IOP PUBLISHING NONLINEARITY

Nonlinearity **22** (2009) 421–436 doi:10.1088/0951-7715/22/2/010

Cycle equivalence of graph dynamical systems

Matthew Macauley[1] **and Henning S Mortveit**[2]

[1] Department of Mathematical Sciences, Clemson University, Clemson, SC 29634, USA
[2] Department of Mathematics and Virginia Bioinformatics Institute, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, USA

Received 18 April 2008, in final form 26 November 2008
Published 20 January 2009
Online at stacks.iop.org/Non/22/421

Recommended by L-S Young

Abstract

Graph dynamical systems (GDSs) generalize concepts such as cellular automata and Boolean networks and can describe a wide range of distributed, nonlinear phenomena. Two GDSs are *cycle equivalent* if their periodic orbits are isomorphic as directed graphs, which captures the notion of having comparable long-term dynamics. In this paper, we study *cycle equivalence* of GDSs in which the vertex functions are applied sequentially through an update sequence. The main result is a general characterization of cycle equivalence based on the underlying graph Y and the update sequences. We construct and analyse two graphs $C(Y)$ and $D(Y)$ whose connected components contain update sequences that induce cycle equivalent dynamical system maps. The number of components in these graphs, denoted $\kappa(Y)$ and $\delta(Y)$, bound the number of possible long-term behaviour that can be generated by varying the update sequence. We give a recursion relation for $\kappa(Y)$ which in turn allows us to enumerate $\delta(Y)$. The components of $C(Y)$ and $D(Y)$ characterize dynamical neutrality, their sizes represent structural stability of periodic orbits and the number of components can be viewed as a system complexity measure. We conclude with a computational result demonstrating the impact on complexity that results when passing from radius-1 to radius-2 rules in asynchronous cellular automata.

Mathematics Subject Classification: 37B99, 93D99, 20F55

1. Introduction

This paper is concerned with dynamical systems on networks. Specifically, we consider graph dynamical systems (GDSs) which are constructed from (i) a finite undirected graph Y where each vertex has a state from a set K, (ii) a sequence of *vertex functions* and (iii) an *update scheme* [6, 12]. The vertex function f_v of vertex v is used to map the state of v at time t to time $t + 1$. That is, f_v maps $y_v(t)$ to $y_v(t + 1)$, and the arguments to the function are the states of vertex v and its neighbouring vertices in Y. We refer to this as a *vertex update*. The update scheme governs how the collection of vertices are updated in a time step. For example, a synchronous update scheme will apply all the vertex functions simultaneously. One complete execution of the update scheme is called a *system update*, or GDS *map*, and it takes the system state from time t to $t + 1$.

In this paper we will consider *sequential dynamical systems* (SDSs). This is a class of GDSs where the update scheme is governed by a word $w = (w_1, \ldots, w_k)$ over the vertex set of Y. A system update for an SDS is conducted by applying the vertex functions in sequence as specified by w: first one applies f_{w_1}, then f_{w_2} and so on up to f_{w_k}. The dynamical system map, or SDS *map*, is essentially the composition of the vertex functions in the sequence specified by w. Clearly, different update schemes may give rise to different GDS maps. We remark that an *asynchronous cellular automaton* is a special case of an SDS.

GDSs represent a useful framework for capturing distributed, dynamical phenomena with local interactions. Application examples include disease dynamics on social contact graphs [8], packet transport in wireless networks, traffic systems with individual cars [14] and dynamics of biological systems [1]. Additionally, many computational algorithms correspond precisely to SDSs. For example, the algorithms of [9, 15] on gene annotation and transport computations on irregular grids amount to fixed point computations of SDSs. It is also interesting to note that the concept of GDSs generalizes classical cellular automata as well as many similar constructions.

A general theme in GDS research is the deduction of global dynamics and phase space properties based on the structure of the graph, the vertex functions and the nature of the update scheme. This is as opposed to exhaustive computations which quickly become intractable for realistic systems. Our paper is about *cycle equivalence* of finite GDSs, and we study this in the context of SDSs. Since the state space is finite, the phase space may be represented as a directed graph. Two GDSs are cycle equivalent if their periodic orbits are isomorphic as directed graphs. In other words, cycle equivalence captures the fact that dynamical systems have the same long-term dynamics.

In this paper we will show how properties of the graph and the update sequence of SDSs affect the periodic orbit structures. As a particular example, we show the surprising result that if the graph Y is a tree then there is only one possible periodic orbit configuration and that this holds for any fixed choice of vertex functions. Additionally, we give a graph measure κ that bounds the number of possible periodic orbit configurations for general graphs. In light of this, κ may serve as a graph–theoretical complexity measure for these systems. We also describe how one can construct a complete set of update sequence representatives to generate all possible orbit configurations.

Paper organization. In section 2, we describe SDS related terminology and relevant background results from [12, 13, 16] on functional and dynamical equivalence of SDSs. In section 3, we prove one of the main results of this paper: SDSs over a common graph that have the same vertex functions and whose update sequences differ by a cyclic shift are cycle equivalent. Additionally, when the vertex states are taken from $K = \{0, 1\}$, which is the standard choice in most studies of cellular automata, then reflections of the update sequence also result in cycle equivalent SDSs. In section 4, we introduce the graphs $C(Y)$ and $D(Y)$ which form the basis for our analysis and characterization of cycle equivalence over general graphs. These graphs are examples of neutral networks, and we characterize some of their structural properties. The components of these graphs correspond to certain equivalence classes of acyclic orientations of Y, and we show how shifts and reflections of update sequences have a natural interpretation in terms of source-to-sink operations on the acyclic orientations. Based on this correspondence, we study the functions κ and δ which count the connected components of $C(Y)$ and $D(Y)$, respectively. Here we show how $\delta(Y)$ is derived from $\kappa(Y)$ and give several results for the computation of $\kappa(Y)$ along with their implications for dynamics.

As explained earlier, κ and δ can be regarded as measures for system complexity, and as a computational example, we demonstrate how $\kappa(Y)$ increases from $\Theta(n)$ for radius-1 elementary cellular automaton rules to $\Theta(n \cdot 2^n)$ upon passing to radius-2 rules. We also show how the presence of symmetries in the graph may allow for improved complexity bounds.

In the final section, we indicate how cycle equivalence of SDSs is closely related to Coxeter theory. Some of the results that we prove in this paper have a natural analogue when translated into the language of Coxeter groups, a field rich in algebra, combinatorics and geometry. This opens the door to using the mathematical tools and results from Coxeter theory to study SDSs, something that has never been done before.

2. Background and definitions

Let Y be a finite undirected graph with vertex set $\mathrm{v}[Y] = \{1, \dots, n\}$ and edge set $\mathrm{e}[Y]$. Since most graphs in this paper are finite and undirected we simply refer to this class of graphs as 'graphs' and specify otherwise if needed. The 1-*neighbourhood* of vertex v in Y is the set $B_1(v; Y) = \{v' \in \mathrm{v}[Y] \mid \{v, v'\} \in \mathrm{e}[Y]\} \cup \{v\}$ and the *ordered* 1-*neighbourhood* $n[v]$ of v is the sequence of vertices from $B_1(v; Y)$, in increasing order. The *degree* of vertex v is written $d(v)$. Each vertex v is assigned a state $y_v \in K$ where K is a finite set. In the following, y_v is called a *vertex state* and the n-tuple $y = (y_1, \dots, y_n)$ is a *system state*[3]. We write

$$y[v] = (y_{n[v](1)}, \dots, y_{n[v](d(v)+1)}) \tag{2.1}$$

for the restriction of the system state to the vertices in $n[v]$ and let $y'[v]$ denote the same tuple but with the vertex state y_v omitted. The finite field with $q = p^k$ elements is denoted $\mathbb{F}_q$.

Let $\mathfrak{f}_Y := (f_i)_{i \in \mathrm{v}[Y]}$ be a sequence of *vertex functions* $f_i\colon K^{d(i)+1} \longrightarrow K$ and define the sequence of Y-*local functions* $\mathfrak{F}_Y := (F_i)_{i \in \mathrm{v}[Y]}$ with $F_i\colon K^n \longrightarrow K^n$ by

$$F_i(y_1, \cdots, y_n) = (y_1, \dots, y_{i-1}, f_i(y[i]), y_{i+1}, \dots, y_n). \tag{2.2}$$

It is clear that $\mathfrak{f}_Y$ completely determines $\mathfrak{F}_Y$ and vice versa. However, there are settings when it is easier to speak of one rather than the other.

Let W_Y denote the set of words over $\mathrm{v}[Y]$[4]. Words are written as $w = (w_1, w_2, \dots, w_m)$, $w = w_1 w_2 \cdots w_m$, $w = (w(1), w(2), \dots, w(m))$, etc. The subset of W_Y where each element of $\mathrm{v}[Y]$ occurs exactly once is denoted S_Y, and thus the elements of S_Y may be thought of as permutations of $\mathrm{v}[Y]$. The symmetric group S_n acts on system states by

$$\gamma \cdot (y_1, \dots, y_n) = (y_{\gamma^{-1}(1)}, \dots, y_{\gamma^{-1}(n)}). \tag{2.3}$$

Definition 2.1 (Sequential dynamical system). *A sequential dynamical system (*SDS*) is a triple* $(Y, \mathfrak{F}_Y, w)$ *where* Y *is a graph,* $\mathfrak{F}_Y = (F_i)_{i \in \mathrm{v}[Y]}$ *is a sequence of* Y*-local functions and* $w = (w_1, \dots, w_m) \in W_Y$. *The associated* SDS *map* $[\mathfrak{F}_Y, w]\colon K^n \longrightarrow K^n$ *is the function composition*

$$[\mathfrak{F}_Y, w] = F_{w_m} \circ F_{w_{m-1}} \circ \cdots \circ F_{w_2} \circ F_{w_1}. \tag{2.4}$$

The graph Y of an SDS is called the *base graph*, and when $w \in S_Y$, the SDS is referred to as a *permutation* SDS. A sequence of Y-local functions $\mathfrak{F}_Y$ is $\mathsf{Aut}(Y)$-*invariant* if $\gamma \circ F_v = F_{\gamma(v)} \circ \gamma$ for all $v \in \mathrm{v}[Y]$ and all $\gamma \in \mathsf{Aut}(Y)$. Here, the composition of a function $K^n \to K^n$ with a permutation of K is interpreted as in (2.3). The corresponding sequence of vertex functions

[3] In the context of, for example, cellular automata a system state is frequently called a *configuration*.

[4] Also referred to as the Kleene star or Kleene closure of $\mathrm{v}[Y]$.

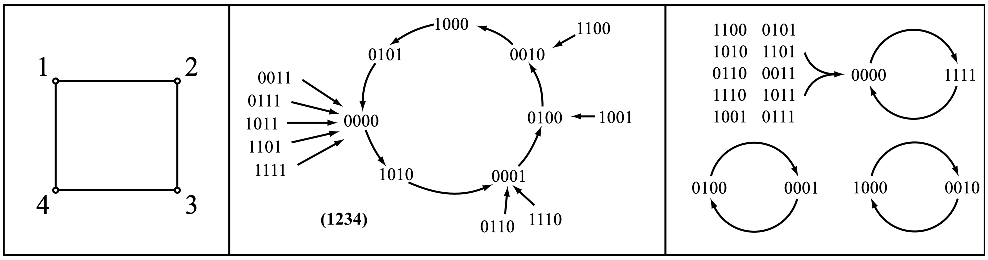


Figure 1. The graph Circ_4 (left), the phase space with the update sequence $(1, 2, 3, 4)$ (middle) and the phase space with a synchronous update scheme (right) of example 2.2.

$\mathfrak{f}_Y$ is $\mathsf{Aut}(Y)$-invariant if $\mathfrak{F}_Y$ is $\mathsf{Aut}(Y)$-invariant. The *phase space* of the map $\phi\colon K^n \longrightarrow K^n$ is the directed graph $\Gamma(\phi)$ with vertex set K^n and edge set $\big\{(y, \phi(y)) \mid y \in K^n\big\}$. The following example illustrates these concepts.

Example 2.2 (Asynchronous elementary cellular automaton rule ECA #1). *Let* $Y = \mathsf{Circ}_4$, *the graph with vertex set* $\{1, 2, 3, 4\}$ *and edges* $\{i, i+1\}$ *modulo* 4 *as shown on the left in figure 1 and with binary vertex states. In this setting we have system state* $y = (y_1, y_2, y_3, y_4)$ *and ordered neighbourhood* $n[1] = (1, 2, 4)$ *with restricted system state* $y[1] = (y_1, y_2, y_4)$. *If we use the Boolean* nor*-function* $\mathsf{nor}_3\colon \mathbb{F}_2^3 \longrightarrow \mathbb{F}_2$ *(i.e. ECA #1) given by* $\mathsf{nor}_3(x, y, z) = (1 + x)(1 + y)(1 + z)$ *to induce the vertex functions we get, for example,* $F_1(y) = (\mathsf{nor}_3(y[1]), y_2, y_3, y_4)$. *With the update sequence* $\pi = (1, 2, 3, 4)$ *we get the* SDS *map*

$$[\mathsf{Nor}_Y, \pi] = F_4 \circ F_3 \circ F_2 \circ F_1. \tag{2.5}$$

It is easily verified that $[\mathsf{Nor}_Y, \pi](0, 0, 0, 0) = (1, 0, 1, 0)$. *In contrast, for a synchronous update scheme, the system state* $(0, 0, 0, 0)$ *would have been mapped to* $(1, 1, 1, 1)$. *The entire phase space of the* SDS *map* $[\mathsf{Nor}_Y, \pi]$ *in (2.5) is displayed in figure 1 (middle) along with the phase space for a synchronous update scheme (right).*

What follows is a short overview of functional and dynamical equivalence of SDSs. The analysis is largely concerned with the aspect of update sequences and characterizes SDS maps for a fixed graph Y and fixed Y-local functions $\mathfrak{F}_Y$ in terms of w. These equivalence notions provide the foundation for cycle equivalence.

2.1. Functional equivalence

Two SDSs are *functionally equivalent* if their SDS maps are identical as functions. For a fixed sequence $\mathfrak{F}_Y$, a natural question to ask is when is $[\mathfrak{F}_Y, w] = [\mathfrak{F}_Y, w']$ for $w, w' \in W_Y$? The *update graph* $\hat{U}(Y)$ provides an answer to this. The update graph of Y has a vertex set W_Y. Two length-m words $w \neq w'$ are adjacent if (i) they differ *only* by a transposition of entries k and $k+1$ and (ii) $\{w_k, w_{k+1}\} \notin \mathrm{e}[Y]$. The finite subgraph $U(Y)$ of $\hat{U}(Y)$ induced by the vertex set S_Y is called the *permutation update graph* and is denoted $U(Y)$. Clearly, it is a union of connected components of $\hat{U}(Y)$. Both $U(Y)$ and $\hat{U}(Y)$ are examples of neutral networks as mentioned in the introduction. The update graph $U(\mathsf{Circ}_4)$ is shown figure 3.

Let $\sim_\alpha$ be the equivalence relation on S_Y defined by $\pi \sim_\alpha \pi'$ iff π and π' belong to the same connected component in $U(Y)$. We denote an equivalence class by $[\pi]_\alpha$ and the set of equivalence classes by $S_Y/\!\sim_\alpha$, i.e.

$$S_Y/\!\sim_\alpha = \{[\pi]_\alpha \mid \pi \in S_Y\}.$$

By construction, we have the implication

$$\pi \sim_\alpha \sigma \implies [\mathfrak{F}_Y, \pi] = [\mathfrak{F}_Y, \sigma].$$

If the vertex functions are Boolean nor-functions as in example 2.2, then $[\mathsf{Nor}_Y, \pi] = [\mathsf{Nor}_Y, \sigma]$ implies $\pi \sim_\alpha \sigma$ (see [12]). In other words,

$$[\mathsf{Nor}_Y, \pi] = [\mathsf{Nor}_Y, \sigma] \implies \pi \sim_\alpha \sigma,$$

and it follows that $|S_Y/{\sim_\alpha}|$ is a sharp upper bound for the number of functionally non-equivalent permutation SDS maps obtainable by varying the update sequence.

Functional equivalence can also be characterized through acyclic orientations. An orientation of Y is a map $O_Y : \mathrm{e}[Y] \longrightarrow \mathrm{v}[Y] \times \mathrm{v}[Y]$ that sends an edge $\{i, j\}$ to either (i, j) or (j, i). Let $\mathsf{Acyc}(Y)$ denote the set of acyclic orientations of Y, that is, orientations that contain no directed cycles. In [16] it is shown that there is a bijection

$$f_Y : S_Y/{\sim_\alpha} \longrightarrow \mathsf{Acyc}(Y), \tag{2.6}$$

a specific example of how dynamics properties of SDSs are captured by invariants of the base graph. A permutation $\pi \in S_Y$ defines a linear order $<_\pi$ on $\mathrm{v}[Y]$ by $\pi_k = i <_\pi j = \pi_\ell$ iff $k < \ell$. This order defines an acyclic orientation O_Y^π where $O_Y^\pi(\{v, v'\})$ equals (v, v') if $v <_\pi v'$ and (v', v) otherwise. The map f_Y in (2.6) sends $[\pi]_\alpha \in S_Y/{\sim_\alpha}$ to O_Y^π. By the above remark, it follows that

$$\alpha(Y) = |\mathsf{Acyc}(Y)|$$

is a sharp upper bound for the number of functionally non-equivalent permutation SDSs that can be obtained by varying the update sequence. The result can be extended to general word update sequences $w \in W_Y$. We do not review this here, but refer to [17].

2.2. *Dynamical equivalence*

Two finite dynamical systems with maps $\phi, \psi : K^n \longrightarrow K^n$ are *dynamically equivalent* if there exists a bijection $h : K^n \longrightarrow K^n$ such that

$$\phi \circ h = h \circ \psi. \tag{2.7}$$

With the discrete topology, the concepts of dynamical equivalence and topological conjugation coincide. Thus, the difference between functional and dynamical equivalence is that in the former case the phase spaces are identical, but in the latter case the phase spaces need just be isomorphic.

It is known that symmetries in the base graph Y give rise to dynamical equivalence. Specifically, update sequences that are related by an automorphism of the base graph give rise to dynamically equivalent SDSs [12, 13]. Moreover, the number of orbits $\bar{\alpha}(Y)$ under the action of $\mathsf{Aut}(Y)$ on $S_Y/{\sim_\alpha}$ given by $\gamma \cdot [\pi]_\alpha = [\gamma * \pi]_\alpha$, where

$$\gamma * w = \big(\gamma(w_1), \ldots, \gamma(w_m)\big), \tag{2.8}$$

is an upper bound for the number of SDS maps up to dynamical equivalence. These statements follow since for SDSs with $\mathsf{Aut}(Y)$-invariant vertex functions,

$$[\mathfrak{F}_Y, \gamma * \pi] \circ \gamma = \gamma \circ [\mathfrak{F}_Y, \pi] \tag{2.9}$$

for all $\pi \in S_Y$ and all $\gamma \in \mathsf{Aut}(Y)$ (see [12]). Via the bijection in (2.6), this action carries over to an action on the set $\mathsf{Acyc}(Y)$, and the number of orbits is given by

$$\bar{\alpha}(Y) = \frac{1}{|\mathsf{Aut}(Y)|} \sum_{\gamma \in \mathsf{Aut}(Y)} \alpha(\langle\gamma\rangle \setminus Y).$$

Here, $\langle\gamma\rangle \setminus Y$ denotes the *orbit graph* of the cyclic group $G = \langle\gamma\rangle$ and Y. This bound is known to be sharp for certain graph classes, but in the general case this is still an open problem (see [4, 5]).

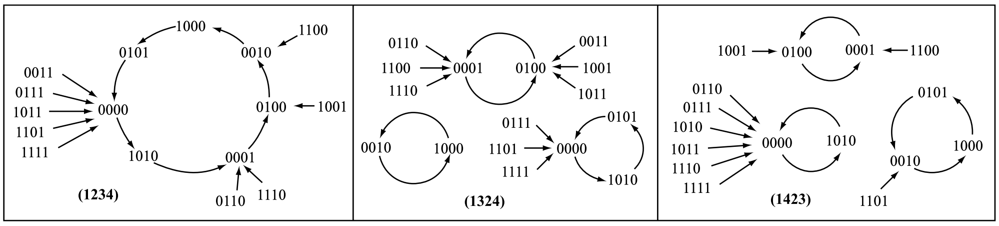


Figure 2. The phase spaces of Nor-SDS in example 3.2.

3. Cycle equivalence

Definition 3.1. *Two finite dynamical systems with maps* $\phi\colon K_1^n \longrightarrow K_1^n$ *and* $\psi\colon K_2^m \longrightarrow K_2^m$ *are cycle equivalent if there exists a bijection* $h\colon \mathsf{Per}(\phi) \longrightarrow \mathsf{Per}(\psi)$ *such that*

$$\psi|_{\mathsf{Per}(\psi)} \circ h = h \circ \phi|_{\mathsf{Per}(\phi)}, \tag{3.1}$$

where $\psi|_{\mathsf{Per}(\psi)}$ *and* $\phi|_{\mathsf{Per}(\phi)}$ *denote the restrictions of the maps to the respective sets of periodic points* $\mathsf{Per}(\psi)$ *and* $\mathsf{Per}(\phi)$. *Two maps* ϕ *and* ψ *with identical periodic orbits are functionally cycle equivalent.*

Example 3.2. *As an illustration, we continue example 2.2 with* $Y = \mathsf{Circ}_4$, *vertex functions* $\mathsf{nor}_3\colon \mathbb{F}_2^3 \longrightarrow \mathbb{F}_2$ *and consider the update sequences* $\pi = (1,2,3,4)$, $\pi' = (1,4,2,3)$ *and* $\pi'' = (1,3,2,4)$. *The phase spaces of the corresponding* SDS *maps are shown in figure 2. None of these* SDS *maps are functionally or dynamically equivalent, but* $[\mathsf{Nor}_Y, \pi']$ *and* $[\mathsf{Nor}_Y, \pi'']$ *are cycle equivalent, as illustrated in the two rightmost phase spaces in figure 2. Later we show that for* $Y = \mathsf{Circ}_4$, *there are at most* 2 *cycle configurations when* $K = \mathbb{F}_2$.

It is clear that both functional equivalence and dynamical equivalence imply cycle equivalence. Define the permutations $\sigma, \rho \in S_m$ by

$$\sigma = (m, m-1, \ldots, 2, 1), \qquad \rho = (1, m)(2, m-1)\cdots\left(\left\lceil \frac{m}{2} \right\rceil, \left\lfloor \frac{m}{2} \right\rfloor + 1\right),$$

and let C_m and D_m be the groups

$$C_m = \langle \sigma \rangle \qquad \text{and} \qquad D_m = \langle \sigma, \rho \rangle. \tag{3.2}$$

Both C_m and D_m act on the set of length-m update sequences via (2.3). Define the cyclic s-shift $\boldsymbol{\sigma}_s(w) = \sigma^s \cdot w$ and the reflection $\boldsymbol{\rho}(w) = \rho \cdot w = (w_m, w_{m-1}, \ldots, w_2, w_1)$. We can now state one of our main results.

Theorem 3.3. *For any* $w \in W_Y$, *the* SDS *maps* $[\mathfrak{F}_Y, w]$ *and* $[\mathfrak{F}_Y, \boldsymbol{\sigma}_s(w)]$ *are cycle equivalent.*

Proof. Set $P_k = \mathsf{Per}[\mathfrak{F}_Y, \boldsymbol{\sigma}_k(w)]$. By the definition of an SDS map, the following diagram commutes

$$\begin{tikzcd} P_{k-1} \arrow[r, "{[\mathfrak{F}_Y, \boldsymbol{\sigma}_{k-1}(w)]}"] \arrow[d, "F_{w(k)}"'] & P_{k-1} \arrow[d, "F_{w(k)}"] \\ P_k \arrow[r, "{[\mathfrak{F}_Y, \boldsymbol{\sigma}_k(w)]}"'] & P_k \end{tikzcd} \tag{3.3}$$

for all $1 \leqslant k \leqslant m = |w|$. Thus we obtain the inclusion $F_{w(k)}(P_{k-1}) \subset P_k$, and since the restriction map $F_{w(k)}\colon P_{k-1} \longrightarrow F_{w(k)}(P_{k-1})$ is an injection, it follows that $|P_{k-1}| \leqslant |P_k|$. We therefore obtain the sequence of inequalities:

$$\big|\mathsf{Per}[\mathfrak{F}_Y, w]\big| \leqslant \big|\mathsf{Per}[\mathfrak{F}_Y, \boldsymbol{\sigma}_1(w)]\big| \leqslant \cdots \leqslant \big|\mathsf{Per}[\mathfrak{F}_Y, \boldsymbol{\sigma}_{m-1}(w)]\big| \leqslant \big|\mathsf{Per}[\mathfrak{F}_Y, w]\big|,$$

from which it follows that all inequalities are, in fact, equalities. Since the graph and state space are finite, the restriction maps $F_{w(k)}$ in (3.3) are bijections. Clearly (3.1) holds with $h = F_{w(k)}$, and the proof follows. □

Theorem 3.3 shows how acting on the update sequence by the cyclic group C_m preserves the cycle structure of the phase space. We point out again that this result holds for any finite set K. For $K = \mathbb{F}_2$ the cycle structure is also preserved under the action of D_m and is a consequence of the following result.

Proposition 3.4 ([12]). *Let $(Y, \mathfrak{F}_Y, w)$ be an* SDS *over $\mathbb{F}_2$ with periodic points $P \subset \mathbb{F}_2^n$. Then*

$$\big([\mathfrak{F}_Y, w]\big|_P\big)^{-1} = [\mathfrak{F}_Y, \boldsymbol{\rho}(w)]\big|_P\ . \tag{3.4}$$

The proof rests on the fact that for each vertex function f_i, the restriction $f_i(-; y'[i])\colon \mathbb{F}_2 \longrightarrow \mathbb{F}_2$ is a bijection for each fixed choice of $y'[i]$. There are only two such maps: the identity map $y_i \mapsto y_i$ and the map $y_i \mapsto 1 + y_i$. From this it follows that composing the two maps in (3.4) in either order gives the identity map. The next proposition is now clear.

Proposition 3.5. *For $K = \mathbb{F}_2$ the* SDS *maps $[\mathfrak{F}_Y, w]$ and $[\mathfrak{F}_Y, \boldsymbol{\rho}(w)]$ are cycle equivalent.*

Thus, for any $g \in G = C_m$ the SDS maps $[\mathfrak{F}_Y, w]$ and $[\mathfrak{F}_Y, g \cdot w]$ are cycle equivalent where $|w| = m$. If $K = \mathbb{F}_2$, the same statement holds for $G = D_m$. We now have the following situation: update sequences π and π' with $[\pi]_\alpha \neq [\pi']_\alpha$ generally give rise to functionally non-equivalent SDS maps. However, if there exists $g \in G, \bar{\pi} \in [\pi]_\alpha$ and $\bar{\pi}' \in [\pi']_\alpha$ such that $g \cdot \bar{\pi} = \bar{\pi}'$, then the classes $[\pi]_\alpha$ and $[\pi']_\alpha$ induce cycle equivalent SDS maps.

As a particular example, let Star_n be the graph with vertex set $\mathrm{v}[\mathsf{Star}_n] = \{0, 1, \ldots, n\}$ and edge set $\mathrm{e}[\mathsf{Star}_n] = \big\{\{0, i\} \mid 1 \leqslant i \leqslant n\big\}$.

Corollary 3.6. *Let $Y = \mathsf{Star}_n$ with $n \geqslant 2$. For a fixed sequence $\mathfrak{F}_Y$ of $\mathsf{Aut}(Y)$-invariant Y-local maps, all permutation* SDS *maps of the form $[\mathfrak{F}_Y, \pi]$ are cycle equivalent. Any* SDS *map of the form $[\mathsf{Nor}_Y, \pi]$ with $\pi \in S_Y$ has precisely one periodic orbit of size three and $2^{n-1} - 1$ periodic orbits of size two.*

Proof. We have $\mathsf{Aut}(\mathsf{Star}_n) \cong S_n$ since the automorphisms of Star_n are precisely the elements of S_Y that fix the vertex 0. An orbit of $\mathsf{Aut}(\mathsf{Star}_n)$ in $S_Y/\!\sim_\alpha$ contains all equivalence classes $[\pi]_\alpha$ for which the positions of 0 in π coincide. Thus for $0 \leqslant i \leqslant n$ all permutations that have vertex 0 in the ith position give rise to dynamically equivalent SDS maps. Pick $\pi = (0, 1, 2, \ldots, n)$. By corollary 3.5, all permutations that are shifts of π give cycle equivalent SDS maps. The second part now follows by inspection of one of the possible phase spaces. They are all listed in [12], but without enumerations of periodic orbits. □

4. Combinatorial constructions for cycle equivalence

4.1. Neutral networks

In the rest of this paper we will only consider permutation update sequences, although it is not hard to see how this analysis can be extended to systems with general word update sequences.

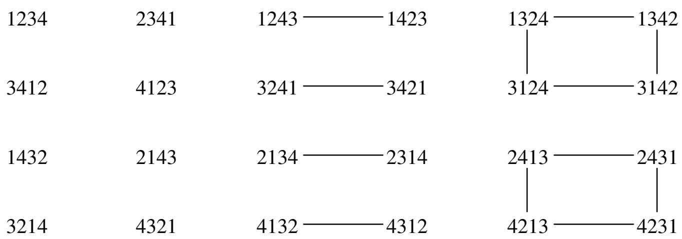


Figure 3. The update graph $U(\mathsf{Circ}_4)$.

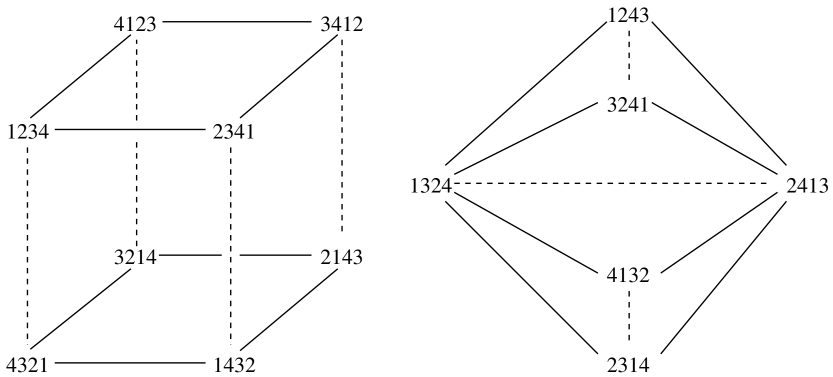


Figure 4. The graphs $C(\mathsf{Circ}_4)$ and $D(\mathsf{Circ}_4)$. The dashed lines are edges in $D(\mathsf{Circ}_4)$ but not in $C(\mathsf{Circ}_4)$.

To start, we define two graphs over $S_Y/\sim_\alpha$ whose connected components give rise to cycle equivalent SDSs for a fixed graph Y and a fixed sequence $\mathfrak{F}_Y$. Since cycle equivalence is a coarsening of functional equivalence, it is natural to construct these graphs using $S_Y/\sim_\alpha$ as the vertex set rather than S_Y.

Let $C(Y)$ and $D(Y)$ be the graphs defined by

$$\mathrm{v}[C(Y)] = S_Y/\sim_\alpha, \qquad \mathrm{e}[C(Y)] = \left\{\{[\pi]_\alpha, [\sigma_1(\pi)]_\alpha\} \mid \pi \in S_Y\right\},$$

$$\mathrm{v}[D(Y)] = S_Y/\sim_\alpha, \qquad \mathrm{e}[D(Y)] = \left\{\{[\pi]_\alpha, [\rho(\pi)]_\alpha\} \mid \pi \in S_Y\right\} \cup \mathrm{e}[C(Y)].$$

Define $\kappa(Y)$ and $\delta(Y)$ to be the number of connected components of $C(Y)$ and $D(Y)$, respectively. It is clear that $C(Y)$ is a subgraph of $D(Y)$ and that $\delta(Y) \leqslant \kappa(Y)$. By theorem 3.3, $\kappa(Y)$ is a general upper bound for the number of different SDS cycle equivalence classes obtainable through update sequence variations. For $K = \mathbb{F}_2$ it follows from proposition 3.4 that $\delta(Y)$ is an upper bound as well.

Example 4.1. *As in example 2.2, let $Y = \mathsf{Circ}_4$. The permutation update graph $U(\mathsf{Circ}_4)$ has $\alpha(\mathsf{Circ}_4) = 14$ connected components as shown in figure 3. The graphs $C(\mathsf{Circ}_4)$ and $D(\mathsf{Circ}_4)$ are shown in figure 4 where the dashed lines represent the edges in $\mathrm{e}[D(\mathsf{Circ}_4)] \setminus \mathrm{e}[C(\mathsf{Circ}_4)]$. The vertices in figure 4 are labelled by a permutation in the corresponding equivalence class in $S_Y/\sim_\alpha$. The vertices of the cube-shaped component are all singletons in $S_Y/\sim_\alpha$. The equivalence classes $[1324]_{\mathsf{Circ}_4}$ and $[2413]_{\mathsf{Circ}_4}$ both consist of four permutations, while the remaining four vertices on that component are equivalence classes that contain precisely two permutations. Clearly, $\kappa(\mathsf{Circ}_4) = 3$ and $\delta(\mathsf{Circ}_4) = 2$.*

The following result gives insight into the structure of $C(Y)$ and $D(Y)$.

Proposition 4.2. *Let Y be a connected graph on n vertices and let $g, g' \in C_n$ with $g \neq g'$. Then $[g \cdot \pi]_\alpha \neq [g' \cdot \pi]_\alpha$. If $g, g' \in D_n$ with $g \neq g'$ then $[g \cdot \pi]_\alpha = [g' \cdot \pi]_\alpha$ can hold if and only if Y is bipartite.*

Proof. Assume $g \neq g'$ with $[g \cdot \pi]_\alpha = [g' \cdot \pi]_\alpha$. By construction, we have $g \cdot \pi = \boldsymbol{\sigma}_s(\pi)$ and $g' \cdot \pi = \boldsymbol{\sigma}_{s'}(\pi)$. Without loss of generality we may assume $s' < s$. Let $V' \subset V = \mathrm{v}[Y]$ be the initial subsequence of vertices in $\boldsymbol{\sigma}_{s'}(\pi)$ that occurs at the end in $\boldsymbol{\sigma}_s(\pi)$. If any of the vertices in V' are adjacent to any of the vertices in $V \setminus V'$ in Y, it would imply that $[\boldsymbol{\sigma}_s(\pi)]_\alpha \neq [\boldsymbol{\sigma}_{s'}(\pi)]_\alpha$. The only possibility is that Y is not connected, but this contradicts the assumptions of the proposition.

Next consider the second statement. From $[g \cdot \pi]_\alpha = [g' \cdot \pi]_\alpha$ it follows from above that g and g' lie in different cosets of C_n in D_n. Thus, we may without loss of generality assume that $g = \sigma^s$ and $g' = \rho\sigma^{s'}$. Let $m = |s' - s|$ and $m' = n - m$. If $s' > s$ (respectively, $s' < s$) the first (respectively, last) m elements of $g \cdot \pi$ and $g' \cdot \pi$ are the same but occur in reverse order. Call the set of these elements V_1. The remaining m' elements occur in reverse order as well in the two permutations. Let V_2 denote the set of these elements. For $[g \cdot \pi]_\alpha = [g' \cdot \pi]_\alpha$ to hold, there cannot be an edge between any two vertices in V_1 or between any two vertices in V_2. Therefore, the graph Y must be a subgraph of $K(V_1, V_2)$, the complete bipartite graph with vertex sets V_1 and V_2. $\square$

Remark 4.3. The pairs $(\sigma^s, \rho\sigma^{s'})$ and $(\sigma^{s'}, \rho\sigma^s)$ determine the same bipartite graph in the above proof. Also, the vertex sets V_1 and V_2 can only consist of consecutive elements in π.

Remark 4.4. If Y is connected and bipartite then $|\{[g \cdot \pi]_\alpha \mid g \in D_n\}| = 2n - 1$. This follows from the fact that at most two $\sim_\alpha$ classes in the first proof argument can coincide as all distinct pairs g and g' for which equality holds lead to different sets $V_1(\{g, g'\})$ and $V_2(\{g, g'\})$ modulo remark 4.3. The existence of two or more distinct partitions of $\mathrm{v}[Y]$ into sets V_1 and V_2 as above would imply that Y is not connected.

4.2. Source-to-sink operations and reflections of acyclic orientations

In this section we show how the component structure of $C(Y)$ is precisely captured through *source-to-sink* operations on acyclic orientations. The bijection in (2.6) identifies $[\pi]_\alpha$ with $O_Y^\pi \in \mathsf{Acyc}(Y)$. For any $\pi \in [\pi']_\alpha$, the orientation $O_Y^{\sigma_1(\pi)}$ is constructed from O_Y^π by converting vertex π_1 from a source to a sink. Following [19] we call such a conversion a *source-to-sink operation* or a *click*. It can be easily verified that this gives rise to an equivalence relation $\sim_\kappa$ on $\mathsf{Acyc}(Y)$. More precisely, two orientations $O_Y, O'_Y \in \mathsf{Acyc}(Y)$ where O_Y can be transformed into O'_Y by a sequence of clicks are said to be κ-equivalent, and we write

$$\mathsf{Acyc}(Y)/\sim_\kappa$$

for the set of equivalence classes. This observation along with theorem 3.3 shows that permutations that belong to κ-equivalent acyclic orientations induce cycle equivalent SDSs. By construction, the source–sink operation precisely encodes adjacency in the graph $C(Y)$, and the connected components are in 1–1 correspondence with the κ-equivalence classes. Therefore, the number of equivalence classes in $\mathsf{Acyc}(Y)$ under the source–sink relation equals $\kappa(Y)$ and is thus an upper bound for the number of cycle equivalent permutation SDS maps $[\mathfrak{F}_Y, \pi]$ for a fixed sequence $\mathfrak{F}_Y$.

If $K = \mathbb{F}_2$ then proposition 3.4 shows that reflections of update sequences also induce cycle equivalent SDSs. On the level of acyclic orientations this corresponds to reversing all edge orientations. Through the bijection (2.6) this identifies O_Y^{π} with the reverse orientation $O_Y^{\rho(\pi)}$, the unique orientation that satisfies $O_Y^{\pi}(\{i, j\}) \neq O_Y^{\rho(\pi)}(\{i, j\})$ for every $\{i, j\} \in \mathrm{e}[Y]$. If two acyclic orientations are related by a sequence of source-to-sink operations and reflections, then they are said to be *δ-equivalent*.

The notion of κ- and δ-equivalence carries over naturally to update sequences as well. Two update sequences in S_Y belonging to $\sim_\alpha$ classes on the same connected component in $C(Y)$ (respectively, $D(Y)$) are called *κ-equivalent* (respectively, *δ-equivalent*). For two κ-equivalent update sequences π and π', there is a sequence of adjacent non-edge transpositions and cyclic shifts that map π to π'. This is simply a consequence of the definition of $S_Y/\sim_\alpha$ and $C(Y)$.

We remark that there is a close connection between κ-classes and the structure of conjugacy classes of Coxeter elements, which we revisit in section 7.

5. Enumeration for $\kappa(Y)$ and $\delta(Y)$

In the introduction we remarked that κ and δ may be viewed as complexity measures for SDSs. To start, we show that $\delta(Y)$ can be characterized in terms of $\kappa(Y)$.

Proposition 5.1. *Let Y be a connected graph. If Y is not bipartite then $\delta(Y) = \frac{1}{2}\kappa(Y)$. If Y is bipartite then $\delta(Y) = \frac{1}{2}(\kappa(Y) + 1)$.*

The proof uses the following technical lemma.

Lemma 5.2. *The reflection map $\rho\colon S_Y \longrightarrow S_Y$ extends to an involution*

$$\rho^*\colon \mathsf{Acyc}(Y)/\sim_\kappa \longrightarrow \mathsf{Acyc}(Y)/\sim_\kappa . \tag{5.1}$$

Proof. By the definition of the update graph $U(Y)$ it follows that if $\pi, \pi' \in S_Y$ are adjacent in $U(Y)$ then so are $\rho(\pi)$ and $\rho(\pi')$. By induction it follows that $\pi \sim_\alpha \pi'$ implies $\rho(\pi) \sim_\alpha \rho(\pi')$. The map ρ therefore extends to a map $\hat{\rho}\colon S_Y/\sim_\alpha \longrightarrow S_Y/\sim_\alpha$ by $\rho([\pi]_\alpha) = [\rho(\pi)]_\alpha$. Likewise, if O_Y and O'_Y are κ-equivalent then so are $\hat{\rho}(O_Y)$ and $\hat{\rho}(O'_Y)$ (using bijection (2.6)), and $\hat{\rho}$ extends to ρ^* as in (5.1) by $\rho^*(A) = \hat{\rho}(O_A)$ for any $O_A \in A \in \mathsf{Acyc}(Y)/\sim_\kappa$. This map is clearly an involution since ρ itself is an involution. □

Proposition 5.1 now follows since ρ^* has no fixed points if Y is not bipartite and has precisely one fixed point if Y is bipartite. Thus, we always have $\delta(Y) = \lceil \kappa(Y)/2 \rceil$, and we also have the following characterization of bipartite graphs.

Corollary 5.3. *A connected graph Y is bipartite if and only if $\kappa(Y)$ is odd.*

In light of proposition 5.1, we focus on the computation of $\kappa(Y)$ in the following.

Proposition 5.4 ([10]). *If Y is the disjoint union of graphs Y_1 and Y_2 or if Y is a graph with a bridge $e = \{v, w\}$ connecting the subgraphs Y_1 and Y_2, then*

$$\kappa(Y) = \kappa(Y_1)\kappa(Y_2). \tag{5.2}$$

Proof. Regarding the first statement, observe that any acyclic orientation of the disjoint union Y of graphs Y_1 and Y_2 is of the form $O_Y = (O_{Y_1}, O_{Y_2})$. From the definitions, it is clear that two such acyclic orientations O_Y and O'_Y of Y are κ-equivalent if and only if the respective acyclic orientations over Y_1 and Y_2 are κ-equivalent.

If $e = \{v, w\}$ is a bridge in Y connecting Y_1 and Y_2, then every acyclic orientation of Y is of the form $O_Y = (O_{Y_1}, (v, w), O_{Y_2})$ or $O'_Y = (O_{Y_1}, (w, v), O_{Y_2})$, where O_{Y_1} and O_{Y_2} are acyclic orientations of Y_1 and Y_2, respectively. However, it is easy to see that clicking every vertex of Y_1 in O_Y precisely once will map O_Y to O'_Y, from which the second statement follows. □

For the computation of $\kappa(Y)$ we may therefore assume that Y is connected and that every edge is a cycle edge. Note that for the empty graph on n vertices E_n we have $\kappa(E_n) = 1$ since $\alpha(E_n) = 1$. The following corollary is immediate from proposition 5.4.

Corollary 5.5. *Let Y be a forest. Then $\kappa(Y) = \delta(Y) = 1$.*

From corollary 5.5 we get the following perhaps surprising results on dynamics of SDSs over trees.

Proposition 5.6. *Let Y be a forest and $\mathfrak{F}_Y$ be a sequence of Y-local functions. Then all permutation SDS maps $[\mathfrak{F}_Y, \pi]$ are cycle equivalent.*

The proof is clear since κ-equivalent update sequences induce cycle equivalent systems. In other words, when Y is a forest, all permutation SDS maps of the form $[\mathfrak{F}_Y, \pi]$ for a fixed $\mathfrak{F}_Y$ share the same cycle configuration. This result may not be that significant if all or most of the periodic points are fixed points. However, for other functions, such as invertible ones, it is powerful. The parity functions $\mathsf{par}_k\colon \mathbb{F}_2^k \longrightarrow \mathbb{F}_2$ are defined as $\mathsf{par}(\mathbf{y}) = \sum_i y_i$, modulo 2, and their corresponding Y-local maps are invertible for every graph Y (see [12]). Let Par_Y be the sequence of Y-local functions induced by the parity vertex functions.

Corollary 5.7. *If Y is a forest, then for any $\pi, \sigma \in S_Y$, the maps $[\mathsf{Par}_Y, \pi]$ and $[\mathsf{Par}_Y, \sigma]$ are dynamically equivalent.*

The same result holds for the logical negation of the parity function, which is also invertible. The next result implies that the quantity $\kappa(Y)$ is a *Tutte–Grothendieck invariant*.

Theorem 5.8 ([10]). *Let e be a cycle edge of Y. Then*

$$\kappa(Y) = \kappa(Y'_e) + \kappa(Y''_e), \tag{5.3}$$

where Y'_e is the graph obtained from Y by deleting e and Y''_e is the graph obtained from Y by contracting e.

The proof of theorem 5.8 is quite involved, and along with proposition 5.4, it implies that $\kappa(Y) = T_Y(1, 0)$, where $T_Y(x, y)$ is the Tutte polynomial [20]. In contrast, it is well known that the number of acyclic orientations of a graph satisfies $\alpha(Y) = T_Y(2, 0)$.

A complete set of cycle equivalence class representatives. For functional equivalence of SDSs, bijection (2.6) allows us to construct a complete set of representative update sequences. There is a similar bijection for cycle equivalence. To show this, let $\mathsf{Acyc}_v(Y)$ denote the acyclic orientations of Y where vertex v is the unique source. We then have the following proposition.

Proposition 5.9 ([10]). *Let Y be a connected graph. For any fixed $v \in \mathrm{v}[Y]$, there is a bijection*

$$\phi_v\colon \mathsf{Acyc}_v(Y) \longrightarrow \mathsf{Acyc}(Y)/\sim_\kappa .$$

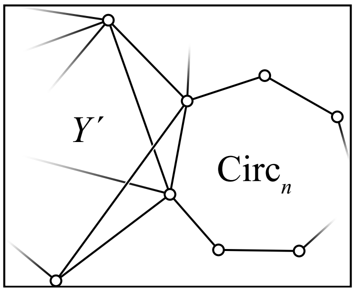


Figure 5. A graph Y with an n-handle.

Thus, from the set $\mathsf{Acyc}_v(Y)$ we can construct a complete set of update sequence representatives for the possible phase space cycle configurations. Incidentally, since $\mathsf{Acyc}_v(Y)$ is a transversal for any $v \in \mathrm{v}[Y]$ it follows that each κ-class contains at least $n = |\mathrm{v}[Y]|$ elements and hence $\kappa(Y) \leqslant \alpha(Y)/n$.

We now give examples of the computation of κ. Even though some of these results may be derived as special cases of more general results, they are needed for the examples in section 6. We begin with a result for $\kappa(Y \oplus v)$, the vertex-join of Y and the vertex v. Recall that the graph $Y \oplus v$ is constructed from Y by adding to Y the vertex v and joining this new vertex to each vertex of Y.

Proposition 5.10. *If Y is a graph with* $\mathrm{e}[Y] \neq \varnothing$, *then*

$$\kappa(Y \oplus v) = 2\delta(Y \oplus v) = \alpha(Y). \tag{5.4}$$

Proof. From proposition 5.9, we know that each κ-class of $\mathsf{Acyc}(Y \oplus v)$ contains a unique acyclic orientation where v is the unique source. Any acyclic orientation of Y thus yields a unique element of $\mathsf{Acyc}_v(Y \oplus v)$ and vice versa, so there is a bijection between $\mathsf{Acyc}(Y \oplus v)/\sim_\kappa$ and $\mathsf{Acyc}(Y)$. Finally, since Y contains an edge, $Y \oplus v$ is not bipartite; hence $\kappa(Y \oplus v) = 2\delta(Y \oplus v)$ and (5.4) follows. $\square$

Corollary 5.11. *Let K_n denote the complete graph on n vertices. For $n \geqslant 2$ we have $\kappa(K_n) = (n-1)!$*

Proof. There are $2^{\binom{n}{2}}$ orientations of K_n, and by the bijection in (2.6), precisely $\alpha(K_n)$ of these are acyclic, and this is equal to the number of components of the update graph $U(K_n)$. Since $U(K_n)$ consists of the $n!$ singleton vertices in S_Y we have $\alpha(K_n) = n!$, and from proposition 5.10 it follows that $\kappa(K_n) = \alpha(K_{n-1}) = (n-1)!$ $\square$

A graph Y has an n-handle if it is of the form $Y = Y' \cup \mathsf{Circ}_n$ where Y' and Circ_n share precisely one edge as illustrated in figure 5.

Proposition 5.12. *Let Y be a graph with an n-handle where $Y = Y' \cup \mathsf{Circ}_n$. Then*

$$\kappa(Y) = (n-1)\kappa(Y'). \tag{5.5}$$

Proof. Let $e' = \{v, v'\}$ be the edge shared by Y' and Circ_n and let e be the edge in Circ_n incident with v. By applying theorem 5.8 and proposition 5.4 we obtain

$$\kappa(Y' \cup \mathsf{Circ}_n) = \kappa(Y') + \kappa(Y' \cup \mathsf{Circ}_{n-1}).$$

Equation (5.5) follows through repeated applications of this recursion process. $\square$

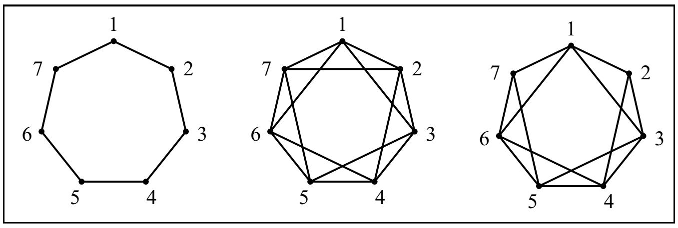


Figure 6. From left to right: the graphs Circ_7, $\mathsf{Circ}_{7,2}$ and $\mathsf{Circ}'_{7,2}$.

As a simple, special case of proposition 5.12 we obtain $\kappa(\mathsf{Circ}_n) = n - 1$. Just take Y' to be the graph with the vertex set $\mathrm{v}[Y'] = \{1, n\}$ and the edge set $\{\{1, n\}\}$ in proposition 5.12.

6. $\kappa(Y)$ as a complexity measure

The number of possible orbit structures that one can obtain by varying the update sequence is a natural measure for system complexity. As we have shown, $\kappa(Y)$ is a general upper bound for this number, and so is $\delta(Y)$ in the case of binary states. Since these bounds are graph measures, we can characterize the complexity of dynamics in terms of the GDS base graph. As we have seen, bridge edges do not contribute to periodic orbit variability, so it suffices to consider the cycles of the graph. As can be seen in the case of Circ_n, increasing the size of a cycle does not contribute much to the value of κ, e.g. $\kappa(\mathsf{Circ}_{n+1}) = \kappa(\mathsf{Circ}_n) + 1$. However, from the result on graphs with handles it follows that even the addition of a minimal handle *doubles* the measure κ, i.e. $\kappa(Y \cup \mathsf{Circ}_3) = 2\kappa(Y)$, where Y and Circ_3 share precisely one edge.

The following example shows the effect on complexity that results from increasing the radius of the rules in asynchronous, elementary cellular automata.

Example 6.1 (Asynchronous elementary CA: rule radius versus periodic orbit complexity). *We have seen that $\kappa(\mathsf{Circ}_n) = n - 1$. Thus, for any fixed sequence of radius-1 vertex functions the number of distinct periodic orbit configurations is $O(n)$. This changes dramatically for radius-2 rules. In this case the GDS base graph is $\mathsf{Circ}_{n,2}$ defined by*

$$\mathrm{v}[\mathsf{Circ}_{n,2}] = \{1, 2, \ldots, n\} \qquad \text{and} \qquad \mathrm{e}[\mathsf{Circ}_{n,2}] = \big\{\{i, j\} \mid 1 \leqslant |i - j| \leqslant 2\big\},$$

with index arithmetic modulo n. The auxiliary graph $\mathsf{Circ}'_{n,2}$ is obtained from $\mathsf{Circ}_{n,2}$ by deleting the edge $\{2, n\}$. The case $n = 7$ is illustrated in figure 6.

For simplicity we set $g_n = \kappa(\mathsf{Circ}_{n,2})$ and $c_n = \kappa(\mathsf{Circ}'_{n,2})$. Successive use of recurrence (5.3) with edges $e_1 = \{1, n\}$ and $e_2 = \{1, n-1\}$ for both $\mathsf{Circ}_{n,2}$ and $\mathsf{Circ}'_{n,2}$ gives

$$c_n = c_{n-1} + 2c_{n-2} + 2^{n-2} \qquad \text{and} \qquad g_n = g_{n-2} + c_n + 2c_{n-2},$$

where $c_5 = 18$, $c_6 = 46$, $g_5 = 24$ and $g_6 = 64$. These recurrence relations are straightforward to solve with

$$\kappa(\mathsf{Circ}'_{n,2}) = \big[(3n - 5)2^n - 4(-1)^n\big]/18$$

and

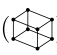
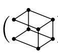
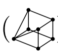
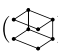
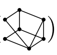
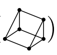
$$\kappa(\mathsf{Circ}_{n,2}) = \big[(2n - 6)2^n + 9 - (2n - 3)(-1)^n\big]/6.$$

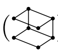
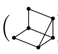
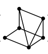
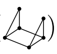
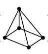
Thus, by increasing the rule radius from 1 to 2 we see that the number of distinct periodic orbit configurations becomes $O(n \cdot 2^n)$. The corresponding bounds for δ are easily obtained from proposition 5.1.

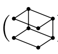
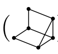
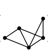
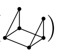
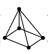
The effect of graph symmetries. We have seen how non-trivial symmetries in the base graph give rise to dynamically equivalent SDS maps when the vertex functions are $\mathsf{Aut}(Y)$-invariant. Since dynamical equivalence implies cycle equivalence we can construct a bound $\bar{\kappa}(Y)$ in the same manner as for $\bar{\alpha}(Y)$. This bound $\bar{\kappa}(Y)$ thus reflects the additional cycle equivalences that are due to graph symmetries and that are applicable in the case of $\mathsf{Aut}(Y)$-invariant vertex functions. We close with an example that illustrates this and the results of theorem 5.8 and propositions 5.4, 5.10 and 5.12.

Example 6.2. *Let $Y = Q_2^3$ be the binary 3-cube, which has an automorphism group isomorphic to $S_4 \times \mathbb{Z}_2$. It is shown in [4] that $\alpha(Q_2^3) = 1862$ and that $\bar{\alpha}(Q_2^3) = 54$. Thus, there are at most 1862 functionally non equivalent permutation SDSs over Q_2^3 for a fixed sequence of vertex functions. Likewise, there are at most 54 dynamically non equivalent $\mathsf{Aut}(Q_2^3)$-invariant permutation SDSs. It is known in this case that the bound $\bar{\alpha}(Q_2^3)$ is sharp, since it is realized for SDSs induced by the nor_4-function.*

The number of cycle equivalence classes is bounded above by $\kappa(Q_2^3)$, and from the recursion relation (5.3) we get (with some foresight at each step)

$\kappa($$) = \kappa($$) + \kappa($$) = \kappa($$) + 2\kappa($$) + \kappa($$)$

$= \kappa($$) + 2\kappa($$) + 2\kappa($$) + \kappa($$) + \kappa($$)$

$= \kappa($$) + 4\kappa($$) + 2\kappa($$) + \kappa($$) + \kappa($$)$

$$= 27 + 64 + 16 + 12 + 14 = 133\,,$$

where propositions 5.10 and 5.12 were used in the last step. Since Q_2^3 is bipartite we also derive $\delta(Q_2^3) = (133 + 1)/2 = 67$, and thus in the case of $K = \mathbb{F}_2$, there are at most 67 cycle classes for a fixed sequence of vertex functions. Straightforward (but somewhat lengthy) calculations show that $\bar{\kappa}(Q_2^3) = \bar{\delta}(Q_2^3) = 8$. In conclusion, we have

$$\alpha(Q_2^3) = 1862, \quad \bar{\alpha}(Q_2^3) = 54, \quad \kappa(Q_2^3) = 133, \quad \delta(Q_2^3) = 67, \quad \bar{\kappa}(Q_2^3) = \bar{\delta}(Q_2^3) = 8.$$

Thus, if $\mathfrak{F}_Y$ is a sequence of $\mathsf{Aut}(Q_2^3)$-invariant Y-local functions, there are at most eight different periodic orbit configurations for permutation SDS maps $[\mathfrak{F}_Y, \pi]$ up to isomorphism. Moreover, because $\bar{\kappa}(Q_2^3) = \bar{\delta}(Q_2^3)$, taking vertex states from $K = \mathbb{F}_2$ does not improve this bound. In summary, using the various notions of equivalence described in this paper, we have shown that the $8! = 40\,320$ different permutation update sequences leads to at most 8 non-equivalent periodic orbit configurations in the case of $Y = Q_2^3$ and $\mathsf{Aut}(Y)$-invariant vertex functions. Although over Q_2^3, the bound 8 is realized for nor-functions, it is not known if this bound is sharp for general graphs.

This example is only meant as an illustration, and a systematic treatment incorporating the analysis of the functions $\bar{\kappa}$ and $\bar{\delta}$ for general graphs will be pursued elsewhere.

7. Summary

In this paper we have given an overview of GDSs and have shown how shifts and reflections of update sequences give rise to SDSs with equivalent long-term dynamics. Additionally, we have shown how to bound the number of periodic orbit configurations and have derived several properties of this bound $\kappa(Y)$. For binary states we have shown how $\delta(Y)$ is a sharper bound. Both quantities κ and δ can serve as graph measures for dynamical complexity. *This work* *follows the broad research theme for* GDSs *which is to link the defining properties such as the graph structure and the global dynamics of the* GDS *map.*

Moreover, we have shown how shifts of update sequences correspond to source-to-sink conversions in acyclic orientations. Source-to-sink conversions also appear in the context of Coxeter theory (see, e.g., [7] for definitions). For a Coxeter group with Coxeter graph Y, the number of conjugacy classes of Coxeter elements (see [18]) is also bounded above by $\kappa(Y)$, e.g. [19]. In general, it is not known if $\kappa(Y)$ is a sharp bound, but it was shown to hold true for (unlabelled) unicyclic graphs by Shi in [19]. Recently, in [11], we proved the sharpness without the unicyclic assumption. This extends Shi's solution of the conjugacy problem to a large class of Coxeter groups, which includes all simply-laced Coxeter groups—those with arbitrary (unlabelled) Coxeter graph. This is an explicit example of how this new connection between Coxeter theory and SDSs has already led to new results on Coxeter groups, and likewise it could be very helpful in further exploring the properties of asynchronous GDSs.

In this paper, we have not explored the question of when κ (and δ when $K = \mathbb{F}_2$) is a sharp bound. That is, for an arbitrary graph Y, does there exist a sequence of vertex functions whose number of non-equivalent orbit configurations equals $\kappa(Y)$? Proving this would require one to construct such functions for any given graph. We have also omitted computational aspects related to cycle equivalence. For results on complexity theoretical properties of SDS, see [2, 3], where fixed point reachability and other problems related to computational issues are analysed. Additional future work includes extending our results from permutation update sequences to general word update sequences, as well as further exploring the effects of symmetries of the base graph and the computation of the bounds $\bar{\kappa}$ and $\bar{\delta}$ as illustrated in example 6.2.

Acknowledgments

The first author would like to thank Jon McCammond for many helpful discussions. Both authors are grateful to the NDSSL group at Virginia Tech for the support to this research. This work was partially supported by Fields Institute in Toronto, Canada.

References

[1] Barrett C L, Bisset K, Eubank S, Kumar V S A, Marathe M V and Mortveit H S 2007 *Modeling and Simulation of Large Biological, Information and Soci-Technical Systems: An Interaction-Based Approach* (*Proc. Symposia in Applied Mathematics* vol 64) (Providence, RI: American Mathematical Society) pp 101–48

[2] Barrett C L, Hunt H B III, Marathe M V, Ravi S S, Rosenkrantz D J and Stearns R E 2006 Complexity of reachability problems for finite discrete sequential dynamical systems *J. Comput. Syst. Sci.* **72** 1317–45

[3] Barrett C L, Hunt H H III, Marathe M V, Ravi S S, Rosenkrantz D, Stearns R and Tosic P 2001 Garden of eden and fixed point configurations in sequential dynamical systems *Proc. Int. Conf. on Discrete Models in Combinatorics, Computation and Geometry* pp 95–110

[4] Barrett C L, Mortveit H S and Reidys C M 2001 Elements of a theory of simulation: III. Equivalence of SDS *Appl. Math. Comput.* **122** 325–40

[5] Barrett C L, Mortveit H S and Reidys C M 2003 Elements of a theory of simulation: IV. Fixed points, invertibility and equivalence *Appl. Math. Comput.* **134** 153–72

[6] Barrett C L and Reidys C M 1999 Elements of a theory of simulation: I. Sequential CA over random graphs *Appl. Math. Comput.* **98** 241–59

[7] Björner A and Brenti F 2005 *Combinatorics of Coxeter Groups* (*GTM* vol 231) (Berlin: Springer)

[8] Eubank S, Guclu H, Kumar V S A, Marathe M V, S A, Toroczkai Z and Wang N 2004 Modelling disease outbreaks in realistic urban social networks *Nature* **429** 180–4

[9] Karaoz U, Murali T, Letovsky S, Zheng Y, Ding C, Cantor C R and Kasif S 2004 Whole-genome annotation by using evidence integration in functional-linkage networks *Proc. Natl Acad. Sci.* **101** 2888–93

[10] Macauley M and Mortveit H S 2008 On enumeration of conjugacy classes of Coxeter elements *Proc. Am. Math. Soc.* **136** 4157–65

[11] Macauley M and Mortveit H S 2008 A solution to the conjugacy problem for Coxeter elements in simply-laced Coxeter groups, submitted

[12] Mortveit H S and Reidys C M 2001 Discrete, sequential dynamical systems *Discrete Math.* **226** 281–95

[13] Mortveit H S and Reidys C M 2007 *An Introduction to Sequential Dynamical Systems* (New York: Springer)

[14] Nagel K and Schreckenberg M 1992 A cellular automaton model for freeway traffic *J. Physique* I **2** 2221–9

[15] Pautz S 2002 An algorithm for parallel S_n sweeps on unstructured meshes *Nucl. Sci. Eng.* **140** 111–36

[16] Reidys C M 1998 Acyclic orientations of random graphs *Adv. Appl. Math.* **21** 181–92

[17] Reidys C M 2006 Sequential dynamical systems over words *Ann. Comb.* **10** 481–98

[18] Shi J Y 1997 The enumeration of Coxeter elements *J. Algebr. Comb.* **6** 161–71

[19] Shi J Y 2001 Conjugacy relation on Coxeter elements *Adv. Math.* **161** 1–19

[20] Tutte W T 1954 A contribution to the theory of chromatic polynomials *Can. J. Math.* **6** 80–91