Dividing permutations in the semiring of functional digraphs

Florian Bridoux[1], Christophe Crespelle[1], Thi Ha Duong Phan[2], and Adrien Richard[1]

[1] Université Côte d'Azur, CNRS, I3S, Sophia Antipolis, France
{florian.bridoux,christophe.crespelle,adrien.richard}@univ-cotedazur.fr
[2] Institute of Mathematics, Vietnam Academy of Science and Technology, Viet Nam
phanhaduong@math.ac.vn

Abstract. Functional digraphs are unlabelled finite digraphs where each vertex has exactly one out-neighbor. They are isomorphic classes of finite discrete-time dynamical systems. Endowed with the direct sum and product, functional digraphs form a semiring with an interesting multiplicative structure. For instance, we do not know if the following division problem can be solved in polynomial time: given two functional digraphs A and B, does A divide B? That A divides B means that there exist a functional digraph X such that AX is isomorphic to B, and many such X can exist. We can thus ask for the number of solutions X. In this paper, we focus on the case where B is a permutation, that is, a disjoint union of cycles. There is then a naïve sub-exponential algorithm to compute the number of non-isomorphic solutions X, and our main result is a polynomial algorithm when A is fixed. It uses a divide-and-conquer technique that should be useful for further developments on the division problem.

Keywords: Finite Dynamical Systems · Functional digraphs · Graph direct product.

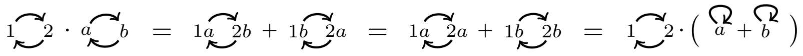
1 Introduction

A deterministic, finite, discrete-time dynamical system is a function from a finite set (of states) to itself. Equivalently, this is a *functional digraph*, that is, a finite directed graph where each vertex has a unique out-neighbor. In addition to being ubiquitous objects in discrete mathematics, such systems have many real-life applications [5]. In this paper, we consider functional digraphs up to isomorphism. An isomorphism class then corresponds to an unlabelled functional digraph, and we write $A = B$ to mean that A is isomorphic to B.

There are two natural algebraic operations to obtain larger systems from smaller ones. Given two functional digraphs A and B, the addition $A + B$ is the disjoint union of A and B, while the multiplication AB is the *direct product* of A and B: the vertex set of AB is the Cartesian product of the vertex set of A and the vertex set of B, and the out-neighbor of (x, y) in AB is (x', y') where

x' is the out-neighbor of x in A and y' is the out-neighbor of vertex y in B. Hence AB describes the parallel evolution of the dynamics described by A and B. Endowed with these two operations, the set of functional digraphs forms a semiring, first introduced in [1].

The multiplicative structure of this semiring has been studied in [4,6,2], and several important problems are highlighted in [6]. A fundamental one is the *division problem*: given two functional digraphs A and B, does A divide B, that is, does there exist a solution X to the equation $AX = B$. This problem is trivially in NP, and nothing else is know in the general case (there are better upper-bounds under some conditions, as explained above). From an applicative point of view, we can see B has an observed dynamical system, and the division problem then ask if B corresponds to the parallel evolution of A and an unknown part X; a positive answer then allows a potentially useful decomposition of B. We stress that, if a solution X exists, then it is not necessarily unique. For instance, denoting C_ℓ the (directed) cycle of length ℓ, the equation $C_2 \cdot X = C_2 + C_2$ has exactly two (non-isomorphic) solutions X, which are $X = C_2$ and $X = C_1 + C_1$; see (1). Given A and B, it is thus interesting not only to decide if A divides B, but to compute the number of solutions.



(1)

A functional digraph A contains two parts: the *cyclic part*, which is the collection of the cycles of A (these cycles are disjoint and thus form a permutation), and the *transient part*, which is obtained by deleting the cycles, and which is a disjoint union of out-trees. From a dynamical point of view, the cyclic part describes the asymptotic behavior. In this paper, we focus one this part: we study the complexity of the division problem when B is a *permutation*, that is, a disjoint union of directed cycles. This restriction has already been considered in [2] as an important step for solving polynomial equations over functional digraphs. On the other side, [6] gives a polynomial algorithm to decide if A divides B when B is a dendron, that is, B contains a unique cycle, of length 1, so that B consists of an out-tree plus a loop on the root. This result should be very useful to treat the transient part in the division problem. One may hope that efficient algorithms for the cyclic and transient parts could be combined to obtain an efficient algorithm for the general case.

If B is a permutation, there is a simple sub-exponential algorithm that computes all the solutions X of $AX = B$. It works as follows. Let $|A|$ and $|B|$ be the number of vertices in A and B, respectively ($|A|+|B|$ is the size of the instance). If $AX = B$ then A, X are permutations, and X has $n = |B|/|A|$ vertices. Now remark that isomorphic classes of permutations with n vertices are in bijection with partitions of n: a permutation with n vertices is completely described, up to isomorphism, by the sequence of the length of its cycles, which form a partition of n; and conversely, the parts of a partition of n describe the lengths of the cycles of a permutation with n vertices. For instance, $C_1 + C_3$ corresponds to the partition $1+3$ of 4. So to compute the solutions, we can enumerate the partitions of n, and check for each if the corresponding permutation X satisfies $AX = B$.

This gives a sub-exponential algorithm: partitions of n can be enumerated with polynomial delay and there are at most $e^{O(\sqrt{n})}$ such partitions, the total running time.

Frustratingly, we were not able to find a faster algorithm, say running in $e^{n^{o(1)}}$, to decide if A divides B. That a polynomial algorithm exists is an interesting open problem, and [3] gives a positive answer under the condition that, in A or B, all the cycles have the same length. Here we give a polynomial algorithm that computes the number of solutions when A is fixed. The precise statement, Theorem 1 below, involves some definitions. The *support* of a permutation A is the set $L(A)$ of positive integers ℓ such that A contains C_ℓ. Given $N \subseteq \mathbb{N}$, $\mathrm{lcm}N$ is the least common multiple of the integers in N, and $\mathrm{div}(n)$ is the number of divisors of n.

Theorem 1. *There is an algorithm that, given two non-empty permutations A, B, computes the number of non-isomorphic permutations X satisfying $AX = B$ with time complexity*

$$O\left(|A||B|^2\left(\frac{|B|}{|A|}\right)^{\mathrm{div}(\mathrm{lcm}L(A))}\right). \tag{2}$$

For the proof, we introduce two operations on an instance (A, B) that we hope to be useful for further progress on the division problem. The first partitions B into $B = B_1 + B_2$ so that any solution of (A, B) is obtained by adding a solution to (A, B_1) with a solution to (A, B_2). The second reduces (A, B) into a smaller instance (A', B') so that any solution of (A, B) is obtained by multiplying the length of the cycles of a solution of (A', B') by some constant d. Repeating as much as possible these operations, we obtain a decomposition of (A, B) into few smaller instances, which can be quickly solved with a brute force approach. The solutions of (A, B) are then obtained with a simple combination of the solutions of the instances of its decomposition. This is described in Lemma 2, the main result, which easily implies Theorem 1.

The rest of the paper is devoted to the proof of Theorem 1 following this decomposition method. Before going on, let us conclude this introduction by mentioning that a natural next step concerning the division problem should consist in proving that, for every fixed functional digraph A (with possibly a non-empty transient part), there is a polynomial time algorithm that, given a functional digraph B, decides if A divides B.

2 Preliminaries

Given $N \subseteq \mathbb{N}$, we denote by $\mathrm{lcm}N$ and $\gcd N$ the least common multiple and the greatest common divisor of the integers in N, respectively. For $n, m \in \mathbb{N}$, we set $n \vee m = \mathrm{lcm}\{n, m\}$, and for $N, M \subseteq \mathbb{N}$ we set $N \vee M = \{n \vee m \mid n \in N, m \in M\}$. We denote by $\mathrm{Div}(n)$ the set of divisors of n and set $\mathrm{div}(n) = |\mathrm{Div}(n)|$. We set $\mathrm{Div}(N) = \cup_{n\in N}\mathrm{Div}(n)$ and $\mathrm{div}(N) = |\mathrm{Div}(N)|$. For a positive integer p, we write $p \mid N$ to means that $p \mid n$ for all $n \in N$. We set $pN = \{pn \mid n \in N\}$ and we use the rather unusual notation $N/p = \{n/p \mid n \in N, p \mid n\}$.

The unlabelled functional digraph that consists of n cycles of length ℓ is denoted by nC_ℓ. Given a permutation A and an integer ℓ, the number of cycles of length ℓ in A is denoted by $A(\ell)$. Thus $A = \sum_{\ell \geq 1} A(\ell)C_\ell$ and $|A| = \sum_{\ell \geq 1} \ell A(\ell)$, and the support $L(A)$ is the set of ℓ such that $A(\ell) > 0$. One easily check that $C_a C_b = (ab/(a \vee b))C_{a \vee b}$. One can then prove (see [2]) that the product AX of two permutations A and X satisfies: for all $\ell \geq 1$,

$$AX(\ell) = \frac{1}{\ell} \sum_{\substack{a,x \in \mathbb{N} \\ a \vee x = \ell}} aA(a)xX(x). \tag{3}$$

Let A, B be non-empty permutations. We call (A, B) an *instance*, and its *size* is $|A| + |B|$. Recall that a solution of the instance (A, B) is a permutation X such that $AX = B$ and that $|X| = |B|/|A|$ for every solution X. We denote by $\mathrm{Sol}(A, B)$ the set of non-isomorphic solutions, and $\mathrm{sol}(A, B) = |\mathrm{Sol}(A, B)|$. It is important to note that, by (3), for every permutations A, X we have

$$L(AX) = L(A) \vee L(X). \tag{4}$$

3 Support of an instance

Let us define the *support* of an instance (A, B) as

$$L(A, B) = \{\ell \in \mathbb{N} \mid L(A) \vee \ell \subseteq L(B)\}.$$

So $L(A, B) \subseteq \mathrm{Div}(L(B))$ and

$$L(A) \vee L(A, B) \subseteq L(B). \tag{5}$$

This set $L(A, B)$ is interesting since it bounds the support of any solution:

$$\forall X \in \mathrm{Sol}(A, B), \qquad L(X) \subseteq L(A, B). \tag{6}$$

Indeed, if $AX = B$ then by (4) we have $L(A) \vee L(X) = L(B)$ and thus $L(X) \subseteq L(A, B)$. Since $L(A, B)$ bounds the support of any solution, and since any solution has obviously at most $|B|/|A|$ cycles, we obtain the following result using a brute force approach.

Lemma 1. *There is an algorithm that, given two non-empty permutations A, B, computes $\mathrm{Sol}(A, B)$ with time complexity $O(|A||B|(|B|/|A|)^{|L(A,B)|})$.*

Proof. Suppose that $n = |B|/|A|$ is an integer, since otherwise there is no solution. Suppose that $X \neq nC_1$ is a solution, so $X(1) < n$. For every $\ell \geq 1$, we have $\ell X(\ell) \leq n$, hence $X(\ell) < n$, and if $\ell \notin L(A, B)$ then $X(\ell) = 0$ by (6). Consequently, X corresponds to a function from $L(A, B)$ to $\{0, \ldots, n-1\}$. Hence, to find all the solutions: we enumerate the $n^{|L(A,B)|}$ such functions; we check for each, in $O(|A||B|)$, if it is a solution; and we then check if nC_1 is a solution. □

Another interesting point is that the support of an instance gives an easy to check necessary condition for the existence of a solution. Let us say that an instance (A,B) is *consistent* if $L(A) \vee L(A,B) = L(B)$. Then non-consistent instances have no solution. Indeed, if X is a solution to (A,B) then (A,B) is consistent since

$$L(B) = L(AX) \overset{(4)}{=} L(A) \vee L(X) \overset{(6)}{\subseteq} L(A) \vee L(A,B) \overset{(5)}{\subseteq} L(B).$$

Example 1. Let (A,B) be an instance with $L(A) = \{6\}$ and $L(B) = \{6,12\}$. Then $L(A,B) = \{1,2,3,4,6,12\}$ and thus (A,B) is consistent. Let (A,B) be an instance with $L(A) = \{6\}$ and $L(B) = \{5,6\}$. Then $L(A,B) = \{1,2,3,6\}$ and thus (A,B) is not consistent since $L(A) \vee L(A,B) = \{6\}$.

4 Decomposition lemma

Let us say that an instance (A,B) is *basic* if $L(B) \subseteq \mathrm{Div}(\mathrm{lcm} L(A))$; this is equivalent to say that for any prime power p^α dividing some $b \in L(B)$, there exists $a \in L(A)$ such that p^α divides a. By the previous lemma, Theorem 1 holds for every basic instance (A,B) since

$$L(A,B) \subseteq \mathrm{Div}(L(B)) \subseteq \mathrm{Div}(\mathrm{lcm} L(A)).$$

The key point is that any instance (A,B) can be decomposed into at most $|B|$ basic instances, with smaller sizes, in such a way that the solutions of (A,B) can be easily reconstructed from that of the basic instances. The precise statement, Lemma 2 below, needs some definitions.

The *cycle length multiplication* of A by p, denoted $A \otimes p$, is the permutation obtained from A by multiplying by p the length of every cycle in A; in other words: for all $a \geq 1$, we have $(A \otimes p)(a) = A(a/p)$ if $p \mid a$ and $(A \otimes p)(a) = 0$ otherwise. For instance, $(2C_1 + 3C_2 + 5C_3) \otimes 3 = 2C_3 + 3C_6 + 5C_9$. Given two sets of permutations $\mathcal{A}$ and $\mathcal{B}$ we set

$$\mathcal{A} + \mathcal{B} = \{A + B \mid A \in \mathcal{A}, B \in \mathcal{B}\}, \qquad \mathcal{A} \otimes p = \{A \otimes p \mid A \in \mathcal{A}\}.$$

Lemma 2. *There is an algorithm that, given a consistent instance (A,B), computes in $O(|A||B|^2)$ a list of $k \leq |B|$ basic instances $(A_1,B_2),\dots,(A_k,B_k)$ and positive integers $p_1,\dots,p_k$ such that: $|A_i| = |A|$ and $\mathrm{lcm} L(A_i) \mid \mathrm{lcm} L(A)$ for all $1 \leq i \leq k$, $|B_1| + \dots + |B_k| \leq |B|$, and*

$$\mathrm{Sol}(A,B) = (\mathrm{Sol}(A_1,B_1) \otimes p_1) + \dots + (\mathrm{Sol}(A_k,B_k) \otimes p_k).$$

Theorem 1 is an easy consequence of Lemma 2.

Proof of Theorem 1 assuming Lemma 2. The algorithm is as follows. First we check if (A,B) is consistent; this is done in $O(|A||B|)$. If not, then (A,B) has no solution and we output 0. Otherwise, we compute in $O(|A||B|^2)$ the $k \leq |B|$

basic instances (A_i, B_i) as in Lemma 2. Then, for all $1 \leq i \leq k$, we use the algorithm of Lemma 1 to compute in $O(|A_i||B_i|(|B_i|/|A_i|)^{|L(A_i,B_i)|})$ the number s_i of solutions of (A_i, B_i). Finally, we output the product $s_1 \cdots s_k$, which is correct by Lemma 2. Since (A_i, B_i) is basic and $\mathrm{lcm}L(A_i)$ divides $\mathrm{lcm}L(A)$, we have $L(A_i, B_i) \subseteq \mathrm{Div}(\mathrm{lcm}L(A))$. Since $|A_i| = |A|$ and $|B_i| \leq |B|$, we deduce that the computation of each s_i is done in $O(|A||B|(|B|/|A|)^{\mathrm{div}(\mathrm{lcm}L(A))})$, and we obtain the running time (2) since $k \leq |B|$. □

The rest of the paper is devoted to the proof of Lemma 2.

5 Instance partitions

Let A be a permutation, and $L \subseteq \mathbb{N}$. We denote by $A[L]$ the permutations obtained from A by removing every cycle of A whose length is not in L: for all $a \geq 1$, $A[L](a) = A(a)$ if $a \in L$ and $A[L](a) = 0$ otherwise. Here is a simple sufficient condition for an instance (A, B) to be decomposable into two independent instances (when we consider partitions, parts are always non-empty).

Lemma 3. *Let (A, B) be a consistent instance. Let L_1, L_2 be a partition of $L(A, B)$, and let $B_i = B[L(A) \vee L_i]$ for $i = 1, 2$. Suppose that $L(B_1) \cap L(B_2) = \emptyset$. Then $B = B_1 + B_2$. Furthermore, (A, B_1) and (A, B_2) are consistent, and*

$$\mathrm{Sol}(A, B) = \mathrm{Sol}(A, B_1) + \mathrm{Sol}(A, B_2). \tag{7}$$

Proof. We deduce from (5) that $L(B_1) = L(A) \vee L_1$ and $L(B_2) = L(A) \vee L_2$, and from that we deduce that (A, B_1) and (A, B_2) are consistent. Since $L_1 \cup L_2 = L(A, B)$ and (A, B) is consistent, we have

$$L(B_1) \cup L(B_2) = (L(A) \vee L_1) \cup (L(A) \vee L_2) = L(A) \vee L(A, B) = L(B).$$

Hence, $L(B_1), L(B_2)$ is a partition of $L(B)$ and thus $B = B_1 + B_2$.

It remains to prove (7). If X_1, X_2 are solutions of $(A, B_1), (A, B_2)$ then

$$A(X_1 + X_2) = AX_1 + AX_2 = B_1 + B_2 = B,$$

thus $X = X_1 + X_2$ is a solution of (A, B). Conversely, let X be a solution of (A, B) and let us prove that $X = X_1 + X_2$ for some solutions X_1, X_2 of $(A, B_1), (A, B_2)$. Let $X_i = X[L_i \cap L(X)]$ for $i = 1, 2$. By (6) we have $L(X) \subseteq L(A, B)$. Hence, $L(X_1) \cup L(X_2) = L(X)$. Thus $X = X_1 + X_2$ and using (4) we obtain

$$\begin{aligned} L(B) = L(AX) &= L(A) \vee L(X) \\ &= (L(A) \vee L(X_1)) \cup (L(A) \vee L(X_2)) = L(AX_1) \cup L(AX_2). \end{aligned}$$

For $i = 1, 2$, we have $L(X_i) \subseteq L_i$ and thus, using (4),

$$L(AX_i) = L(A) \vee L(X_i) \subseteq L(A) \vee L_i = L(B_i).$$

Since $L(B_1), L(B_2)$ is a partition of $L(B)$, we deduce that $L(AX_i) = L(B_i)$ for $i = 1, 2$. Hence, to prove that X_i is a solution of (A, B_i), it is sufficient to prove that $AX_i(b) = B_i(b)$ for all $b \in L(B_i)$. So let $b \in L(B_i)$. For every $a \in L(A)$ and $x \in L(X)$ with $a \vee x = b$ we have $x \in L_i$ and thus $x \in L(X_i)$. Consequently,

$$\begin{aligned} AX_i(b) = \frac{1}{b} \sum_{\substack{a\in L(A)\\ x\in L(X_i)\\ a\vee x=b}} aA(a)xX_i(x) &= \frac{1}{b} \sum_{\substack{a\in L(A)\\ x\in L(X_i)\\ a\vee x=b}} aA(a)xX(x) \\ &= \frac{1}{b} \sum_{\substack{a\in L(A)\\ x\in L(X)\\ a\vee x=b}} aA(a)xX(x) = B(b) = B_i(b). \end{aligned}$$

□

We now prove that a non-basic instance (A, B) with $\gcd L(A, B) = 1$ is decomposable; we will then prove that, in some sense, the condition on the gcd can be suppressed, leading to a decomposition of every non-basic instance. For a positive integer n, and a prime p, let $\nu_p(n)$ be the greatest integer α such that p^α divides n.

Lemma 4. *Let (A, B) be a non-basic consistent instance with $\gcd L(A, B) = 1$. Let $b \in L(B)$ and a prime p such that $\nu_p(b) > \nu_p(a)$ for all $a \in L(A)$ (these exist since (A, B) is not basic). Let L_1 be the set of $x \in L(A, B)$ with $\nu_p(x) = \nu_p(b)$, and $L_2 = L(A, B) \setminus L_1$. Let $B_i = [L(A) \vee L_i]$ for $i = 1, 2$. Then (A, B_1) and (A, B_2) are consistent instances such that $B = B_1 + B_2$ and*

$$\mathrm{Sol}(A, B) = \mathrm{Sol}(A, B_1) + \mathrm{Sol}(A, B_2).$$

Proof. Since (A, B) is consistent, there exists $a \in L(A)$ and $x \in L(A, B)$ such that $a \vee x = b$. Since $\nu_p(a) < \nu_p(b)$ we have $\nu_p(x) = \nu_p(b)$ and thus $x \in L_1$; so L_1 is not empty. Since $p \mid \gcd L_1$ and $\gcd L(A, B) = 1$, we have $L_1 \neq L(A, B)$ and thus L_2 is also non-empty.

Let $a \in L(A)$. For all $x \in L_1$, we have $\nu_p(a) < \nu_p(b) = \nu_p(x)$, thus $\nu_p(a \vee x) = \nu_p(b)$, and for all $y \in L_2$ we have $\nu_p(a), \nu_p(y) \neq \nu_p(b)$ thus $\nu_p(a \vee y) \neq \nu_p(b)$. Consequently, $L(A) \vee L_1$ is disjoint from $L(A) \vee L_2$. By Lemma 3 we have $B = B_1 + B_2$, and the instances (A, B_1) and (A, B_2) have the desired properties. □.

Example 2. Let $A = C_6$ and $B = 3C_6 + 8C_{12}$. Then (A, B) is consistent but not basic, and $\gcd L(A, B) = 1$ (see Ex. 1). Applying Lemma 4 with $b = 12$ and $p = 2$ we obtain $L_1 = \{4, 12\}$ and $L_2 = \{1, 2, 3, 6\}$, giving $B_1 = 8C_{12}$ and $B_2 = 3C_6$. Since the support of any solution X_1 of (A, B_1) is included in L_1,

$$\begin{aligned} C_6X_1 = 8C_{12} &\iff C_6(X_1(4)C_4 + X_1(12)C_{12}) = 8C_{12} \\ &\iff 2X_1(4)C_{12} + 6X_1(12)C_{12} = 8C_{12} \\ &\iff 2X_1(4) + 6X_1(12) = 8. \end{aligned}$$

Thus each solution X_1 corresponds to a partition of 8 with parts in $\{2,6\}$: these are $2+6$ and $2+2+2+2+2$, giving $X_1 = C_4 + C_{12}$ and $X_1 = 4C_4$. Proceeding similarly, since the support of any solutions X_2 of (A, B_2) is included in L_2, we have $C_6X_2 = 3C_6$ iff $X_2(1)+2X_2(2)+3X_2(3)+6X_2(6) = 3$. Thus each solution X_2 corresponds to a partition of 3 with parts in $\{1,2,3,6\}$: these are 3, $1+2$, and $1+1+1$, giving $X_2 = C_3$, $X_2 = C_1 + C_2$ and $X_2 = 3C_1$. By Lemma 4, we have $\mathrm{Sol}(A, B) = \mathrm{Sol}(A, B_1) + \mathrm{Sol}(A, B_2)$. Hence (A, B) has 6 solutions, obtained by adding a solution X_1 to (A, B_1) with a solution X_2 to (A, B_2).

6 Instance reduction

We say that an instance (A, B) is *compact* if $\gcd L(A, B) = 1$. This condition is used in Lemma 4 to decompose non-basic instances, but in this section we show that every instance can be reduced to an "equivalent" compact instance, which can then be decomposed. For this reduction, we need two operations.

The *cycle length division* of a permutation A by a positive integer p, denoted $A \oslash p$, is the permutation obtained from A by deleting every cycle whose length is not a multiple of p, and by dividing by p the length of the remaining cycles; in other words: for all $a \geq 1$, $(A \oslash p)(a) = A(pa)$. Note that $L(A \oslash p) = L(A)/p$ and if $p \mid L(A)$ then $L(A) = pL(A \oslash p)$. For instance,

$$(2C_1 + 3C_3 + 5C_4 + 7C_6) \oslash 3 = 3C_1 + 7C_2.$$

The cycle length division $\oslash$ is the inverse of the cycle length multiplication $\otimes$.

Lemma 5. *Let A be a permutation and let p be a positive integer. Then $(A \otimes p) \oslash p = A$, and if $p \mid L(A)$ then $(A \oslash p) \otimes p = A$.*

Proof. For all $a \geq 1$, we have $((A \otimes p) \oslash p)(a) = (A \otimes p)(pa) = A(a)$. Suppose that $p \mid L(A)$ and let $a \geq 1$. If $p \nmid a$ then $A(a) = 0$ and $((A \oslash p) \otimes p)(a) = 0$ (since $p \mid L((A \oslash p) \otimes p)$). If $p \mid a$ then $((A \oslash p) \otimes p)(a) = (A \oslash p)(a/p) = A(a)$. $\square$

The second operation is for the moment only defined when p is a prime; it will be extended to every positive integers later. The *contraction* of A by a prime p is the sum of cycle $A \boxslash p$ defined by: for all $a \geq 1$,

$$(A \boxslash p)(a) = \begin{cases} A(a) + pA(pa) & \text{if } p \nmid a \\ pA(pa) & \text{otherwise.} \end{cases} \tag{8}$$

This operation transforms each cycle of length pa into p cycles of length a (and thus keeps the number of vertices unchanged). Note that $L(A \boxslash p)$ is the set of integers a such that either $a \in L(A)$ and $p \nmid a$ or $pa \in L(A)$. For instance,

$$\begin{aligned}(2C_1 + 3C_3 + 5C_4 + 7C_6) \boxslash 3 &= 2C_1 + 9C_1 + 5C_4 + 21C_2 \\ &= 11C_1 + 5C_4 + 21C_2.\end{aligned}$$

Our interest for these two operations lies in the following property.

Lemma 6. *Let A, X be permutations. If $p \mid L(X)$ for some prime p, then*

$$(A \boxslash p)(X \oslash p) = (AX) \oslash p.$$

Proof. Suppose that $p \mid L(X)$ for some prime p, and let $A' = A \boxslash p$ and $X' = X \oslash p$. We have to prove that $A'X' = AX \oslash p$, that is, for all $\ell \geq 1$, $A'X'(\ell) = (AX \oslash p)(\ell) = AX(p\ell)$. Let us fix $\ell \geq 1$. We have

$$p\ell A'X'(\ell) = \sum_{\substack{a,x \\ a \vee x = \ell}} paA'(a)xX'(x) = \sum_{\substack{a,x \\ a \vee x = \ell}} aA'(a)pxX(px).$$

Denoting by Ω the of couples $(a,x) \in \mathbb{N}^2$ with $p \mid x$ and $a \vee \frac{x}{p} = \ell$, we obtain

$$p\ell A'X'(\ell) = \sum_{(a,x) \in \Omega} aA'(a)xX(x).$$

By splinting the sum according to the definition of A' we obtain

$$\begin{aligned} p\ell A'X'(\ell) &= \sum_{\substack{(a,x) \in \Omega \\ p \nmid a}} \big(aA(a) + paA(pa)\big)xX(x) + \sum_{\substack{(a,x) \in \Omega \\ p \mid a}} paA(pa)xX(x) \\ &= \sum_{\substack{(a,x) \in \Omega \\ p \nmid a}} aA(a)xX(x) + \sum_{\substack{(a,x) \in \Omega \\ p \nmid a}} paA(pa)xX(x) + \sum_{\substack{(a,x) \in \Omega \\ p \mid a}} paA(pa)xX(x). \end{aligned}$$

Denoting by Ω' the set of $(a,x) \in \mathbb{N}^2$ with $p \mid x$, $p \mid a$ and $\frac{a}{p} \vee \frac{x}{p} = \ell$, we obtain

$$p\ell A'X'(\ell) = \sum_{\substack{(a,x) \in \Omega \\ p \nmid a}} aA(a)xX(x) + \sum_{\substack{(a,x) \in \Omega' \\ p \nmid \frac{a}{p}}} aA(a)xX(x) + \sum_{\substack{(a,x) \in \Omega' \\ p \mid \frac{a}{p}}} aA(a)xX(x).$$

If $p \nmid a$ then $a \vee \frac{x}{p} = \ell$ iff $a \vee x = p\ell$; and $\frac{a}{p} \vee \frac{x}{p} = \ell$ iff $a \vee x = p\ell$. Consequently

$$p\ell A'X'(\ell) = \sum_{\substack{a,x \\ p \mid x \\ a \vee x = p\ell}} aA(a)xX(x).$$

Since $p \mid L(X)$, if $p \nmid x$ then $X(x) = 0$, so

$$p\ell A'X'(\ell) = \sum_{\substack{a,x \\ a \vee x = p\ell}} aA(a)xX(x) = p\ell AX(p\ell).$$

Thus $A'X'(\ell) = AX(p\ell)$ for all $\ell \geq 0$, as desired. □.

We obtain that every non compact instance can be reduced.

Lemma 7. *Let (A, B) be a consistent instance, and suppose $p \mid L(A, B)$ for some prime p. Then $(A \boxslash p, B \oslash p)$ is consistent with support $L(A, B)/p$, and*

$$\operatorname{Sol}(A, B) = \operatorname{Sol}(A \boxslash p, B \oslash p) \otimes p. \qquad (9)$$

Proof. Let $A' = A \boxslash p$ and $B' = B \oslash p$. We first prove (9). Let X be a solution of (A, B). By (6) we have $L(X) \subseteq L(A, B)$ and since $p \mid L(A, B)$ we have $p \mid L(X)$. Hence, by Lemma 6, $A'(X \oslash p) = AX \oslash p = B \oslash p = B'$, that is, $X \oslash p$ is a solution of (A', B'). Since $p \mid L(X)$, by Lemma 5 we have $(X \oslash p) \otimes p = X$ and thus $X \in \mathrm{Sol}(A' B') \otimes p$.

We now prove the converse direction. Let X' be a solution of (A', B'), and let $X = X' \otimes p$. We have to prove that X is a solution of (A, B). By Lemma 5 we have $X \oslash p = X'$ thus $A'(X \oslash p) = B' = B \oslash p$. Since $p \mid L(X)$, by Lemma 6, we have $A'(X \oslash p) = (AX) \oslash p$. Thus $(AX) \oslash p = B \oslash p$. Since $p \mid L(A, B)$ and (A, B) is consistent, we have $p \mid L(B)$. Since $X = X' \otimes p$ we obviously have $p \mid L(X)$. Thus p divides $L(A) \vee L(X) = L(AX)$. Using Lemma 5 we obtain $AX = ((AX) \oslash p) \otimes p = (B \oslash p) \otimes p = B$. Thus X is a solution of (A, B). This proves (9).

We now prove that $L(A, B)/p \subseteq L(A', B')$. For that, we fix $x \in L(A, B)/p$, and we prove that $a \vee x$ is in $L(B')$ for any $a \in L(A')$. Indeed, if $pa \in L(A)$ then $pa \vee px = b$ for some $b \in L(B)$ and we deduce that $a \vee x = b/p \in L(B')$. If $pa \notin L(A)$, then $p \nmid a$ and $a \in L(A)$. Thus $a \vee px = b$ for some $b \in L(B)$ and since $p \nmid a$ we deduce that $a \vee x = b/p \in L(B')$.

We now prove the converse inclusion. For that, we fix $x \in L(A', B')$, and we prove that $a \vee px$ is in $L(B)$ for any $a \in L(A)$. Indeed, if $p \mid a$ then $a/p \in L(A')$ and thus $(a/p) \vee x = b$ for some $b \in L(B')$ so that $a \vee px = pb \in L(B)$. If $p \nmid a$ then $a \in L(A')$ and thus $a \vee x = b$ for some $b \in L(B')$, and since $p \nmid a$ we have $a \vee px = pb \in L(B)$.

We finally prove that (A', B') is consistent. By (5) we only have to prove that $L(B') \subseteq L(A') \vee L(A', B')$. Let $b \in L(B')$. Then $pb \in L(B)$ and since (A, B) is consistent, there is $a \in L(A)$ and $x \in L(A, B)$ with $a \vee x = pb$. Hence $x/p \in L(A', B')$. If $p \mid a$ then $(a/p) \vee (x/p) = b$ and we are done since $a/p \in L(A')$. If $p \nmid a$ then $a \vee (x/p) = b$ and we are done since $a \in L(A')$. □

Applying several times the previous lemma we obtain a compact "equivalent" instance. Let us first extend the contraction operation from primes to any positive integer, inductively as follows: $A \boxslash 1 = A$, if p is a prime then $A \boxslash p$ is defined as previously (see (8)), and if p is composite, we take the largest prime q that divides p and set

$$A \boxslash p = (A \boxslash p/q) \boxslash q.$$

Note that, for every positive integers p, q, we have

$$(A \otimes p) \otimes q = A \otimes pq, \quad (A \oslash p) \oslash q = A \oslash pq, \quad (A \boxslash p) \boxslash q = A \boxslash pq. \tag{10}$$

The first two equalities are obvious. The third results from the following easy to check commutativity property: $(A \boxslash p) \boxslash q = (A \boxslash q) \boxslash p$ when p and q are primes.

Lemma 8. *Let (A, B) be a consistent instance and $d = \gcd L(A, B)$. Then $(A \boxslash d, B \oslash d)$ is a compact consistent instance with support $L(A, B)/d$, and*

$$\mathrm{Sol}(A, B) = \mathrm{Sol}(A \boxslash d, B \oslash d) \otimes d.$$

Proof. Suppose that $d > 1$ since otherwise there is nothing to prove. Let us write d as the product of $k \geq 1$ primes, not necessarily distinct, say $d = p_1 p_2 \dots p_k$ with $p_1 \leq p_2 \leq \cdots \leq p_k$. Let $A_0 = A$, $B_0 = B$ and, for $1 \leq \ell \leq k$, let $A_\ell = A_{\ell-1} \boxslash p_\ell$ and $B_\ell = B_{\ell-1} \oslash p_\ell$. By Lemma 7, (A_ℓ, B_ℓ) is a consistent instance and $\mathrm{Sol}(A_{\ell-1}, B_{\ell-1}) = \mathrm{Sol}(A_\ell, B_\ell) \otimes p_\ell$. By (10) we have $A_k = A \boxslash d$, $B_k = B \boxslash d$, $L(A_k, B_k) = L(A, B)/d$ and $\mathrm{Sol}(A, B) = \mathrm{Sol}(A_k, B_k) \otimes d$. Since $L(A_k, B_k) = L(A, B)/d$, we obviously have $\gcd L(A_k, B_k) = 1$. □

Example 3. The support of $(C_6, 8C_{12})$ is $\{4, 12\}$. We have $C_6 \boxslash 4 = (C_6 \boxslash 2) \boxslash 2 = 2C_3 \boxslash 2 = 2C_3$, and $8C_{12} \oslash 4 = 8C_3$. By Lemma 8, $\mathrm{Sol}(C_6, 8C_{12}) = \mathrm{Sol}(2C_3, 8C_3) \otimes 4$. Since the support of $(2C_3, 8C_3)$ is $\{1, 3\}$,

$$\begin{aligned} 2C_3 X' = 8C_3 &\iff 2C_3(X'(1)C_1 + X'(3)C_3) = 8C_3 \\ &\iff 2X'(1)C_3 + 6X'(3)C_3 = 8C_3 \\ &\iff 2X'(1) + 6X'(3) = 8. \end{aligned}$$

Thus each solution X' corresponds to a partition of 8 with parts in $\{2, 6\}$: these are $2 + 6$ and $2 + 2 + 2 + 2 + 2$, giving $X' = C_1 + C_3$ and $X' = 4C_1$. Hence the solutions of $(C_6, 8C_{12})$ are $(C_1 + C_3) \otimes 4 = C_4 + C_{12}$ and $(4C_1) \otimes 4 = 4C_4$, which is consistent with the direct computation given in Ex. 2.

7 Proof of Lemma 2

We start with a definition. Let (A, B) be a consistent instance. A *decomposition* of (A, B) is a list $\mathcal{L}$ of triples (A_i, B_i, p_i), $1 \leq i \leq k$, such that

- (A_i, B_i) is a compact and consistent instance, and p_i is a positive integer,
- $|A_i| = |A|$ and $\mathrm{lcm} L(A_i)$ divides $\mathrm{lcm} L(A)$,
- $|B_1| + \cdots + |B_k| \leq |B|$,
- $\mathrm{Sol}(A, B) = (\mathrm{Sol}(A_1, B_1) \otimes p_1) + \cdots + (\mathrm{Sol}(A_k, B_k) \otimes p_k)$.

We call k the *length* of $\mathcal{L}$; note that by the third point, $k \leq |B|$. Furthermore, we say that $\mathcal{L}$ is *basic* if (A_i, B_i) is basic for all $1 \leq i \leq k$. We will prove that we can compute in $O(|A||B|^2)$ a basic decomposition, which clearly proves Lemma 2. For that we first prove that if (A, B) has a non-basic decomposition, we can obtain a longer decomposition by partitioning a non-basic instance (Lemma 4) and then contracting its parts (Lemma 8).

Lemma 9. *There is an algorithm that, given a consistent instance (A, B) and a non-basic decomposition $\mathcal{L}$ of (A, B) of length k, computes in $O(|A||B|)$ a decomposition $\mathcal{L}'$ of (A, B) of length $k + 1$.*

Proof. The algorithm is as follows. Let $(A_1, B_1, p_1), \dots, (A_k, B_k, p_k)$ be the triples of $\mathcal{L}$. Since $\mathcal{L}$ is not basic, we find in $O(|A||B|)$ a non-basic instance (A_i, B_i). Since (A_i, B_i) is compact and consistent, by Lemma 4, we can compute in $O(|A||B|)$ two consistent instances (A_i, B_{i1}) and (A_i, B_{i2}) such that $B_i = B_{i1} + B_{i2}$ and

$$\mathrm{Sol}(A_i, B_i) = \mathrm{Sol}(A_i, B_{i1}) + \mathrm{Sol}(A_i, B_{i2}).$$

For $j = 1, 2$, we compute in $O(|A||B|)$ the integer $p_{ij} = \gcd L(A_i, B_{ij})$ and the permutations $A_{ij} = A_i \boxslash p_{ij}$ and $B'_{ij} = B_{ij} \oslash p_{ij}$. Finally, we output the list $\mathcal{L}'$ of length $k+1$ obtained from $\mathcal{L}$ by deleting (A_i, B_i, p_i) and adding $(A_{i1}, B'_{i1}, p_i p_{i1})$ and $(A_{i2}, B'_{i2}, p_i p_{i2})$. So the runnig time is $O(|A||B|)$.

Let us prove that $\mathcal{L}'$ is a decomposition. By Lemma 8, (A_{ij}, B'_{ij}) is compact and consistent. Furthermore, $|A_{ij}| = |A_i| = |A|$ and since A_{ij} is a contraction of A_i, each member of $L(A_{ij})$ divides some member of $L(A_i)$, thus $\mathrm{lcm} L(A_{ij})$ divides $\mathrm{lcm} L(A_i)$, which divides $\mathrm{lcm} L(A)$. Thus $\mathrm{lcm} L(A_{ij})$ divides $\mathrm{lcm} L(A)$. Next, since $|B'_{i1}| + |B'_{i2}| \leq |B_{i1}| + |B_{i2}| = |B_i|$, the third point of the definition of a decomposition is preserved. Finally, by Lemma 8, $\mathrm{Sol}(A_i, B_{ij}) = \mathrm{Sol}(A_{ij}, B'_{ij}) \otimes p_{ij}$. Consequently,

$$\begin{aligned} \mathrm{Sol}(A_i, B_i) \otimes p_i &= (\mathrm{Sol}(A_i, B_{i1}) \otimes p_i) + (\mathrm{Sol}(A_i, B_{i2}) \otimes p_i) \\ &= (\mathrm{Sol}(A_{i1}, B'_{i1}) \otimes p_i p_{i1}) + (\mathrm{Sol}(A_{i2}, B'_{i2}) \otimes p_i p_{i2}) \end{aligned}$$

and this proves that the last point of the definition of a decomposition is preserved. So $\mathcal{L}'$ is indeed a decomposition of (A, B).

Iterating the previous lemma, we get the following, which implies Lemma 2.

Lemma 10. *There is an algorithm that, given a consistent instance* (A, B)*, computes in* $O(|A||B|^2)$ *a basic decomposition of* (A, B)*.*

Proof. The algorithm constructs recursively a list $\mathcal{L}_1, \ldots, \mathcal{L}_{|B|}$ of decompositions of (A, B), where the length of $\mathcal{L}_r$ is at most r, and output $\mathcal{L}_{|B|}$. First we compute $\mathcal{L}_1 = \{(A \boxslash d, B \oslash d, d)\}$ where $d = \gcd L(A, B)$ in $O(|A||B|)$; by Lemma 8, $\mathcal{L}_1$ is a decomposition of length one. Now, suppose that the decomposition $\mathcal{L}_r$ of length $k \leq r < |B|$ has already been computed. If $\mathcal{L}_r$ is basic, we set $\mathcal{L}_{r+1} = \mathcal{L}_r$. Otherwise, using Lemma 9, we compute in $O(|A||B|)$ a decomposition $\mathcal{L}_{r+1}$ of length $k + 1$. Hence the running time is $O(|A||B|^2)$. It remains to prove that $\mathcal{L}_{|B|}$ is basic. If $\mathcal{L}_r = \mathcal{L}_{r+1}$ for some $r < |B|$ then $\mathcal{L}_r$ is basic and $\mathcal{L}_s = \mathcal{L}_r$ for all $r < s \leq |B|$ thus $\mathcal{L}_{|B|}$ is basic. Otherwise, $\mathcal{L}_1, \ldots, \mathcal{L}_{|B|}$ are all distinct thus the length of $L_{|B|}$ is $|B|$. If $L_{|B|}$ is not basic, by Lemma 9, (A, B) has a decomposition of length $|B| + 1$, a contradiction. Thus $\mathcal{L}_{|B|}$ is basic. □

Example 4. Let $A = C_6$ and $B = 3C_6 + 8C_{12}$. Combining Ex. 2 and 3, we get that the basic decomposition of (A, B) is $(2C_3, 8C_3, 4), (C_6, 3C_6, 1)$ and so $\mathrm{Sol}(A, B) = (\mathrm{Sol}(2C_3, 8C_3) \otimes 4) + \mathrm{Sol}(C_6, 3C_6)$.

Acknowledgments This work has been funded by the HORIZON-MSCA-2022-SE-01 project 101131549 "Application-driven Challenges for Automata Networks and Complex Systems (ACANCOS)".

References

1. Alberto Dennunzio, Valentina Dorigatti, Enrico Formenti, Luca Manzoni, and Antonio E Porreca. Polynomial equations over finite, discrete-time dynamical systems.

In *Cellular Automata: 13th International Conference on Cellular Automata for Research and Industry, ACRI 2018, Como, Italy, September 17–21, 2018, Proceedings 13*, pages 298–306. Springer, 2018.

2. Alberto Dennunzio, Enrico Formenti, Luciano Margara, and Sara Riva. An algorithmic pipeline for solving equations over discrete dynamical systems modelling hypothesis on real phenomena. *Journal of Computational Science*, 66:101932, 2023.
3. Alberto Dennunzio, Enrico Formenti, Luciano Margara, and Sara Riva. A note on solving basic equations over the semiring of functional digraphs. *arXiv preprint arXiv:2402.16923*, 2024.
4. Caroline Gaze-Maillot and Antonio E Porreca. Profiles of dynamical systems and their algebra. *arXiv preprint arXiv:2008.00843*, 2020.
5. AS Jarrah and R Laubenbacher. Finite dynamical systems: A mathematical framework for computer simulation. In *Mathematical Modeling, Simulation, Visualization and e-Learning*, pages 343–358. Springer, 2007.
6. Émile Naquin and Maximilien Gadouleau. Factorisation in the semiring of finite dynamical systems. *arXiv preprint arXiv:2210.11270*, 2022.