I$$

Proof is straightforward by induction:

$$\underline{0}^{\,2}(\underline{1}^{\,2}(\underline{2}\,(ax))) = \underline{0}^{\,2}(a\,\underline{0}^{\,2}(\underline{1}\,(x))) = a\,\underline{1}\,\underline{2}\,(\underline{0}^{\,2}(\underline{1}\,(x))) = x$$

The group is 2-generated and we have an epimorphism

$$\Phi : \mathbb{Z}^2 \to \mathcal{S} \qquad (a, b) \mapsto \underline{0}^{\,a} \underline{1}^{\,b}$$

There might be more identities, but a computer search turns up nothing: if they exist, the corresponding machines are too large to handle. After more fumbling one winds up with

Conjecture: $\mathcal{S}$ is isomorphic to $\mathbb{Z}^2$.

To show that our (semi-)group $\mathcal{S}$ is isomorphic to $\mathbb{Z}^2$ it is a good idea to try to find its "natural habitat."

Claim: $\mathrm{Aut}(\mathbf{2}^\star)$, the group of automorphisms of the binary tree, is the right environment for our semigroup $\mathcal{S}$.

To be clear, $\mathrm{Aut}(\mathbf{2}^\star)$ is a monster of a group, uncountable and hopelessly complicated.

We only care about a small, well-behaved subgroup, but it helps to have the big ambient group in the background.

We think of $\mathbf{2}^\star$ as a rooted, regular, infinite tree $\mathcal{T}$: ε is the root, and node $x \in \mathbf{2}^\star$ has two successors $x0$ and $x1$.

An automorphism of $\mathcal{T}$ is a bijection $f : \mathbf{2}^\star \to \mathbf{2}^\star$ that preserves adjacencies (and in particular length):

$$\begin{aligned} f(\varepsilon) &= \varepsilon \\ f(xa) &= f(x)\, b \qquad \text{some } b \in \mathbf{2} \end{aligned}$$

Since f is a bijection there must be a permutation $\sigma_x \in \mathfrak{S}_2$ such that

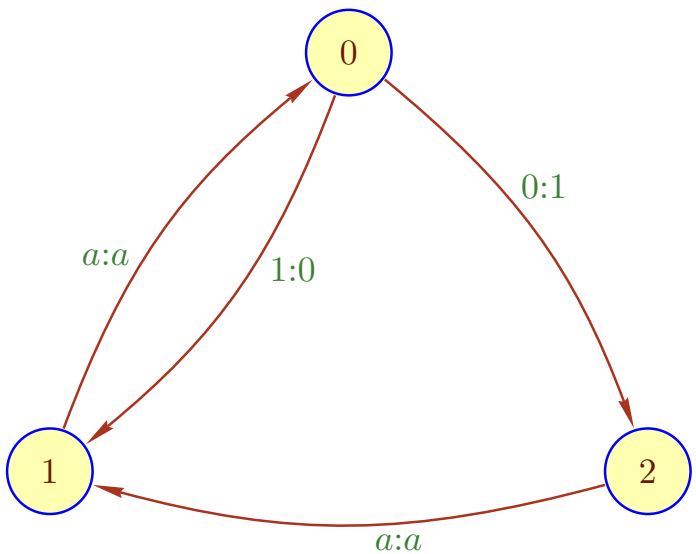
$$f(xa) = f(x)\, \sigma_x(a)$$

We can push the last observation a bit further:

$$f(xy) = f(x)\ \partial_x f(y)$$

where $\partial_x f$ is another automorphism depending on f and x.

The operator ∂ is called residuation.



$$\partial_0\, \underline{0} = \underline{2}$$
$$\partial_1\, \underline{0} = \underline{1}$$
$$\partial_a\, \underline{1} = \underline{0}$$
$$\partial_a\, \underline{2} = \underline{1}$$

From a computational angle, we can describe the action of an automorphism f recursively by specifying 3 data items:

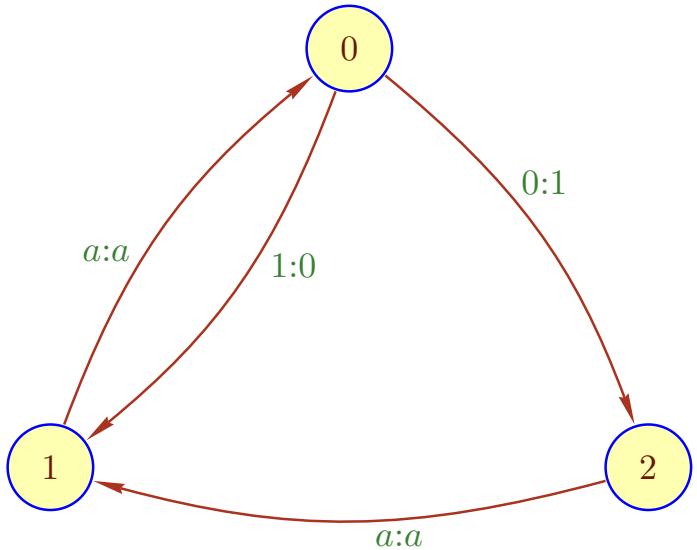
- a parity bit indicating copy/toggle, and
- two automorphisms acting on the subtrees.

One often writes

$$f = (f_0, f_1)s \qquad \text{where } \partial_a f = f_a, s \in \mathfrak{S}_2$$

This is analogous to the transition function of an invertible Mealy automaton.

And residuation is analogous to left quotients for DFA.



$$\underline{0} = (\underline{2}\,,\underline{1}\,)\sigma$$
$$\underline{1} = (\underline{0}\,,\underline{0}\,)$$
$$\underline{2} = (\underline{1}\,,\underline{1}\,)$$

Here σ stands for the transposition $(1,2)$ and the identity is not written.

Can we describe any automorphism f of $\mathcal{T}$ as a Mealy machine?

Yes and no. We have to allow infinitely many states.

Similarly we could build a "DFA" for any language. Finiteness is really critical here.

We have a natural isomorphism

$$\mathsf{Aut}(\mathbf{2}^\star) \cong \mathsf{Aut}(\mathbf{2}^\star) \times \mathsf{Aut}(\mathbf{2}^\star) \times \mathfrak{S}_2$$

The group operation on the left is just composition of functions.

What is the group operation on the right?

Unfortunately, the pointwise approach in a plain Cartesian product does not work:

$$(f_0, f_1, s) \cdot (g_0, g_1, t) = (f_0 g_0, f_1 g_1, st)$$

is plain wrong.

Suppose you have a ring of n lamps; each lamp is either on or off.

There is an eponymous lamplighter, some dude who walks around and turns lights on and off.

More Precisely . . .

The lamplighter can perform two atomic actions:

α	move to the next lamp, or
τ	toggle the state of the current lamp.

The actions are clearly reversible, so there must be a group plus action hiding somewhere.

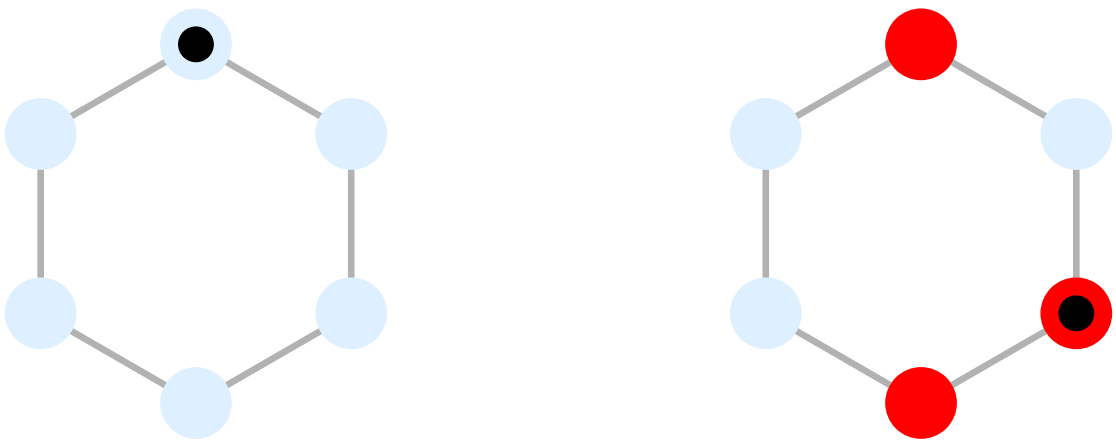
It is obvious that $\alpha^n = 1$ and $\tau^2 = 1$.

The group does not commute, $\alpha\tau \neq \tau\alpha$ (assuming $n > 1$).

But what exactly is the group, and how does it act?

Clearly we can describe the space of configurations as

$$X = \mathbf{2}^n \times \mathbb{Z}_n$$



So we are dealing with bitvectors and modular numbers.

The picture shows the configurations $(000000, 0)$ and $(101100, 2)$.

We need the group G generated by α and τ, something like

$$\langle \alpha, \tau \mid \alpha^n, \tau^2, ??? \rangle$$

but we don't have all the necessary identities for this kind of description.

For example, one can check that

$$(\tau\alpha\tau\alpha^{-1})^2 = 1$$

It seem like a good idea to try to write the group elements in two parts:

- part 1 describes the changes in lights (toggle pattern), and
- part 2 describes the movement of the lamplighter.

Surely, G must be some sort of product.

A first and not unreasonable guess would be the Cartesian product

$$G = \mathbf{2}^n \times \mathbb{Z}_n$$

The first part describes the toggles, the second the lamplighter's displacement.

$(\boldsymbol{a}, s)$ should act on $(\boldsymbol{x}, p)$ as follows:

flip the lights in $\boldsymbol{x}$ according to $\boldsymbol{a}$, change p according to s.

Let $\boldsymbol{e_i} \in \mathbf{2}^n$ be ith unit vector (0-indexed).

Our intended interpretation of the two generators is

$\tau \rightsquigarrow (\boldsymbol{e}_0, 0)$**:** toggle the lamp at the current position.

$\alpha \rightsquigarrow (\mathbf{0}, 1)$**:** the lamplighter moves forward by one.

And $(\boldsymbol{e}_0 \oplus \boldsymbol{e}_1, 3)$ means: toggle the lights in relative positions 0 and 1, then move forward by 3 places, corresponding to $\tau\alpha\tau\alpha^2$.

Does $G = \mathbf{2}^n \times \mathbb{Z}_n$ properly reflect this intuition?

$$(\boldsymbol{a}, r)(\boldsymbol{b}, s) = (\boldsymbol{a} \oplus \boldsymbol{b}, r + s)$$

where $\oplus$ stands for bit-wise xor and addition is modulo n.

Consider the group element $\tau\alpha\tau\alpha$.

According to our attempt, we have

$$(\boldsymbol{e}_0, 1)(\boldsymbol{e}_0, 1) = (\boldsymbol{e}_0 \oplus \boldsymbol{e}_0, 2) = (\mathbf{0}, 2)$$

Our model thinks that $\tau\alpha\tau\alpha = \alpha^2$, which sadly is false: the position is right, but not the toggle pattern.

What we need instead is

$$(\boldsymbol{e}_0, 1)(\boldsymbol{e}_0, 1) = (\boldsymbol{e}_0 \oplus \boldsymbol{e}_1, 2)$$

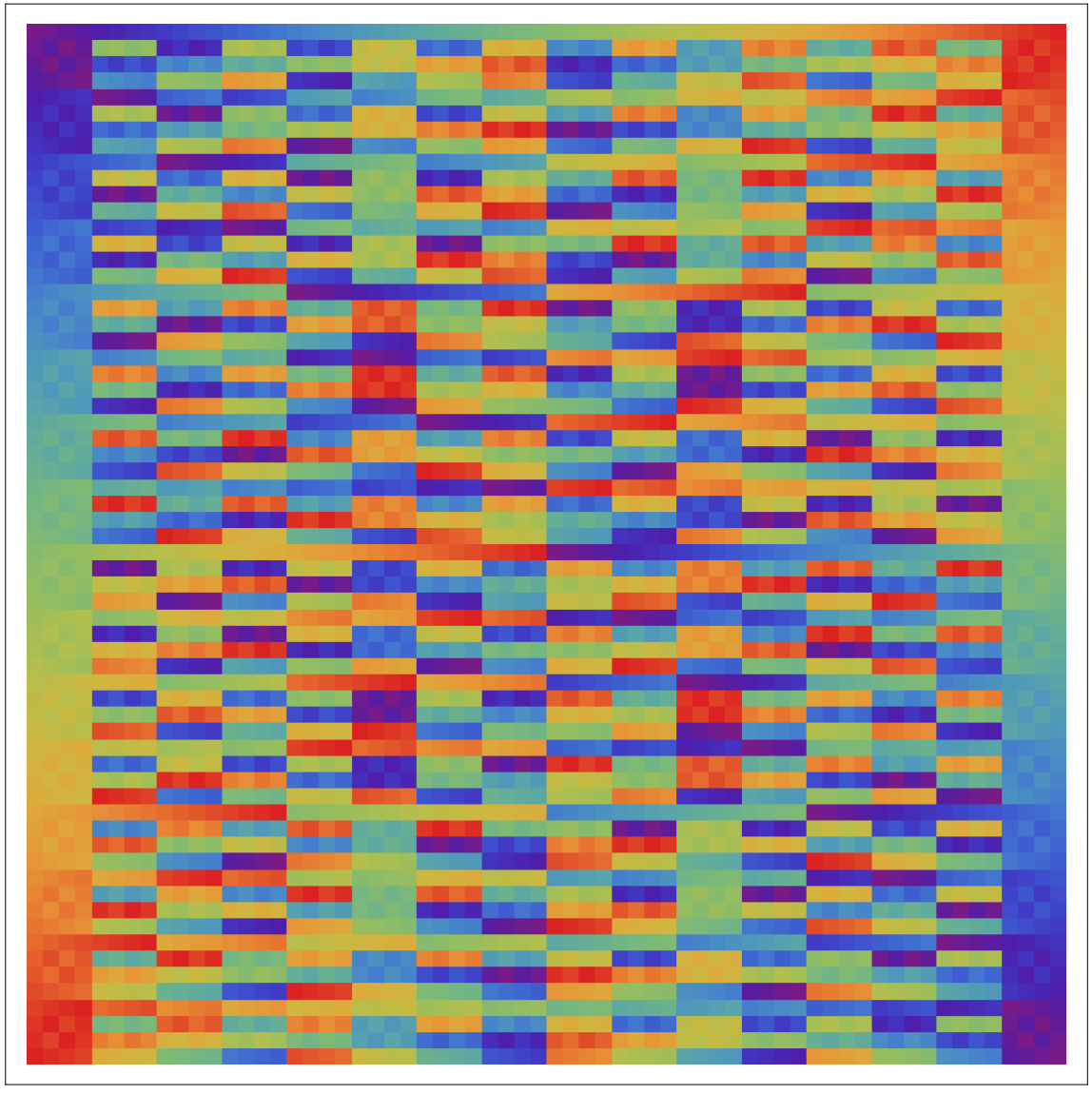
We need to take into account the position of the lamplighter: when the lamplighter moves, the next switch-vector has to be adjusted accordingly.

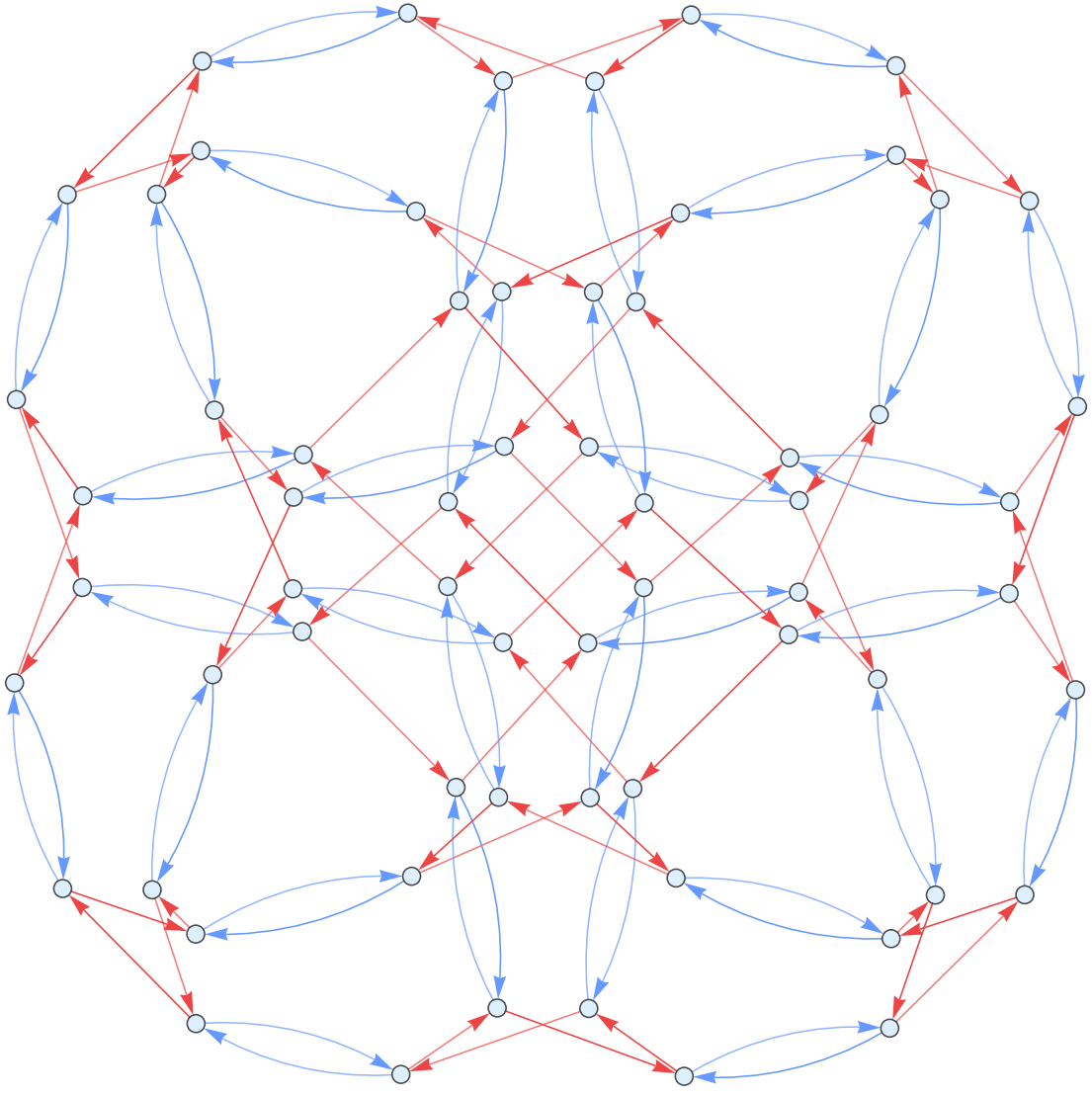
More technically, a modular number s acts on $\boldsymbol{a}$ by rotating the sequence by s places.

To fix our group, keep the carrier set, but adjust the group operation to

$$(\boldsymbol{a}, r)(\boldsymbol{b}, s) = (\boldsymbol{a} \oplus \mathsf{rot}^r(\boldsymbol{b}), r + s)$$

where rot is a cyclic rotation.





The actual lamplighter group in the literature is based on a biinfinite row of lamps, with only finitely many lamps lit.

So the configurations are

$$X = \left(\bigoplus_{\mathbb{Z}} \mathbf{2}\right) \times \mathbb{Z}$$

The identities here are

$$\tau^2 = 1 \qquad (\alpha^i \tau \alpha^{-i})(\alpha^j \tau \alpha^{-j}) = (\alpha^j \tau \alpha^{-j})(\alpha^i \tau \alpha^{-i})$$

The group is finitely generated, but not finitely presentable.

Exercise

Verify $(\alpha\tau^i\alpha\tau^{-i})^2 = 1$.

Exercise

Interpret $(\alpha\tau^i\alpha\tau^{-i})^2 = 1$ *geometrically.*

Exercise

What would happen if we were to interpret $(\boldsymbol{a}, r)$ *as "move first by* r*, then toggle according to* $\boldsymbol{a}$*"?*

Here is a more systematic look at the last construction.

Definition

Let $\varphi : B \to \mathrm{Aut}(A)$ be a group homomorphism. The semidirect product $A \rtimes_\varphi B$ is defined on the Cartesian product of the carrier sets of A and B by

$$(a_1, b_1)(a_2, b_2) = \big(a_1\, \varphi(b_1)(a_2), b_1\, b_2\big)$$

For φ the constant map $b \mapsto I$, we get the ordinary direct product.

It is not even clear that this operation is associative.

$$\begin{aligned}((a_1, b_1)(a_2, b_2))(a_3, b_3) &= (a_1\, \varphi(b_1)(a_2), b_1 b_2)(a_3, b_3) \\ &= (a_1\, \varphi(b_1)(a_2)\, \varphi(b_1 b_2)(a_3), b_1 b_2 b_3)\end{aligned}$$

$$\begin{aligned}(a_1, b_1)((a_2, b_2)(a_3, b_3)) &= (a_1, b_1)(a_2\, \varphi(b_2)(a_3), b_2 b_3) \\ &= (a_1\, \varphi(b_1)(a_2\, \varphi(b_2)(a_3)), b_1 b_2 b_3) \\ &= (a_1\, \varphi(b_1)(a_2)\, \varphi(b_1 b_2)(a_3), b_1 b_2 b_3)\end{aligned}$$

The neutral element is $(1_A, 1_B)$ and the inverse is

$$(a, b)^{-1} = (\varphi(b^{-1})(a^{-1}), b^{-1})$$

In the dihedral group D_n, rotation α has order n, and reflection β has order 2, but D_n is not isomorphic to the Cartesian product $\mathbb{Z}_n \times \mathbb{Z}_2$. To fix this, we define $\varphi : B \to \mathrm{Aut}(A)$

$$\varphi(0)(x) = x \qquad \varphi(1)(x) = x^{-1}$$

φ duly is a group homomorphism since $\mathbb{Z}_n$ is commutative.

Then the dihedral group D_n is isomorphic to $A \rtimes_\varphi B$: rotation corresponds to $(1, 0)$ and reflection corresponds to $(0, 1)$.

$$\begin{aligned}
\beta\alpha &= (0,1)(1,0) \\
&= \big(0 + \varphi(1)(1), 1 + 0\big) \\
&= (n-1, 1) \\
&= \alpha^{-1}\beta
\end{aligned}$$

We need one more type of product, a special kind of semidirect product.

The first group is a Cartesian product A^n, A a group and $n \geq 2$.

Suppose $B \subseteq \mathfrak{S}_n$ is some permutation group and define $\varphi : B \to \mathrm{Aut}(A^n)$ by

$$\varphi(b)(\boldsymbol{a}) = (a_{b^{-1}(1)}, \ldots, a_{b^{-1}(n)})$$

So $\varphi(b)$ simply permutes the elements of vector $\boldsymbol{a} \in A^n$.

Definition

The wreath product $A \wr B$ of A and B is the semidirect product $A^n \rtimes_\varphi B$.

We are only interested in $n = 2$ and $B = \mathfrak{S}_2$.

So our wreath products look like

$$A \wr \mathfrak{S}_2 = \Big(A \times A\Big) \rtimes \mathfrak{S}_2$$

For convenience, we will use 0-indexing, so the elements are $\Big((a_0, a_1), s\Big)$ and the group operation is

$$\Big((a_0, a_1), s\Big)\,\Big((b_0, b_1), t\Big) = \Big((a_0 b_{s(0)}, a_1 b_{s(1)}), s\,t\Big)$$

This is exactly the way we wrote our tree automorphisms: two residuals plus one parity bit.

$$\mathsf{Aut}(\mathbf{2}^\star) \simeq \mathsf{Aut}(\mathbf{2}^\star) \wr \mathfrak{S}_2 = (\mathsf{Aut}(\mathbf{2}^\star) \times \mathsf{Aut}(\mathbf{2}^\star)) \rtimes \mathfrak{S}_2$$

$\mathsf{Aut}(\mathbf{2}^\star)$ is a perfect example of a wreath product.

This may sound like much ado about nothing, but it actually helps quite a bit.

E.g., we can establish commutativity by induction in the group directly, without the detour of induction on words (which is really an argument about the group action).

$$
\begin{aligned}
\underline{0}\ \underline{1} &= (\underline{2}\,,\underline{1}\,)\sigma\,(\underline{0}\,,\underline{0}\,) \\
&= (\underline{2}\,\underline{0}\,,\underline{1}\,\underline{0}\,) \\
&= (\underline{0}\,\underline{2}\,,\underline{0}\,\underline{1}\,) \\
&= (\underline{0}\,,\underline{0}\,)(\underline{2}\,,\underline{1}\,)\sigma \\
&= \underline{1}\ \underline{0}
\end{aligned}
$$

A wreath induction proof for the group identity, using commutativity:

$$\begin{aligned}
\underline{0}^{\,2}\,\underline{1}^{\,2}\,\underline{2} &= \big((\underline{2},\underline{1})\sigma\big)^2(\underline{0},\underline{0})^2(\underline{1},\underline{1}) \\
&= (\underline{2}\,\underline{1},\underline{1}\,\underline{2})(\underline{0},\underline{0})(\underline{0},\underline{0})(\underline{1},\underline{1}) \\
&= (\underline{2}\,\underline{1}\,\underline{0}\,\underline{0}\,\underline{1},\underline{1}\,\underline{2}\,\underline{0}\,\underline{0}\,\underline{1}) \\
&= \big(\underline{0}^{\,2}\,\underline{1}^{\,2}\,\underline{2},\underline{0}^{\,2}\,\underline{1}^{\,2}\,\underline{2}\big)
\end{aligned}$$

Again, this is a bit more elegant than the approach using induction on words.

Recall NP-equivalence of Boolean functions: one can negate and/or permuted the inputs.

Negation is handled by the Boolean group $\mathbf{2}^k$ and permutation by the symmetric group $\mathfrak{S}_k$. A Cartesian product does not work, essentially for the same reasons as in the lamplighter/dihedral scenario.

Instead one needs a wreath product $\mathbf{2} \wr \mathfrak{S}_k$.

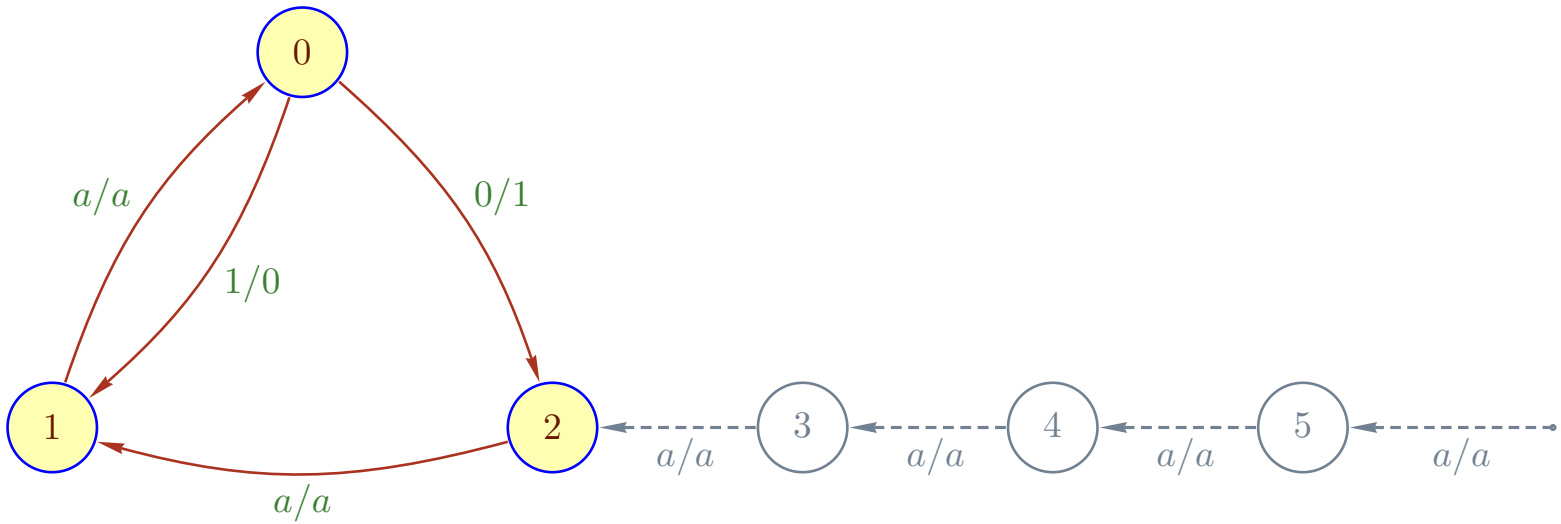
And one needs to figure out how to calculate the cycle index polynomial.

Our 3-state machine $\mathcal{A}_2^3$ produces

$$\underline{0} = (\underline{2}\,,\underline{1}\,)\,\sigma \qquad \underline{1} = (\underline{0}\,,\underline{0}\,) \qquad \underline{2} = (\underline{1}\,,\underline{1}\,)$$

Insane Idea:
How about adding a state 3 with $\underline{3} = (\underline{2}\,,\underline{2}\,)$?
And infinitely more, 4, 5, $\ldots$, n, $\ldots$

So we add an infinite chain of copy states, anchored at the the old state 2.
This produces a new, infinite machine $\mathcal{A}_2^{3\infty}$.


0
1
2
3
4
5
a/a
1/0
0/1
a/a
a/a
a/a
a/a
a/a

First we have to make sure the new machine does not produce any new group elements, $\mathsf{Grp}(\mathcal{A}_2^3) = \mathsf{Grp}(\mathcal{A}_2^{3\infty})$.

This is handled by establishing a new identity:

$$\underline{k}^{\,2} = \underline{k{+}2}\ \underline{k{+}3}$$

To see why, note that both sides are even, and residuation produces $\underline{k{-}1}^{\,2} = \underline{k{+}1}\ \underline{k{+}2}$ for $k > 1$. For $k = 0$ we get

$$\underline{0}^{\,2} = (\underline{2}\,,\underline{1}\,)\sigma\,(\underline{2}\,,\underline{1}\,)\sigma = (\underline{1}\,\underline{2}\,,\underline{1}\,\underline{2}\,) = (\underline{1}\,,\underline{1}\,)(\underline{2}\,,\underline{2}\,) = \underline{2}\,\underline{3}$$

Done by wreath induction.

A similar argument shows that

$$\underline{k}^{\,2}\ \underline{k{+}1}^{\,2}\ \underline{k{+}2} = I$$

The new identities give us a rewrite system for automorphisms in $\mathcal{S}$:

cancellation identity

$$\underline{k}^{\,2}\ \underline{k{+}1}^{\,2}\ \underline{k{+}2} \rightsquigarrow I$$

shift identity

$$\underline{k}^{\,2} \rightsquigarrow \underline{k{+}2}\ \underline{k{+}3}$$

So cancellation removes terms and shift flattens them out, at the cost of pushing things to the right. The hope is that by using both we can write all group elements in a particulary nice form.

Normal Form Theorem

Theorem (Knuth)

Every automorphism f of $\mathcal{S}$ has a unique normal form

$$f = \underline{k_1}\ \underline{k_2}\ \ldots\ \underline{k_n}$$

where $k_i < k_{i+1}$, $n \geq 0$.

Corollary

$\mathcal{S}$ is isomorphic to $\mathbb{Z}^2$.

Unfortunately, our rewrite system is not quite terminating.

We must use cancellation before shift, otherwise we wind up with

$$\underline{0}^{\,2}\,\underline{1}^{\,2}\,\underline{2} \rightsquigarrow \underline{2}^{\,2}\,\underline{3}^{\,2}\,\underline{4} \rightsquigarrow \underline{4}^{\,2}\,\underline{5}^{\,2}\,\underline{6} \rightsquigarrow \ldots$$

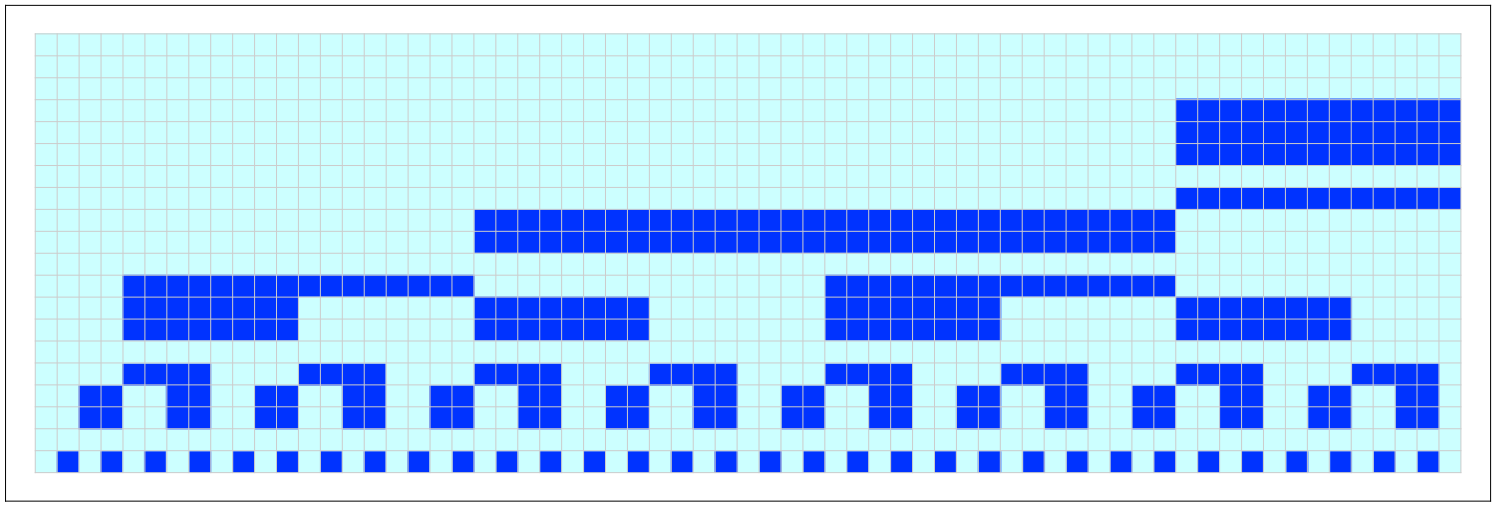
It is a character building exercise to show that the system terminates with a unique result given this "cancellation-first" proviso.

KNF for $\underline{0}^{20}\underline{1}^{20}$

0	1	2	3	4	5	6	7	8	9
20	20								
0	20	10	10						
0	10	0	5						
0	0	0	10	5					
0	0	0	0	5	5	5			
0	0	0	0	1	1	3			
0	0	0	0	1	1	1	0	1	1

$$\mathsf{KNF}(\underline{0}^{20}\underline{1}^{20}) = \underline{4}\ \underline{5}\ \underline{6}\ \underline{8}\ \underline{9}$$

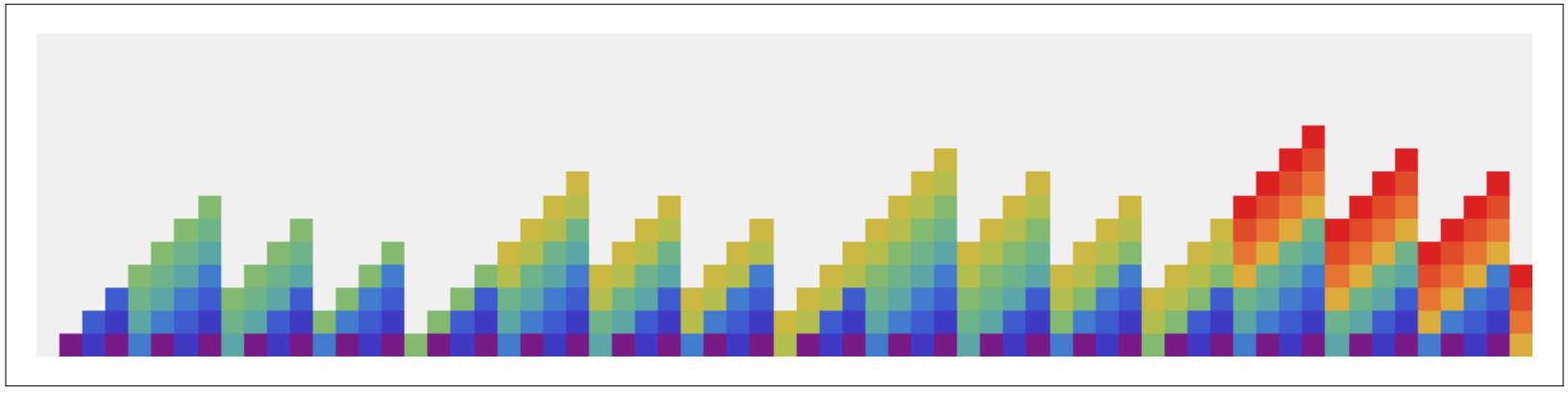
KNF for $\underline{0}^i$



KNFs for $\underline{0}^i$, $0 \leq i \leq 64$.

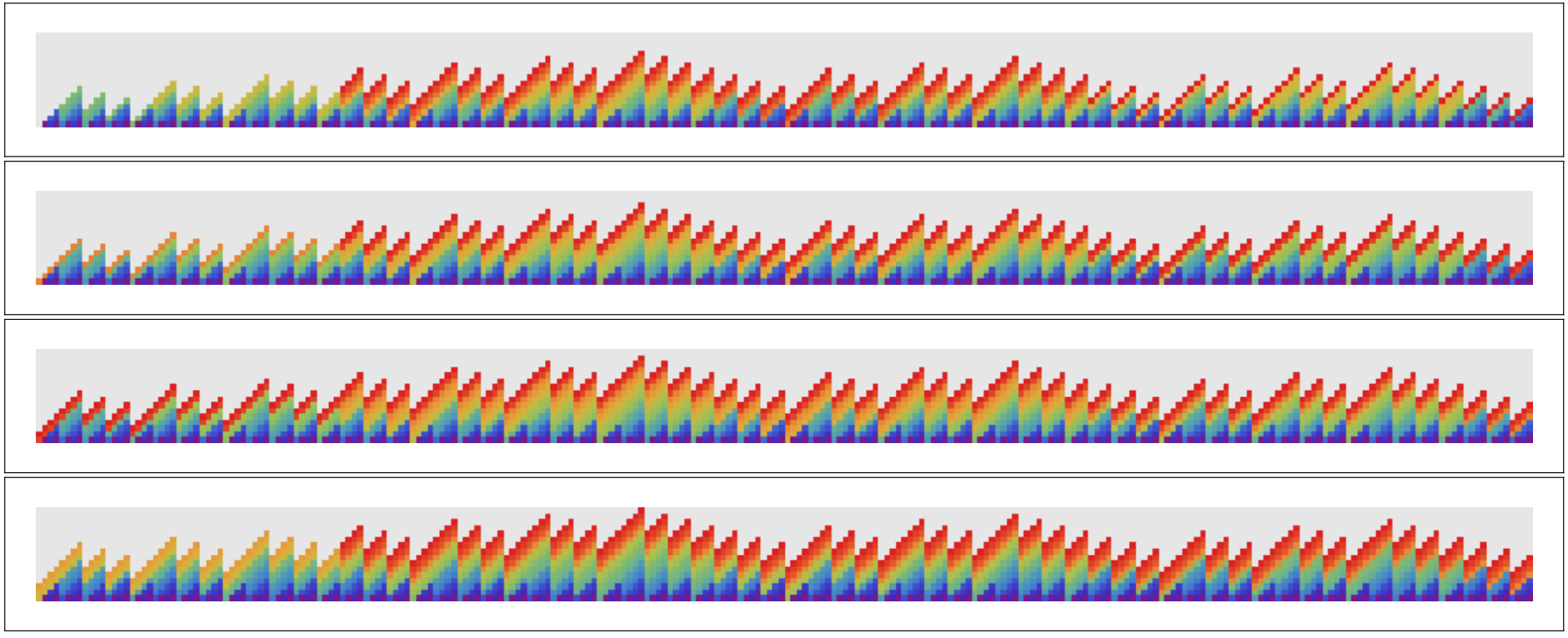
Each column represents the normal form as a bitvector.

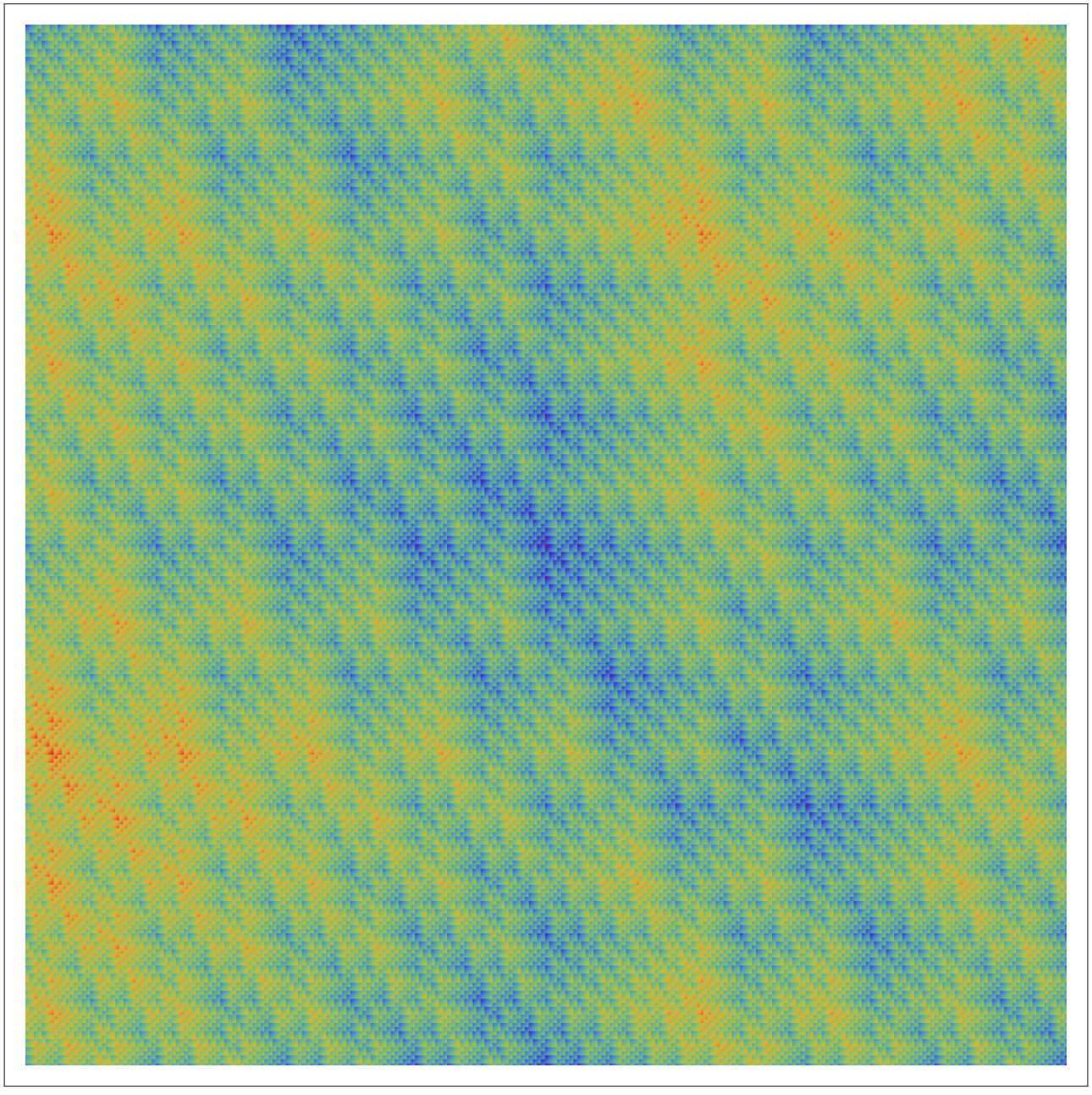
KNF for $\underline{0}^i$

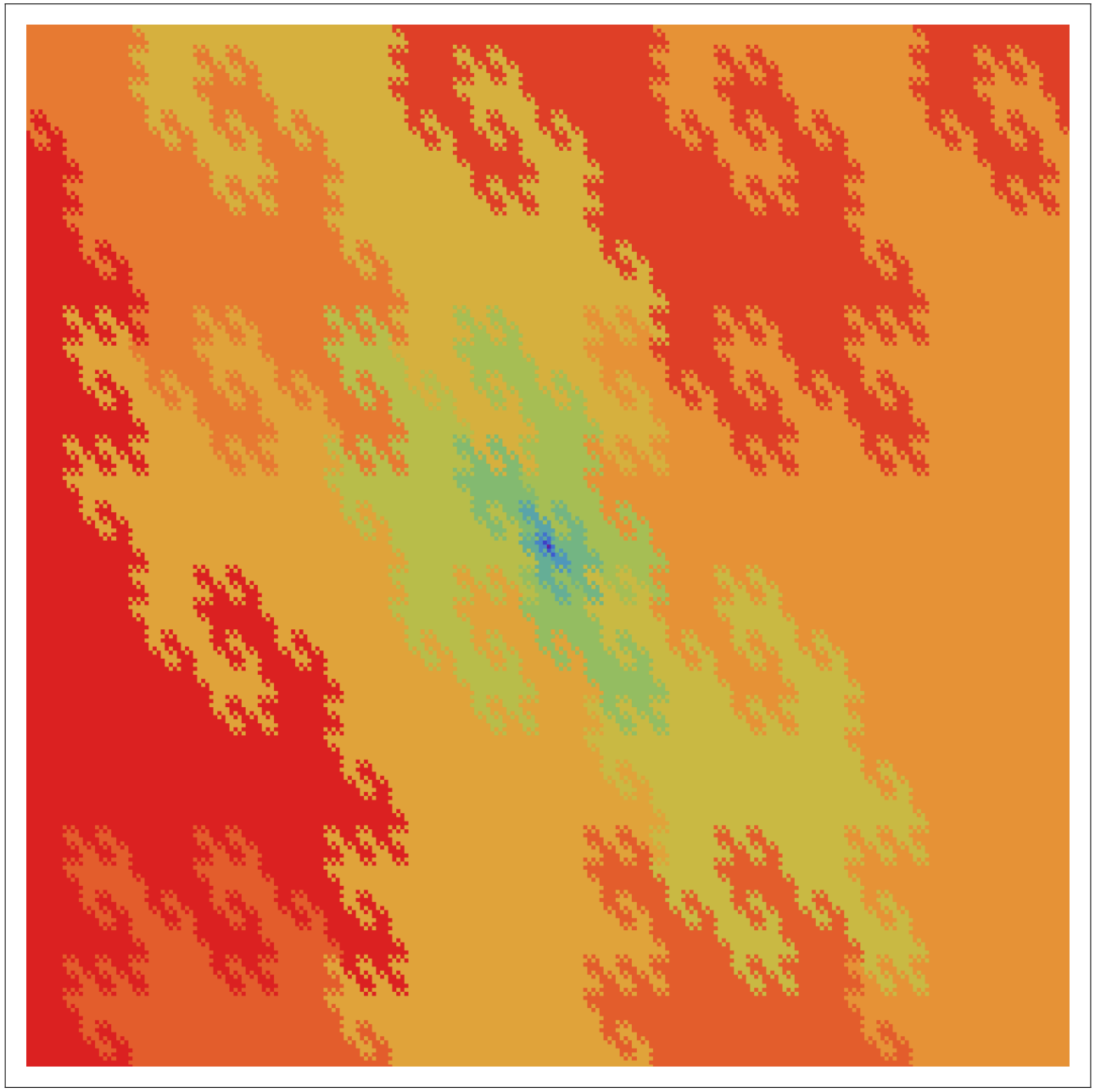


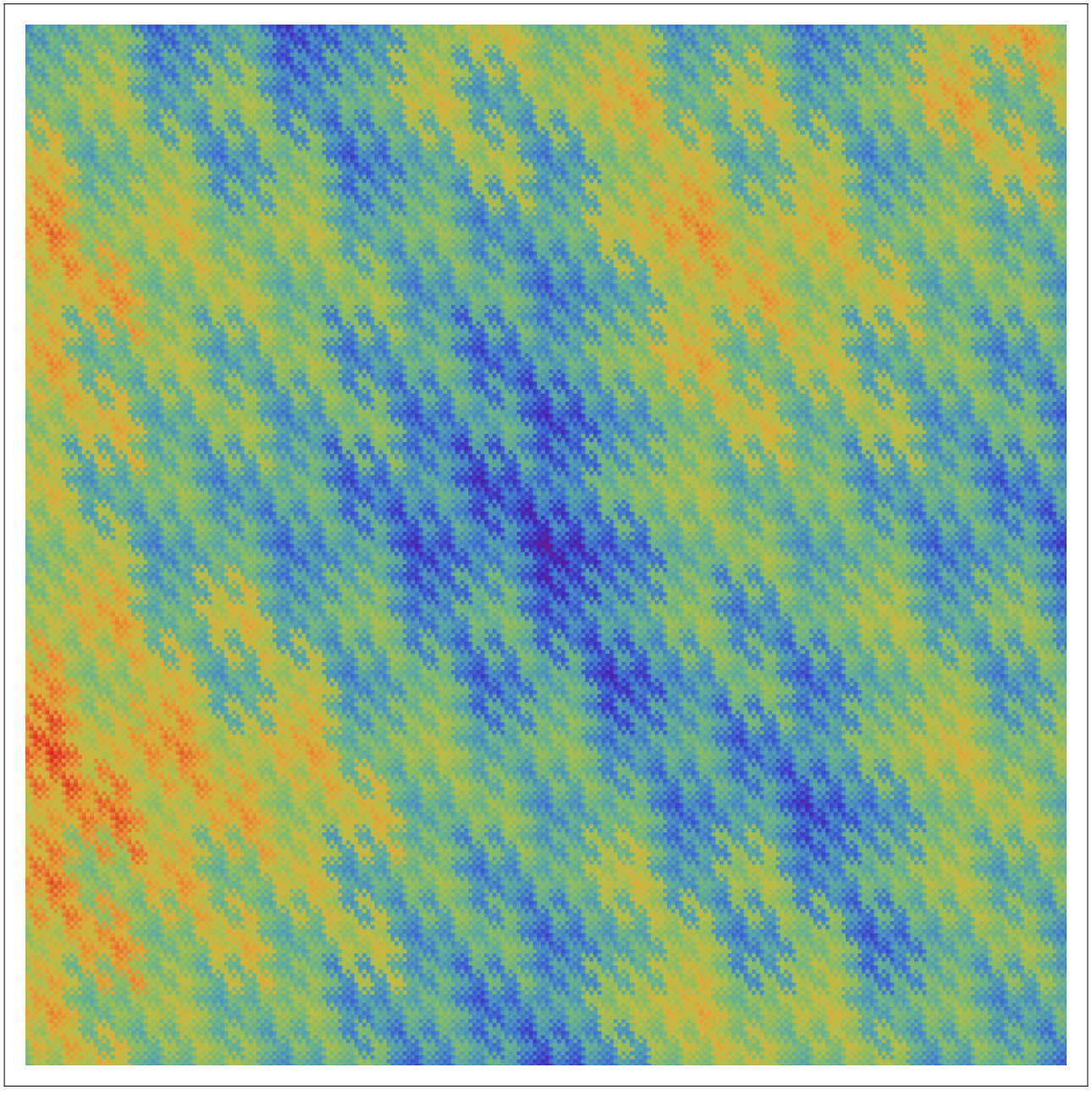
Again, KNFs for $\underline{0}^i$, $0 \leq i \leq 64$.

The coefficients k_u are color coded.









Since $\mathcal{S} \cong \mathbb{Z} \times \mathbb{Z}$, it is natural to ask whether there is some particularly simple isomorphism.

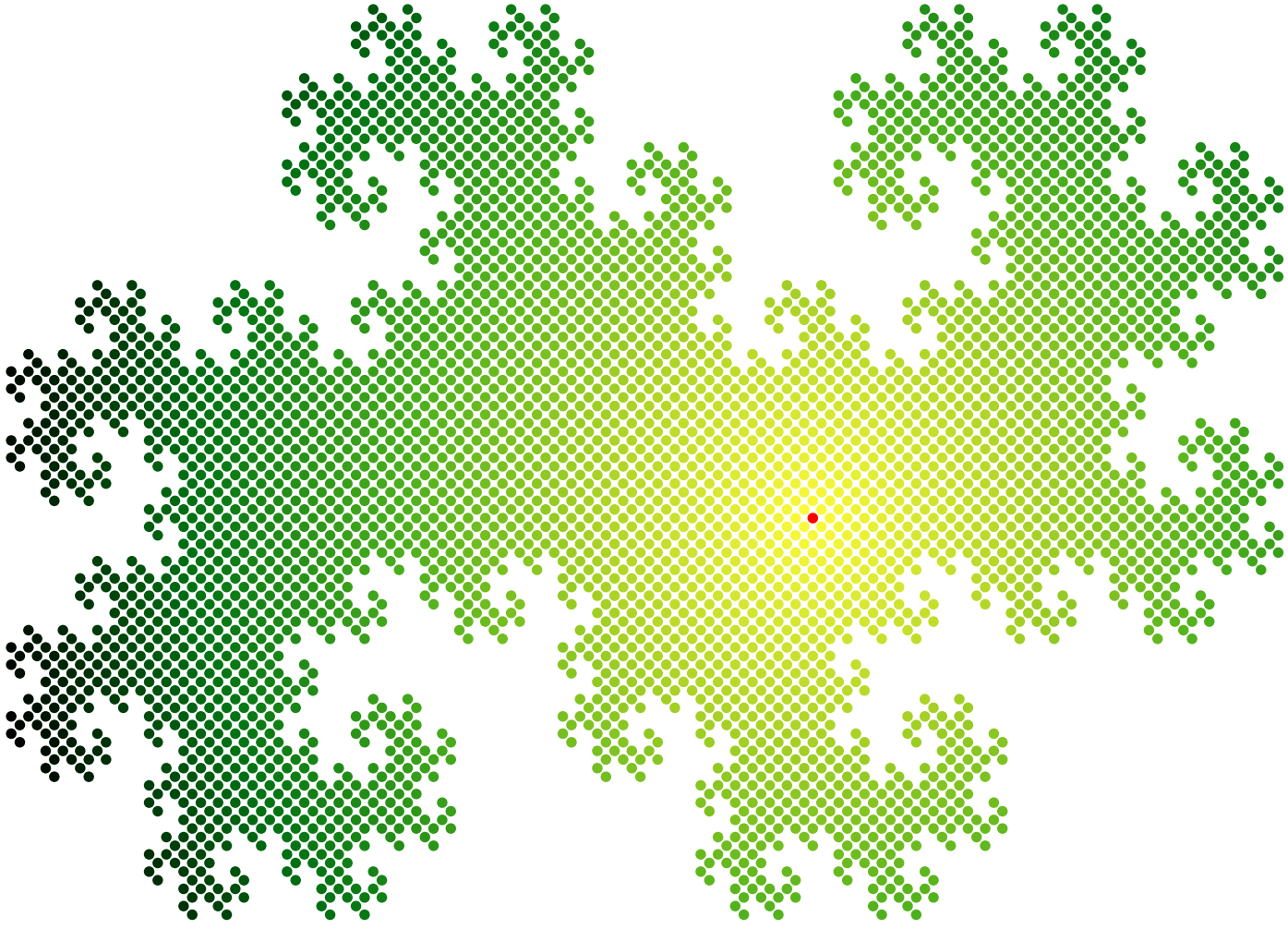
A constructive way of describing an isomorphism is to translate Knuth normal form into a Gaussian integer: Let $\rho = i - 1 \in \mathbb{C}$ and define

$$\varphi(f) = \rho^{k_1} + \rho^{k_2} + \ldots + \rho^{k_t}$$

where $f = \underline{k_1}\ \underline{k_2}\ \ldots\ \underline{k_n}$ is the normal form. Here we assume $\varphi(\text{nil}) = 0$.

Theorem

The map $\varphi : \mathcal{S} \to \mathbb{Z}[\mathbf{i}]$ is an isomorphism.



Call a group torsion or periodic iff all its elements have finite order.

Burnside Problem (1902)

Suppose G is finitely generated torsion group.
Is G necessarily finite?

Some positive results were obtained initially, but the problem remained open for a good long time.

Theorem (Golod, Shafarevich 1964)

The Burnside Problem fails in general.

In the bounded version we strengthen periodicity to: for some fixed n, $x^n = 1$ for all $x \in G$.

Theorem (Novikov, Adian 1968)

The bounded Burnside Problem fails for all odd $n > 4381$.

The rather complicated proof uses the Prouhet-Thue-Morse sequence.

Growth of a Group

Suppose we have a finitely generated group G. How many distinct group elements can be reached in the Cayley graph of G starting at 1 on a path of length at most n?

More precisely, use words over the symmetric alphabet $\Sigma \cup \overline{\Sigma}$ to name group elements. Write $\|x\|$ for the shortes word that denotes $x \in G$ and define the growth function by

$$\gamma(n) = |\{\, x \in G \mid \|x\| \leq n \,\}|$$

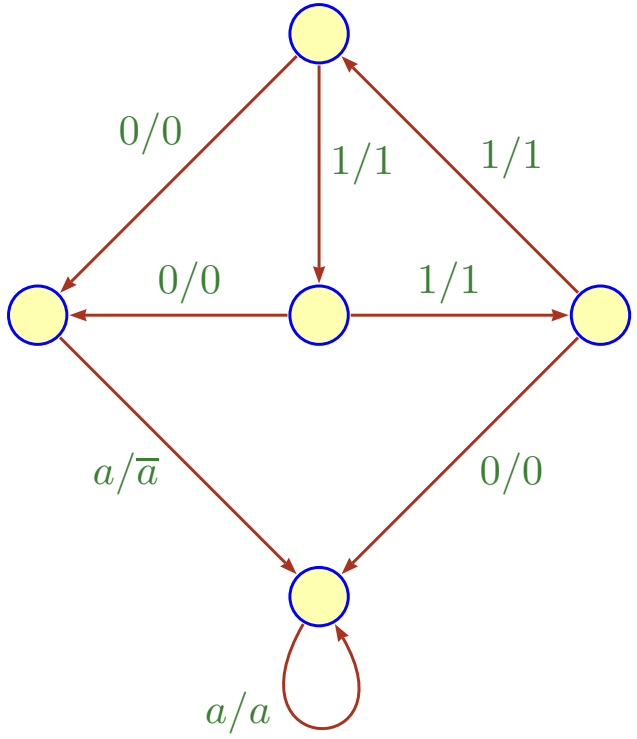
It is easy to find groups of exponential growth (free groups) or polynomial growth ($\mathbb{Z}^n$).

Milnor's Problem (1960)

Are there finitely generated groups of intermediate growth?

E.g., growth like $\gamma(n) \sim 2^{\sqrt{n}}$ is intermediate.

Such groups exist, but are rather difficult to construct.



An invertible Mealy automaton on 5 states, with just a single toggle state, the machine changes at most one bit in any input string.

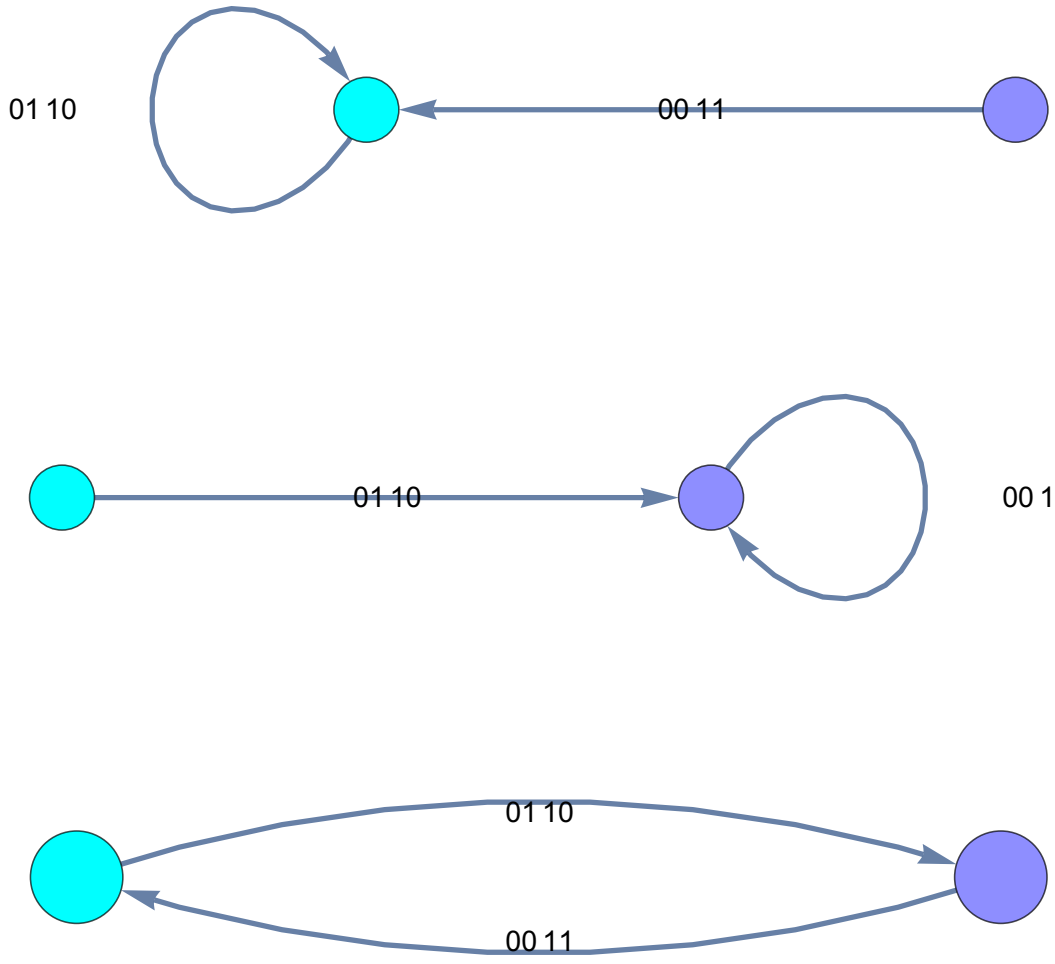
The group has intermediate growth and also provides a counterexample for Burnside's problem.

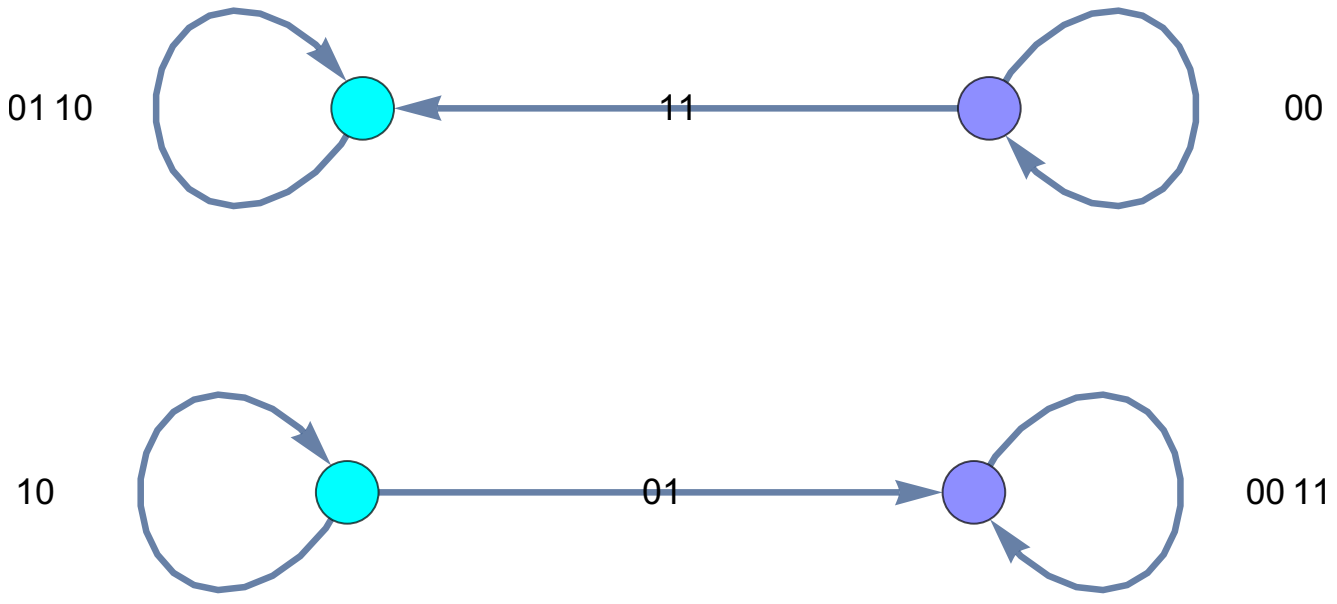
Easy (?) Challenge: Find all groups generated by invertible Mealy machines with just 2 states.

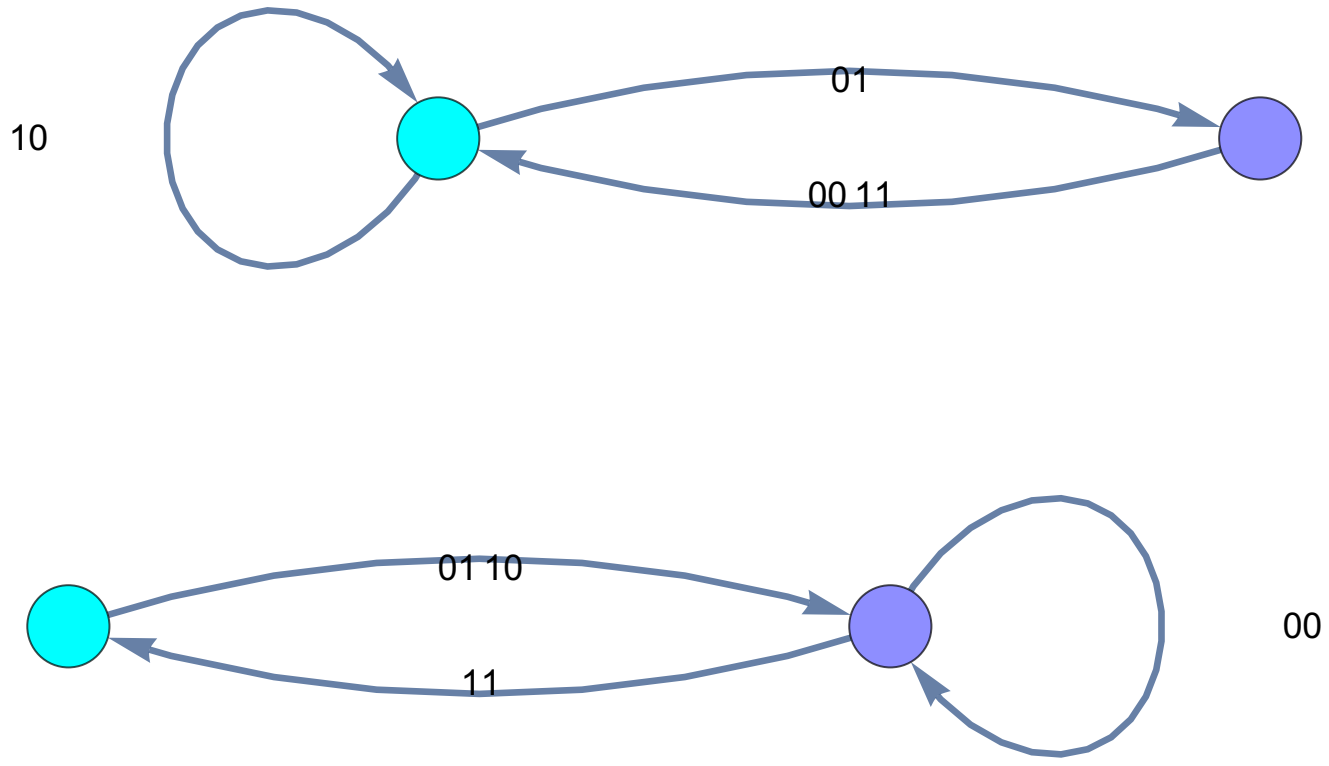
To exclude boring cases, we may make the following assumptions:

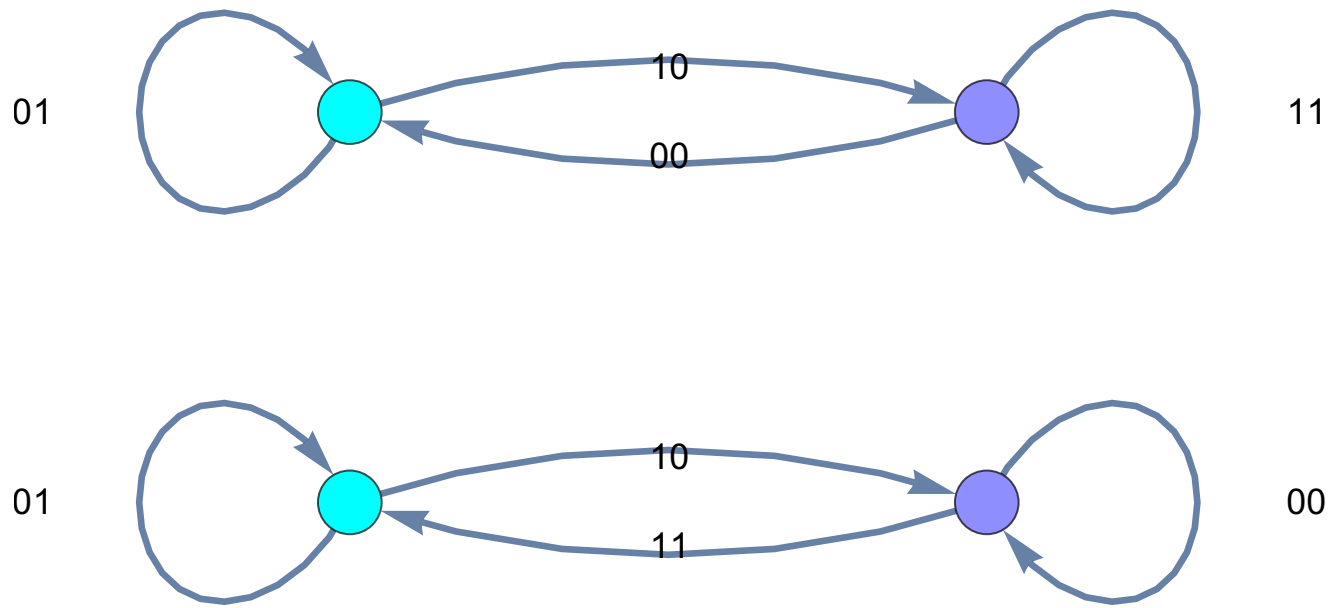
- State 1 is toggle, state 2 is copy.
- There is at least one transition between the states.
- We can interchange symbols 0 and 1.

Here is what's left.

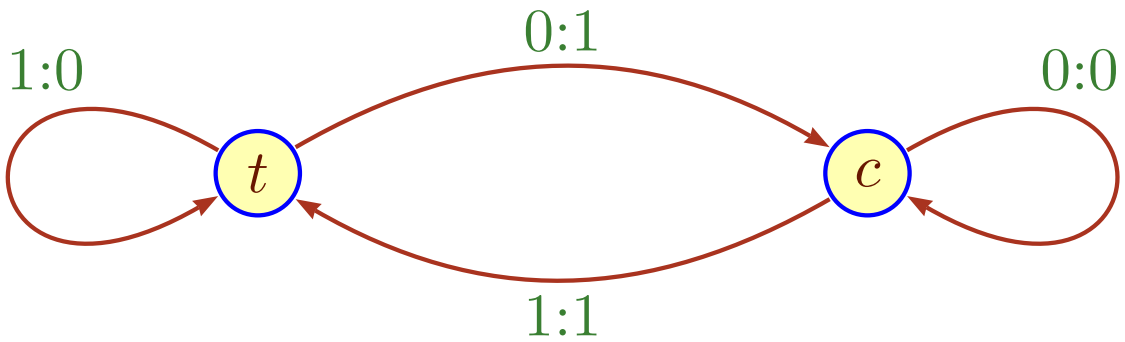

01 10
00 11
01 10
00 11
01 10
00 11


01 10
11
00
10
01
00 11


10
01
00 11
01 10
11
00


01
10
00
11
01
10
11
00

Here is a slightly nicer diagram:



In wreath notation this whole transducer comes down to this:

$$t = (c, t)\sigma \qquad\qquad c = (c, t)$$

This is very similar to the exotic counter from above where $c = (t, t)$. It seems like a reasonable guess that the group generated by this machine is pretty simple . . .

Big Surprise:

This machine generates the lamplighter group.

This is the real, infinite Lamplighter group, not the finite versions we talked about. It has lots of interesting properties and has been studied extensively; Google Scholar shows 22,000 hits.

It is quite amazing that a group of this complexity can be described in terms of a 2-state machine.

On classification of groups generated by 3-state automata over a 2-letter alphabet

I. Bondarenko, R. Grigorchuk, R. Kravchenko, Y. Muntyan, V. Nekrashevych, D. Savchuk and Z. Šunić

Algebra and Discrete Mathematics, 1 (2008) 1–163

There are 5832 automata, though many of them produce the same group.

Theorem

There are at most 122 distinct groups generated by 3-state automata.