

An Algorithmic Pipeline for Solving Equations over Discrete Dynamical Systems Modelling Hypothesis on Real Phenomena

Alberto Dennunzio^a, Enrico Formenti^b, Luciano Margara^c, Sara Riva^b

^aDipartimento di Informatica, Sistemistica e Comunicazione, Università degli Studi di Milano-Bicocca, Viale Sarca 336/14, 20126 Milano, Italy

^bUniversité Côte d'Azur, CNRS, I3S, France

^cDepartment of Computer Science and Engineering, University of Bologna, Cesena Campus, Via Sacchi 3, Cesena, Italy

Abstract

This paper provides an algorithmic pipeline for studying the intrinsic structure of a finite discrete dynamical system (DDS) modelling an evolving phenomenon. Here, by intrinsic structure we mean, regarding the dynamics of the DDS under observation, the feature of resulting from the ‘cooperation’ of the dynamics of two or more smaller DDS. The intrinsic structure is described by an equation over DDS which represents a hypothesis over the phenomenon under observation. The pipeline allows solving such an equation, *i.e.*, validating the hypothesis over the phenomenon, as far the asymptotic behavior and the number of states of the DDS under observation are concerned. The results are about the soundness and completeness of the pipeline and they are obtained by exploiting the algebraic setting for DDS introduced in [10].

Keywords: discrete modelling, finite discrete dynamical systems, hypothesis on phenomena

1. Introduction

Finite Discrete Dynamical Systems (DDS for short) are useful tools that have been used since at least the seventies of the last century for modelling many evolving phenomena, especially those rising from complex systems, that can go through a finite number of states. They can be found in many disciplines ranging for instance from biology to chemistry, stepping through computer science, physics, economics, sociology, *etc.*. Boolean automata networks, genetic regulations networks, and metabolic networks are just a few examples of DDS used in bioinformatics [6, 16, 17, 3, 9]. Cellular Automata with a finite number of cells and a finite alphabet are other examples of DDS exploited in a wide range of scientific domains for modelling complex phenomena [1, 2, 13, 14, 7].

When studying an evolving phenomenon modelled by a DDS, an important task is describing its dynamical behavior from experimental data. Namely, one aims at providing a finer structure of the observed dynamics, or, in other terms, answering the following question which the central issue of this paper:

Question 1. *Is the dynamics that we observe (from experimental data for instance) the action of a single basic system or does it come from the cooperation between two or more simpler systems?*

In this sense, a DSS can be viewed as a complex object with a certain intrinsic structure, *i.e.*, the feature of resulting from cooperating basic components. It is crucial, of course, to precise the meaning of the word ‘cooperation’ in Question 1. In [10], two forms of cooperation have been devised. The additive form, denoted by $+$, in which two DDS with independent dynamics provide together the observed system and the product one, denoted by \cdot , in which the observed system results from the joint parallel action of two DDS. The formalisation of these concepts leads to endow the set of all DDS with the algebraic structure of commutative semiring [10]. In this way, to face Question 1 it is quite natural consider multivariate monomials of the type $a \cdot x^w$ to represent

Email addresses: alberto.dennunzio@unimib.it (Alberto Dennunzio), enrico.formenti@unice.fr (Enrico Formenti), luciano.margara@unibo.it (Luciano Margara), sara.riva@univ-cotedazur.fr (Sara Riva)

a hypothesis about a finer structure of a given DDS. Here, considering $a \cdot x^w$ means that in the first place the observed DDS is supposed to result from the joint parallel action of a known DDS, *i.e.*, the coefficient a , and w copies of some yet unknown DDS x . Following this new point of view, a polynomial $P(x_1, \dots, x_v)$ is hence a more complex and realistic hypothesis on the observed DDS b and then Question 1 can be rephrased as:

Question 2. *Does the dynamics that we observe result from several independent smaller systems, each of them having dynamics determined by the joint parallel action of a known part and an unknown part to be computed? In other words, does the equation $P(x_1, \dots, x_v) = b$ have a solution? If any, what are its solutions?*

We stress that answering this question is always possible since the constant right-hand side bounds the space of admissible solutions. However, it might be highly non trivial, as illustrated in [10] for classes of more general polynomial equations $P(x_1, \dots, x_v) = Q(x_1, \dots, x_v)$, although DDS have very simple dynamics. Indeed, since the set of states is finite, all their points are ultimately periodic and then any system ultimately evolves towards one of its cycle sets, each of them consisting of periodic points and called attractor, that together describe the so-called *asymptotic behavior* of the system.

Since we aim at proposing a solid and effective tool to be used in applications that answers Question 2, in this paper we start to focus our attention on equations involving a multivariate polynomial without products of distinct variables. From now on, when no confusion is possible, we will always refer to that type of equations.

Our idea for solving the equation $P(x_1, \dots, x_v) = b$ is based on three abstractions of such an equation: the *c*-abstraction, the *a*-abstraction and the *t*-abstraction. By means of the abstractions, potential solutions of the considered equation are filtered out. Indeed, each abstraction provides the DDSs with a specific property induced by the equation. Namely, the *c*-abstraction allows computing all DDS that satisfy the condition on the cardinality of the state set induced by the original equation. The *a*-abstraction provides all DDS having a set of attractors that satisfies the equation obtained by the original one when constants and variables are restricted to their asymptotic behavior. Finally, the *t*-abstraction is similar to the *a*-abstraction but it focuses on the transient behaviour, *i.e.*, on the states that do not belong to any attractor. We stress that the set of the solutions of the equation $P(x_1, \dots, x_v) = b$ just turns out to be the intersection of the sets of DDSs selected by these three abstractions. For this reason, the enumeration of all solutions of each abstraction is needed to reach the goal.

As a first important step, in this paper we consider the *c*- and *a*-abstractions and, as results, we provide two methods solving the corresponding abstraction equations, leaving the *t*-abstraction for a future work. Let us explain their relevance. First of all, as we will see, solving the only *a*-abstraction equation requires a non trivial pipeline (*i.e.*, a sequence of processes with the output of one process being the input of the next one), including the computation of the w -th root of the asymptotic behavior of a DDS, that is a necessary intermediate step. Furthermore, the results allow understanding the structure of the asymptotic behavior of a phenomenon and it is well-known that this is very important in applicative scenarios.

Both the procedures make use of *Multi-valued Decision Diagrams (MDD)* ([5, 8, 4]) suitably defined according to our settings. Actually, they have been applied in several domains for representing formal objects in a compressed form. Their advantage is that they perform many operations without decompressing information. In this work, MDD are exploited to provide in an efficient way the needed solutions of all the further equations and systems derived from the abstractions, especially the *a*-abstraction. Moreover, they allow efficiently performing some important operations that are required by our procedures as for instance the product of solutions and the intersection of sets of them.

The paper is structured as follows. Next section introduces the background on DDS and the their semiring. The *c*- and the *a*-abstraction equations are dealt with in Section 3 and Section 4, respectively. Section 5 illustrates by a full worked out example how the solutions of the two abstraction equation can be combined. In the last section we draw our conclusions and some perspectives.

2. Background and basic facts

A *Discrete Dynamical System (DDS)* is a pair (\mathcal{X}, f) where \mathcal{X} is a finite set of states and $f : \mathcal{X} \rightarrow \mathcal{X}$ is a function called *next state map*. Any DDS (\mathcal{X}, f) can be identified with the directed graph $G = (V, E)$, called *dynamics graph*, where $V = \mathcal{X}$ and $E = \{(v, f(v)) \mid v \in V\}$ is the graph of f .

Let S be a DDS (\mathcal{X}, f) and let G be its dynamics graph. If \mathcal{Y} is any subset of \mathcal{X} such that $f(\mathcal{Y}) \subseteq \mathcal{Y}$, then the DDS $(\mathcal{Y}, f|_{\mathcal{Y}})$ is said to be the *dynamical subsystem* of (\mathcal{X}, f) induced by \mathcal{Y} (here, $f|_{\mathcal{Y}}$ means the restriction of f to \mathcal{Y}). Clearly, the dynamics graph of $(\mathcal{Y}, f|_{\mathcal{Y}})$ is nothing but the subgraph of G induced by \mathcal{Y} . A state $v \in \mathcal{X}$ is a *periodic point* of S if there exists an integer $p > 0$ such that $f^p(v) = v$. The smallest p with the previous property is called *period* of v . If $p = 1$, the state v is simply a *fixed point*. A *cycle* (of length p) of S is any set $\mathcal{C} = \{v, f(v), \dots, f^{p-1}(v)\}$ where $v \in \mathcal{X}$ is a periodic point of period p . Clearly, the set \mathcal{P} of all the periodic points of S can be viewed as union of disjoint cycles. Moreover, both $(\mathcal{C}, f|_{\mathcal{C}})$ and $(\mathcal{P}, f|_{\mathcal{P}})$ are dynamical subsystems of S and their dynamics graphs just consist of one among, resp., all, the strongly connected components of G . In the sequel, we will identify \mathcal{C} and \mathcal{P} with the DDS $(\mathcal{C}, f|_{\mathcal{C}})$ and $(\mathcal{P}, f|_{\mathcal{P}})$ (and then with their dynamics graphs too), respectively.

Two DDS are *isomorphic* if their dynamics graph are so in the usual sense of graph theory. When this happens, the systems are indistinguishable from the dynamical point of view. In particular, periodic points and cycles of a system are in one-to-one correspondence with periodic points and cycles of the other system. Therefore, the dynamical subsystems induced by them in the respective DDS are isomorphic too.

Recall that the disjoint union of two sets \mathcal{X}_1 and \mathcal{X}_2 is the set $\mathcal{X}_1 \sqcup \mathcal{X}_2 = (\mathcal{X}_1 \times \{0\}) \cup (\mathcal{X}_2 \times \{1\})$. In [10], an abstract algebraic setting for DDS was introduced. In particular, the following operations over the set of DDS were defined where the notion of disjoint union is extended to functions.

Definition 1 (Sum and product of DDS). *The sum $(\mathcal{X}_1, f_1) + (\mathcal{X}_2, f_2)$ and the product $(\mathcal{X}_1, f_1) \cdot (\mathcal{X}_2, f_2)$ of any two DDS (\mathcal{X}_1, f_1) and (\mathcal{X}_2, f_2) are the DDS $(\mathcal{X}_1 \sqcup \mathcal{X}_2, f_1 \sqcup f_2)$ and $(\mathcal{X}_1 \times \mathcal{X}_2, f_1 \times f_2)$, respectively, where the function $f_1 \sqcup f_2 : \mathcal{X}_1 \sqcup \mathcal{X}_2 \rightarrow \mathcal{X}_1 \sqcup \mathcal{X}_2$ is defined as:*

$$\forall (v, i) \in \mathcal{X}_1 \sqcup \mathcal{X}_2 \quad (f_1 \sqcup f_2)(v, i) = \begin{cases} (f_1(v), i) & \text{if } v \in \mathcal{X}_1 \wedge i = 0 \\ (f_2(v), i) & \text{if } v \in \mathcal{X}_2 \wedge i = 1 \end{cases},$$

while $f_1 \times f_2 : \mathcal{X}_1 \times \mathcal{X}_2 \rightarrow \mathcal{X}_1 \times \mathcal{X}_2$ is the standard product of functions defined as $\forall (v_1, v_2) \in \mathcal{X}_1 \times \mathcal{X}_2, (f_1 \times f_2)(v_1, v_2) = (f_1(v_1), f_2(v_2))$ (also called *direct product* in the graph literature).

It is not difficult to see that the set of all DDS equipped with the sum and product operations turns out to be a semiring R in which both the operations are commutative (up to an isomorphism). In the sequel, the symbols 0 and 1 stand for their neutral elements. Moreover, for any natural $k > 0$ and any DDS S , the sum $S + \dots + S = \sum^k S$ and the product $S \cdot \dots \cdot S = \prod^k S$ of k copies of S will be naturally denoted by kS and S^k , respectively. In this way, we can state the following proposition which is nothing but the counterpart in our setting of the well-known standard multinomial theorem.

Proposition 1. *For any positive naturals w, l , and any DDS S_1, \dots, S_l it holds that*

$$(S_1 + \dots + S_l)^w = \sum_{\substack{k_1 + \dots + k_l = w \\ 0 \leq k_1, \dots, k_l \leq w}} \binom{w}{k_1, \dots, k_l} \prod_{t=1}^l S_t^{k_t}.$$

Now, consider the semiring $R[x_1, x_2, \dots, x_{\mathcal{V}}]$ of polynomials over R in the variables $x_1, x_2, \dots, x_{\mathcal{V}}$, naturally induced by R . Polynomial equations of the following form model hypotheses about a certain dynamics deduced from experimental data:

$$a_1 \cdot x_1^{w_1} + a_2 \cdot x_2^{w_2} + \dots + a_m \cdot x_m^{w_m} = b \quad (1)$$

The known term b is the DDS deduced from experimental data. The coefficients a_z (with $z \in \{1, \dots, m\}$) are hypothetical DDS that should cooperate to produce the observed dynamics b . Finding valid values for the unknowns in (1) provides a finer structure for b which can bring further knowledge about the observed phenomenon. We point out that Equation (1) might contain duplicated pairs (x_z, w_z) since it is the direct formulation of a hypothesis over that phenomenon. Indeed, the process of such a formulation might run into a $x_z^{w_z}$ which has been already considered but it has to be differently weighted.

3. Abstraction over the cardinality of the set of states (c-abstraction)

Given a polynomial equation over DDS, a natural abstraction concerns the number of states of the DDS involved in it. Performing such an abstraction leads to new equation in which the coefficients of the polynomial, the variables, and the constant term become those natural numbers corresponding to the cardinalities of the state sets of the DDSs involved in the original equation.

Definition 2 (c-abstraction). *The c-abstraction of a DDS S is the cardinality of its set of states. With an abuse of notation, the c-abstraction of S is denoted by $|S|$.*

The following lemma links c-abstractions with the operations over DDS.

Lemma 1 ([10]). *For any pair of DDS S_1 and S_2 , it holds that $|S_1 + S_2| = |S_1| + |S_2|$ and $|S_1 \cdot S_2| = |S_1| \cdot |S_2|$.*

Using the notion of c-abstraction and the previous lemma, Equation (1) turns into the following *c-abstraction equation*:

$$|a_1| \cdot |x_1|^{w_1} + |a_2| \cdot |x_2|^{w_2} + \dots + |a_m| \cdot |x_m|^{w_m} = |b| . \quad (2)$$

To reach our overall goal, we need to enumerate all solutions of Equation (2). In this way, all possible cardinalities of the state sets of the unknown DDSs from the original Equation (1) will be identified. To perform that task, we proceed as follows. First of all, we present the enumeration problem from a combinatorial point of view. Then, we will provide an algorithmic approach allowing the enumeration of the solutions of Equation (2) in an efficient way.

Let us consider the case with just one monomial (i.e., $m = 1$) corresponding to a simpler equation of form $|a| \cdot |x|^w = |b|$ (*basic case*). It is clear that

- if $w = 0$, then $|x|^w$ is the c-abstraction of a DDS consisting of a unique cycle of length one (a fixed point) and $|a| = |b|$, while the equation is impossible, otherwise;
- if $w \neq 0$, the equation admits a (unique) solution iff $\sqrt[w]{|b|/|a|}$ is an integer number.

Given now an equation with $m > 1$ monomials, it is clear that each state of the DDS b must come from one of them. Thus, we have to consider the all the ways of arranging $|b|$ states among m monomials. Since there can be arrangements in which not all the monomials are involved, by the Stars and Bars method (see [12], for instance), the number of such arrangements is $\binom{|b|+m-1}{m-1}$. Moreover, any arrangement consisting of b_1, \dots, b_m states from b in the respective monomials, i.e., any weak composition b_1, \dots, b_m of $|b|$ into exactly m parts, gives rise to the following system

$$\begin{cases} |a_1| \cdot |x_1|^{w_1} &= b_1 \\ |a_2| \cdot |x_2|^{w_2} &= b_2 \\ &\vdots \\ |a_m| \cdot |x_m|^{w_m} &= b_m \end{cases}, \quad (3)$$

where $\sum_{z=1}^m b_z = |b|$ and each equation falls into the basic case.

Therefore, we need an efficient method that solves all feasible Systems (3), i.e., those systems admitting a solution. Since any System (3) consists of equations that are all from the basic case and establishing whether each of them admits a solution is easy, the method can be designed in such a way that the space of possible solutions to be explored is reduced.

Due to the combinatorial nature of the problem, we provide a method based on a *Multi-valued Decision Diagrams* (MDD) to enumerate the solutions of a c-abstraction equation. Recall that an MDD is a rooted acyclic graph able to represent a multivalued function having a finite set as domain and the set $\{true, false\}$ as codomain. Both vertices and edge are labelled. In the structure, each level represents a variable, except for the final one with the true terminal node (called *tt*). The first level contains the root node (called *root*). A path from the *root* to the *tt*

node represents a valid set of variable assignments given by the labels of the edges of that path. We stress that there can be distinct vertices (possibly on the same level) with the same label. For a sake of simplicity, we will often define the specific MDDs under the unconventional assumption that vertices form a multiset. This abuse will allow us to identify vertexes with the values of their labels. For more on MDD, we redirect the interested reader to [5, 8, 4].

Consider any c-abstraction equation with m monomials and a number \mathcal{V} of distinct variables. We associate such an equation with an MDD (V, E, lab) in which there are \mathcal{V} levels (one for each variable) and one final level for the tt node. The vertices form the multiset $V = \sum_{i \in \{1, \dots, \mathcal{V}+1\}} V_i$ where $V_1 = \{root\}$, $V_{\mathcal{V}+1} = \{tt\}$, and for each level $i \in \{2, \dots, \mathcal{V}\}$ the set $V_i \subseteq \{0, \dots, |b|\}$ of the vertexes of the level i will be defined in the sequel. Indeed, the structure is built level by level. Moreover, for any node $\alpha \in V$, let $val(\alpha) = \alpha$ if $\alpha \neq root$ and $\alpha \neq tt$, while $val(root) = 0$ and $val(tt) = |b|$.

To define the edges outgoing from the vertexes of any level along with the corresponding labels and then the vertexes of the next level too, first of all we associate each level i with the inequality $\sum_{z=1}^m var(i, z) \cdot |a_z| \cdot |x_z|^{w_z} \leq |b|$, where $var(i, z) = 1$ if $|x_z|$ is the variable associated with the level i , 0 otherwise, and the set $D_i = \{d \in \mathbb{N} \mid \sum_{z=1}^m var(i, z) \cdot |a_z| \cdot |x_z|^{w_z} \leq |b| \text{ with } |x_z| = d\} \cup \{0\}$ of the labels of the edges outgoing from the vertexes of the level i . These labels represent the possible values for the variable corresponding to the level i .

Now, for each level $i \in \{1, \dots, \mathcal{V}-1\}$, for any vertex $\alpha \in V_i$ and any $\beta \in \{0, \dots, |b|\}$ it holds that $\beta \in V_{i+1}$ and $(\alpha, \beta) \in E$ iff there exists $d \in D_i$ such that

$$\beta = val(\alpha) + \sum_{z=1}^m var(i, z) \cdot |a_z| \cdot d^{w_z} \leq val(tt).$$

Similarly, regarding the level \mathcal{V} , for any vertex $\alpha \in V_{\mathcal{V}}$, it holds that $(\alpha, tt) \in E$ iff there exists $d \in D_i$ such that

$$val(tt) = val(\alpha) + \sum_{z=1}^m var(\mathcal{V}, z) \cdot |a_z| \cdot d^{w_z}.$$

In both cases the edge (α, β) is associated with the label $lab((\alpha, \beta)) = d$. In this way, the labelling function $lab: E \rightarrow \bigcup_{i \in \{1, \dots, \mathcal{V}\}} D_i$ has been defined too.

The value $val(\alpha)$ associated with any node α represents the amount of states obtained from a partial set of variable assignments, i.e., a set of assignments involving the variables until the level the node α belongs to, each of them corresponding to a path from $root$ to α . The c-abstraction equation admits no solution if there is no path from $root$ to tt on the associated MDD.

Finally, the MDD is reduced by performing a pReduction i.e. a procedure that merges equivalent nodes (on the same layer) and delete all nodes (and the corresponding edges) which are not on a path from $root$ to tt [15].

Example 1. Consider the following equation:

$$2 \cdot |x_3| + 5 \cdot |x_1|^2 + 4 \cdot |x_2| + 4 \cdot |x_1|^4 + 4 \cdot |x_3|^2 = 593.$$

Hence, there are $\binom{593+5-1}{5-1} = 5.239776465 \times 10^9$ way of arranging $|b| = 593$ states among $m = 5$ monomials. However, not all of them give rise to solutions of that equation. According to the definition, the resulting reduced MDD is illustrated in Figure 1. The first level of the structure represents the possible values for the variable $|x_1|$. In the second one, the red edges along with the corresponding label represent the possible values for $|x_2|$, in the case $|x_1| = 1$, while the blue ones are the possible assignments for $|x_2|$, in the case $|x_1| = 3$. The last edge layer represents the possible values for $|x_3|$.

We stress that the MDD allows the exploration of the solution space of the equation in a efficient way. In fact, at each level only a part of the possible values for a variable are considered depending on the feasible assignments of the variables of the previous levels. Moreover, the MDD can gain up to an exponential factor in representation space through the reduction process.

The worst case space complexity is $O(|b|\mathcal{V} + \delta)$, in terms of number of nodes and edges, where $\delta = \sum_{i=1}^{\mathcal{V}} |D_i|$. The p-reduction reduces the total number of edges to $\delta' \ll \delta$ and the bound of the number of nodes of any level to $\mu \leq |b|$, giving rise to a lower complexity $O(\mu\mathcal{V} + \delta')$. Actually, this bound is never reached in our experiments. As an illustrative case, consider Example 1. The MDD could have up to 1188 nodes and 352835 edges, but its reduced version has only 10 nodes and 18 edges (see Figure 1).

Let us recall that the equation over c-abstractions is a polynomial equation over natural numbers. Therefore, simplifications are possible and the whole approach can be applied to the simplified equation.

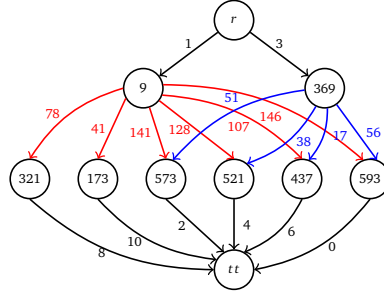


Figure 1: The reduced MDD representing all the solutions of $2 \cdot |x_3| + 5 \cdot |x_1|^2 + 4 \cdot |x_2| + 4 \cdot |x_1|^4 + 4 \cdot |x_3|^2 = 593$. There are $\mathcal{V} = 3$ variables, which are represented in the structure in the following order: $|x_1|$, $|x_2|$, and $|x_3|$.

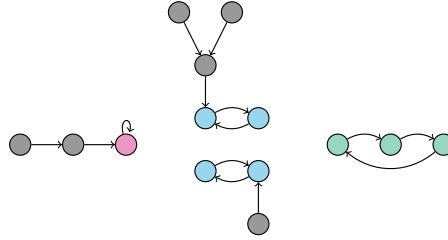


Figure 2: A DDS with four cycles ($l = 3$): $(C_1^1 \oplus C_2^2 \oplus C_3^1)$ in our notation.

4. Abstraction over the asymptotic behaviour (α -abstraction)

In this section we deal with a further abstraction, namely, the asymptotic one, describing the long-term behaviour of a DDS, *i.e.*, its ultimate periodic behaviour. In particular, we provide a method for solving the version of Equation (1) obtained considering the asymptotic behaviour of constants and variables.

Notation. In the sequel, for any pair of positive integers n and p , C_p^n will stand for the union of any n disjoint cycles of length p of a DDS S . To stress that we are dealing with sets consisting of union of disjoint cycles, each of them identifying a dynamical subsystem of S , the operations of disjoint union and product of two of such sets, or, by identification, the sum and product of the corresponding dynamical (sub)systems, will be denoted by \oplus and \odot instead of $+$ and \cdot , respectively. According to this notation, it is clear that $C_p^{n_1} \oplus C_p^{n_2} = C_p^{n_1+n_2}$ for any pair of positive naturals n_1, n_2 and $kC_p^n = C_p^{kn}$ for any positive natural k . Finally, for any positive natural i and any positive naturals p_1, \dots, p_i , denote $\lambda_i = \text{lcm}(p_1, \dots, p_i)$.

Definition 3 (α -abstraction). The α -abstraction of a DDS S , denoted by \hat{S} , is the dynamical subsystem of S induced by the set \mathcal{P} of all its periodic points, or, by identification, the set \mathcal{P} itself.

Remark 1. It immediately follows from the previous definition that the α -abstraction of the sum, *resp.*, the product, of two DDS, is the sum, *resp.*, the product of the α -abstractions of the two DDS. Moreover, the α -abstraction of a DDS S can be written as

$$\hat{S} = \bigoplus_{i=1}^l C_{p_i}^{n_i},$$

for some positive naturals l, n_1, \dots, n_l , and pairwise distinct positive naturals p_1, \dots, p_l , where, for each $i \in \{1, \dots, l\}$, n_i is the number of disjoint cycles of length p_i (see Figure 2 for an illustrative example).

The following proposition provides an explicit expression for the product of several unions of cycles. It will be very useful in the sequel.

Proposition 2. For any natural $l > 1$ and any positive naturals $n_1, \dots, n_l, p_1, \dots, p_l$, it holds that

$$\bigodot_{i=1}^l C_{p_i}^{n_i} = C_{\lambda_l}^{\frac{1}{\lambda_l} \prod_{i=1}^l (p_i n_i)}.$$

Proof. We proceed by finite induction over l . First of all, we prove that the statement is true for $l = 2$, i.e.,

$$C_{p_1}^{n_1} \odot C_{p_2}^{n_2} = C_{\lambda_2}^{\frac{1}{\lambda_2} \cdot p_1 n_1 \cdot p_2 n_2}. \quad (4)$$

Let us consider the case $n_1 = n_2 = 1$. Since $C_{p_1}^1$ and $C_{p_2}^1$ can be viewed as finite cyclic groups of order p_1 and p_2 , respectively, each element of the product of such cyclic groups has order $\text{lcm}(p_1, p_2)$ or, in other words, each element of $C_{p_1}^1 \odot C_{p_2}^1$ belongs to some cycle of length λ_2 . So, $C_{p_1}^1 \odot C_{p_2}^1$ just consists of $(p_1 \cdot p_2)/\lambda_2$ cycles, all of length λ_2 , and therefore

$$C_{p_1}^1 \odot C_{p_2}^1 = C_{\lambda_2}^{\frac{1}{\lambda_2} \cdot p_1 \cdot p_2}.$$

In the case $n_1 \neq 1$ or $n_2 \neq 1$, since the product is distributive over the sum, we get

$$C_{p_1}^{n_1} \odot C_{p_2}^{n_2} = \bigoplus_{i=1}^{n_1} C_{p_1}^1 \odot \bigoplus_{j=1}^{n_2} C_{p_2}^1 = \bigoplus_{i=1}^{n_1} \bigoplus_{j=1}^{n_2} (C_{p_1}^1 \odot C_{p_2}^1) = \bigoplus_{i=1}^{n_1} \bigoplus_{j=1}^{n_2} C_{\lambda_2}^{\frac{1}{\lambda_2} \cdot p_1 \cdot p_2} = C_{\lambda_2}^{\frac{1}{\lambda_2} \cdot p_1 n_1 \cdot p_2 n_2}.$$

Assume now that the equality holds for any $l > 1$. Then, we get

$$\bigodot_{i=1}^{l+1} C_{p_i}^{n_i} = C_{\lambda_l}^{\frac{1}{\lambda_l} \prod_{i=1}^l (p_i n_i)} \odot C_{p_{l+1}}^{n_{l+1}} = C_{\frac{\text{lcm}(\lambda_l, p_{l+1})}{\text{lcm}(\lambda_l, p_{l+1})}}^{\frac{1}{\text{lcm}(\lambda_l, p_{l+1})} \cdot \prod_{i=1}^l (p_i n_i) \cdot (p_{l+1} n_{l+1})} = C_{\lambda_{l+1}}^{\frac{1}{\lambda_{l+1}} \cdot \prod_{i=1}^{l+1} (p_i n_i)}.$$

Therefore, the equality also holds for $l + 1$ and this concludes the proof. \square

We now consider the w -th power of the union of cycles of a certain lengths and the w -th power of the sum of such unions. Before proceeding, for any DDS \mathcal{S} , we naturally define \mathcal{S}^0 as C_1^1 , i.e., the neutral element 1 of the product operation.

Corollary 1. For any natural numbers $w \geq 1$, $n \geq 1$, and $p \geq 1$, it holds that:

$$(C_p^n)^w = C_p^{p^{w-1} n^w}.$$

Proof. It is an immediate consequence of Proposition 2. \square

Proposition 3. For any positive naturals $l > 1$, $w > 1$, n_1, \dots, n_l , and p_1, \dots, p_l , it holds that

$$\left(\bigoplus_{i=1}^l C_{p_i}^{n_i} \right)^w = \bigoplus_{\substack{k_1 + \dots + k_l = w \\ 0 \leq k_1, \dots, k_l \leq w}} \binom{w}{k_1, \dots, k_l} C_{\lambda_l^*}^{\frac{1}{\lambda_l^*} \cdot \prod_{i=1}^l (p_i n_i)^{k_i}}$$

where, for any tuple k_1, \dots, k_l , λ_l^* is the lcm of those p_j with $j \in \{1, \dots, l\}$ and $k_j \neq 0$ (while $\lambda_l^* = 1$ iff all $k_j = 0$).

Proof. By Proposition 1, Proposition 2, and Corollary 1, we get

$$\begin{aligned}
\left(\bigoplus_{i=1}^l C_{p_i}^{n_i} \right)^w &= \bigoplus_{\substack{k_1+\dots+k_l=w \\ 0 \leq k_1, \dots, k_l \leq w}} \binom{w}{k_1, \dots, k_l} \bigodot_{t=1}^l (C_{p_t}^{n_t})^{k_t} \\
&= \bigoplus_{\substack{k_1+\dots+k_l=w \\ 0 \leq k_1, \dots, k_l \leq w}} \binom{w}{k_1, \dots, k_l} \bigodot_{t=1, k_t \neq 0}^l C_{p_t}^{p_t^{k_t-1} n_t^{k_t}} \\
&= \bigoplus_{\substack{k_1+\dots+k_l=w \\ 0 \leq k_1, \dots, k_l \leq w}} \binom{w}{k_1, \dots, k_l} C_{\lambda_l^*}^{\frac{1}{\lambda_l^*} \cdot \left(\prod_{t=1, k_t \neq 0}^l (p_t^{k_t} n_t^{k_t}) \right)} \\
&= \bigoplus_{\substack{k_1+\dots+k_l=w \\ 0 \leq k_1, \dots, k_l \leq w}} \binom{w}{k_1, \dots, k_l} C_{\lambda_l^*}^{\frac{1}{\lambda_l^*} \cdot \prod_{i=1}^l (p_i n_i)^{k_i}}.
\end{aligned}$$

□

We can now write the *a-abstraction equation* obtained by considering just the asymptotic behavior of all constants and variables in Equation (1):

$$\mathring{a}_1 \cdot \mathring{x}_1^{w_1} + \dots + \mathring{a}_m \cdot \mathring{x}_m^{w_m} = \mathring{b}, \quad (5)$$

where, according to Remark 1, for each $z \in \{1, \dots, m\}$ the a-abstraction of the coefficient a_z and the a-abstraction of the known term b are

$$\mathring{a}_z = \bigoplus_{i=1}^{l_z} C_{p_{zi}}^{n_{zi}} \quad \text{and} \quad \mathring{b} = \bigoplus_{j=1}^{l_b} C_{p_j}^{n_j}.$$

To solve the a-abstraction equation, we first carry out some simplifications. First of all, we consider the actual number $m \leq m$ of distinct pairs (\mathring{x}_z, w_z) appearing in such an equation. In this way, Equation (5) can be rewritten as

$$\bigoplus_{i=1}^{\ell_1} C_{p_{1i}}^{n_{1i}} \odot X_1 \oplus \dots \oplus \bigoplus_{i=1}^{\ell_m} C_{p_{mi}}^{n_{mi}} \odot X_m = \bigoplus_{j=1}^{l_b} C_{p_j}^{n_j}, \quad (6)$$

where, for each $z \in \{1, \dots, m\}$, X_z denotes $\mathring{x}_z^{w_z}$, ℓ_z is the number of the distinct lengths of the cycles forming the coefficient of X_z , and, with an abuse of notation, the number n'_{zi} of cycles of length p_{zi} inside that coefficient is still denoted by n_{zi} even though it may hold that $n'_{zi} \neq n_{zi}$.

Equation (6) is still hard to solve in this form. We can further simplify it by performing a **contraction step** which consists in rewriting it in an equivalent way as union of systems of the following type, one for each vector $(n_1^{11}, \dots, n_{l_b}^{11})$ obtained varying each $n_j^{11} \in \{0, \dots, n_j\}$ with $j \in \{1, \dots, l_b\}$:

$$\begin{cases} C_{p_{11}}^{n_{11}} \odot X_1 = \bigoplus_{j=1}^{l_b} C_{p_j}^{n_j^{11}} \end{cases} \quad (7a)$$

$$\begin{cases} C_1^1 \odot \mathring{y} = \bigoplus_{j=1}^{l_b} C_{p_j}^{n_j - n_j^{11}} \end{cases} \quad (7b)$$

where $\mathring{y} = \left(\bigoplus_{i=2}^{\ell_1} C_{p_{1i}}^{n_{1i}} \odot X_1 \right) \oplus \left(\bigoplus_{i=1}^{\ell_2} C_{p_{2i}}^{n_{2i}} \odot X_2 \right) \oplus \dots \oplus \left(\bigoplus_{i=1}^{\ell_m} C_{p_{mi}}^{n_{mi}} \odot X_m \right)$.

At this point, let us repeat as long as possible the application of the contraction step over the last equation of each system obtained by the previous contraction step. We stress that such an application essentially consists in

- i) updating \hat{y} by removing a term $C_{p_{zi}}^{n_{zi}} \odot X_z$ with $z \in \{1, \dots, m\}$ and $i \in \{1, \dots, \ell_z\}$,
- ii) considering all possible vectors $(n_1^{zi}, \dots, n_{l_b}^{zi})$ obtained varying each n_j^{zi} with $j \in \{1, \dots, l_b\}$ from 0 to the remaining number of cycles of length p_j of the right-hand side,
- iii) introducing, for each of the above mentioned vectors, a new system obtained by adding the following equation

$$C_{p_{zi}}^{n_{zi}} \odot X_z = \bigoplus_{j=1}^{l_b} C_{p_j}^{n_j^{zi}}$$

to the considered initial system just before the equation involving \hat{y} .

- iv) updating the right-hand side of the equation involving \hat{y} by removing n_j^{zi} cycles from the unions of cycles of length p_j .

In this way, we eventually get that Equation (6) can be equivalently rewritten as a union of systems, each of them having the following form

$$\left\{ \begin{array}{lcl} C_{p_{11}}^{n_{11}} \odot X_1 & = & \bigoplus_{j=1}^{l_b} C_{p_j}^{n_j^{11}} \\ C_{p_{12}}^{n_{12}} \odot X_1 & = & \bigoplus_{j=1}^{l_b} C_{p_j}^{n_j^{12}} \\ & \vdots & \\ C_{p_{1\ell_1}}^{n_{1\ell_1}} \odot X_1 & = & \bigoplus_{j=1}^{l_b} C_{p_j}^{n_j^{1\ell_1}} \\ C_{p_{21}}^{n_{21}} \odot X_2 & = & \bigoplus_{j=1}^{l_b} C_{p_j}^{n_j^{21}} \\ & \vdots & \\ C_{p_{m\ell_m}}^{n_{m\ell_m}} \odot X_m & = & \bigoplus_{j=1}^{l_b} C_{p_j}^{n_j^{m\ell_m}} \end{array} \right. . \quad (8)$$

Referring to Equation (6), we stress that, for each $j \in \{1, \dots, l_b\}$, it holds that the number of cycles of length p_j involved in know term is just $n_j = \sum_{z=1}^m \sum_{i=1}^{\ell_z} n_j^{zi}$, where n_j^{zi} represents the number of those that the monomial $C_{p_{zi}}^{n_{zi}} \odot X_z$ contributes to form.

Now, to solve any equation $C_{p_{zi}}^{n_{zi}} \odot X_z = \bigoplus_{j=1}^{l_b} C_{p_j}^{n_j^{zi}}$ from (8), it is enough to solve the following l_b equations

$$C_{p_{zi}}^{n_{zi}} \odot X_z = C_{p_1}^{n_1^{zi}}, \quad \dots, \quad C_{p_{zi}}^{n_{zi}} \odot X_z = C_{p_{l_b}}^{n_{l_b}^{zi}} \quad (9)$$

and compute the Cartesian product among their solutions. Since, for each $j \in \{1, \dots, l_b\}$, equation $C_{p_{zi}}^{n_{zi}} \odot X_z = C_{p_j}^{n_j^{zi}}$ can be rewritten as

$$C_{p_{zi}}^1 \odot X_z = C_{p_j}^{n_j^{zi}/n_{zi}},$$

if n_j^{zi}/n_{zi} is a natural number, while it has no solution, otherwise, solving Equation (6) reduces to identify all the Systems (8) and perform the products and intersections of the solutions of a certain number of simpler equations, called **basic equations**, with the following form:

$$C_p^1 \odot X = C_q^n, \quad (10)$$

where X is some X_z , $p \in \{p_{11}, p_{12}, \dots, p_{m\ell_m}\}$, $q \in \{p_1, \dots, p_{l_b}\}$, and, making reference to the right-hand side, n is smaller or equal to n_j , i.e., the number of cycles of length $q = p_j$.

To solve Equation (6), we need an efficient method that: 1) enumerates the solutions of all Equations (10), *i.e.*, the values of X_z , 2) computes the suitable products of these solutions and the intersections of sets of them, 3) retrieves the value of \hat{x}_z from X_z . The algorithmic pipeline illustrated in Figure 3 just performs all these tasks. Since a finite but potentially large number of basic equations have to be solved, the pipeline is designed in order that first of all the basic equations admitting solution are identified. In this way, Systems (8) involving basic equations without solutions are avoided, or, in other words, only feasible contraction steps, *i.e.*, feasible Systems (8) generated by contraction steps are considered. An MDD-based technique that enumerates the solutions of any basic equation is illustrated in Section 4.1 (task 1), while the identification of all the feasible contraction steps is presented in Section 4.2 along with the way of solving their corresponding feasible systems starting from the solutions of basic equations (task 2). Finally, Section 4.3 explains how to compute the DDS \hat{x}_z starting from the solutions X_z (task 3).

4.1. An MDD-based method for solving a basic equation

In this section we are going to solve a basic equation by means of a suitable MDD¹. Let us start by considering any basic equation $C_p^1 \odot X = C_q^n$. According to Remark 1, each of its solutions is expressed as a sum of unions of disjoint cycles.

By Proposition 2, among a certain number of cycles all of length p' and that form an addend of a solution, one cycle of length p' gives rise to r cycles of length q inside C_q^n when it is multiplied by C_p^1 iff r divides q , $p' = \frac{q}{p} \cdot r$, $\gcd(p, \frac{q}{p} \cdot r) = r$, and $\text{lcm}(p, \frac{q}{p} \cdot r) = q$. If such a cycle $C_{p'}^1$ satisfies the previous conditions, then it is called **feasible** and r is said to be a **feasible divisor** of q . Following this idea, let $D_{p,q} = \{d_1, \dots, d_e\}$ be the set of the feasible divisors of q . Therefore, the basic equation admits at least one solution iff there exists a set of non negative integers y_1, \dots, y_e such that $\sum_{i=1}^e d_i \cdot y_i = n$. In that case, the solution corresponding to the tuple y_1, \dots, y_e is the sum of all those $C_{p'}^{d_i \cdot y_i}$ with $y_i \neq 0$ and where $p' = \frac{q}{p} \cdot d_i$.

We now describe a method based on *Symmetry Breaking MDD* (SB-MDD) enumerating the solutions of the considered basic equation. First of all, let us introduce the MDD $M_{p,q,n}$ which is the labelled digraph (V, E, lab) with vertices forming $V = \sum_{i=1}^Z V_i$, where $Z = \lfloor \frac{n}{\min D_{p,q}} \rfloor + 1$, $V_1 = \{\text{root}\}$, V_i is a multiset of $\{1, \dots, n-1\}$ for $i \in \{2, \dots, Z-1\}$, and, finally, $V_Z = \{tt\}$. For any node $\alpha \in V$, let $\text{val}(\alpha) = \alpha$ if $\alpha \neq \text{root}$ and $\alpha \neq tt$, $\text{val}(\text{root}) = 0$, and $\text{val}(tt) = n$.

The structure is defined level by level as follows. For each level $i \in \{1, \dots, Z-2\}$, for any $\alpha \in V_i$ and any $\beta \in \{1, \dots, n-1\}$, it holds that $\beta \in V_{i+1}$ and $(\alpha, \beta) \in E$ iff $\beta - \text{val}(\alpha) \in D_{p,q}$ and $\beta \leq \text{val}(tt)$. As far as the level $i = Z-1$ is concerned, for any $\alpha \in V_i$ it holds that $(\alpha, tt) \in E$ iff $\text{val}(tt) - \text{val}(\alpha) \in D_{p,q}$ and $\beta \leq \text{val}(tt)$. The labelling map $\text{lab} : E \rightarrow D_{p,q}$ associates any edge $(\alpha, \beta) \in E$ with the value $\text{lab}(\alpha, \beta) = \text{val}(\beta) - \text{val}(\alpha) \in D_{p,q}$.

Once $M_{p,q,n}$ is built and reduced according to the p-Reduction from [15], all the solutions of the considered basic equation can be computed. Indeed, each solution corresponds to the sequence of the edge labels of a path from root to tt consisting of possibly repeated values of $D_{p,q}$ with sum equal to n . From such a sequence it is immediate to identify the above mentioned tuple y_1, \dots, y_e and then the corresponding solution.

We stress that possible permutations of each of the above mentioned sequences can be provided by $M_{p,q,n}$. In other words, distinct paths from root to tt can lead to the same solution of the given basic equation. To reduce the size of such a MDD, during its construction and before the p-reduction, a symmetry breaking constraint can be imposed: for each node $\alpha \neq tt$ the only allowed outgoing edges are those having a label which is less or equal to that of any of its incoming edges. In this way, any sequence of edge labels read on the paths of the structure turns out to be ordered and the size of the structure becomes smaller. The obtained MDD is called SB-MDD, *i.e.*, one which satisfies the symmetry breaking constraint.

Example 2. Consider the basic equation $C_4^1 \odot X = C_{12}^{12}$. The set of divisors of $q = 12$ (smaller or equal to $n = 12$) is $\{12, 6, 4, 3, 2, 1\}$. Thus, $D_{p,q} = \{4, 2, 1\}$. In fact, the following situations occur

¹This subsection and the next one are an improved version of the conference paper [11].

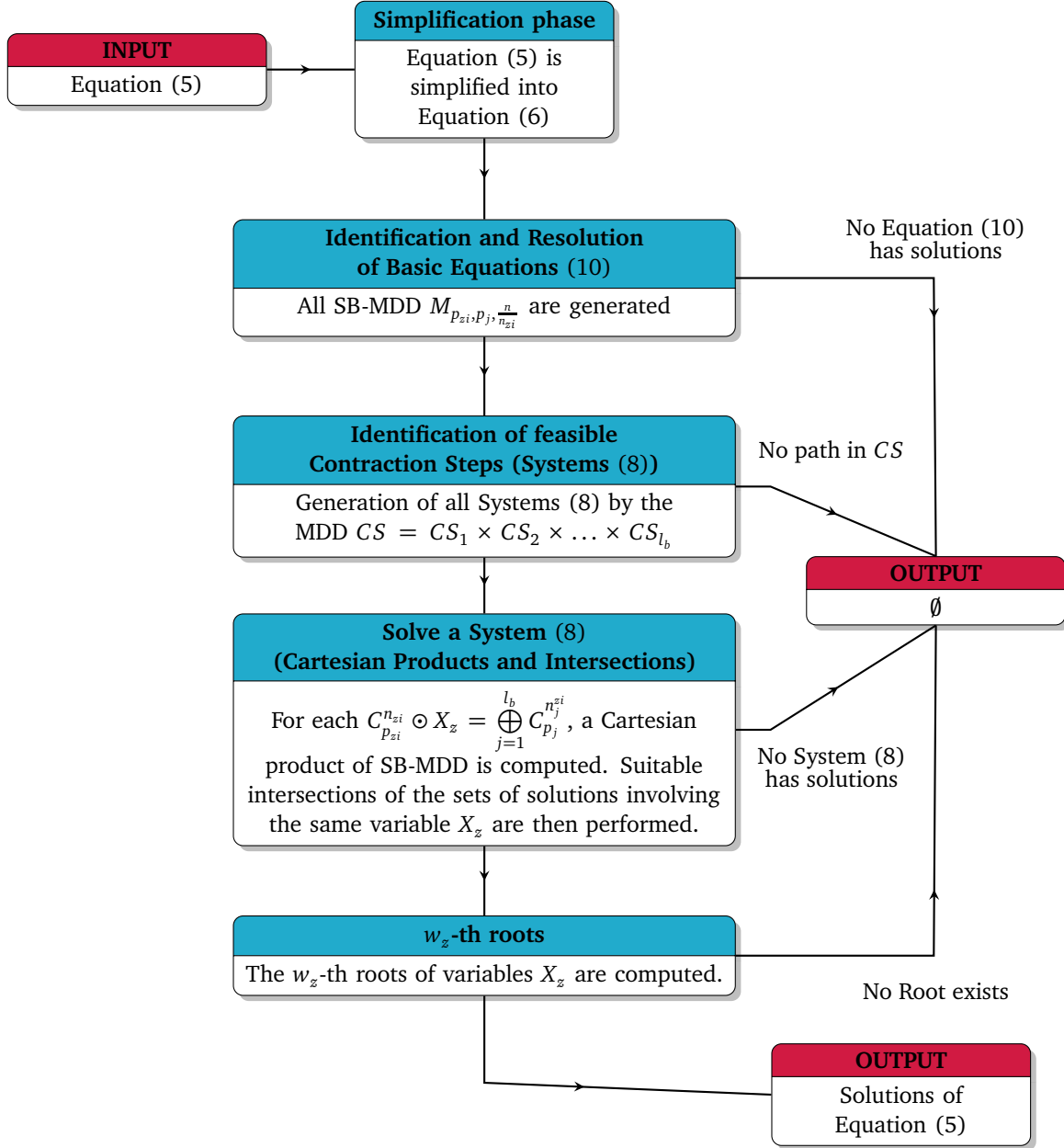


Figure 3: The MDD-based algorithmic pipeline for solving an a -abstraction equation.

$$\begin{aligned}
r = 12 \text{ and } p' = 36 &\rightarrow \gcd(4, 36) \neq 12 \text{ and } \text{lcm}(4, 36) \neq 12 \\
r = 6 \text{ and } p' = 18 &\rightarrow \gcd(4, 18) \neq 6 \text{ and } \text{lcm}(4, 18) \neq 12 \\
r = 4 \text{ and } p' = 12 &\rightarrow \gcd(4, 12) = 4 \text{ and } \text{lcm}(4, 12) = 12 \\
r = 3 \text{ and } p' = 9 &\rightarrow \gcd(4, 9) \neq 3 \text{ and } \text{lcm}(4, 9) \neq 12 \\
r = 2 \text{ and } p' = 6 &\rightarrow \gcd(4, 6) = 2 \text{ and } \text{lcm}(4, 6) = 12 \\
r = 1 \text{ and } p' = 3 &\rightarrow \gcd(4, 3) = 1 \text{ and } \text{lcm}(4, 3) = 12
\end{aligned}$$

Figure 4 shows the result of the reduction over $M_{4,12,12}$. Solutions correspond to sequences of edge labels of paths from root to tt . These sequences form the following set:

$$\begin{aligned}
&\{[4, 4, 4], [4, 4, 2, 2], [4, 4, 2, 1, 1], [4, 4, 1, 1, 1, 1], [4, 2, 2, 2, 2], [4, 2, 2, 2, 1, 1], [4, 2, 2, 1, 1, 1, 1], [4, 2, 1, 1, 1, 1, 1, 1], \\
&\quad [4, 1, 1, 1, 1, 1, 1, 1, 1], [2, 2, 2, 2, 2, 2], [2, 2, 2, 2, 2, 1, 1], [2, 2, 2, 2, 1, 1, 1, 1], [2, 2, 2, 1, 1, 1, 1, 1, 1], \\
&\quad [2, 2, 1, 1, 1, 1, 1, 1, 1, 1], [2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1], [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]\}.
\end{aligned}$$

Each element r of a sequence belongs to $D_{p,q} = \{4, 2, 1\}$ and it corresponds to a cycle of length $p' = \frac{q}{p} \cdot r$ of the solution

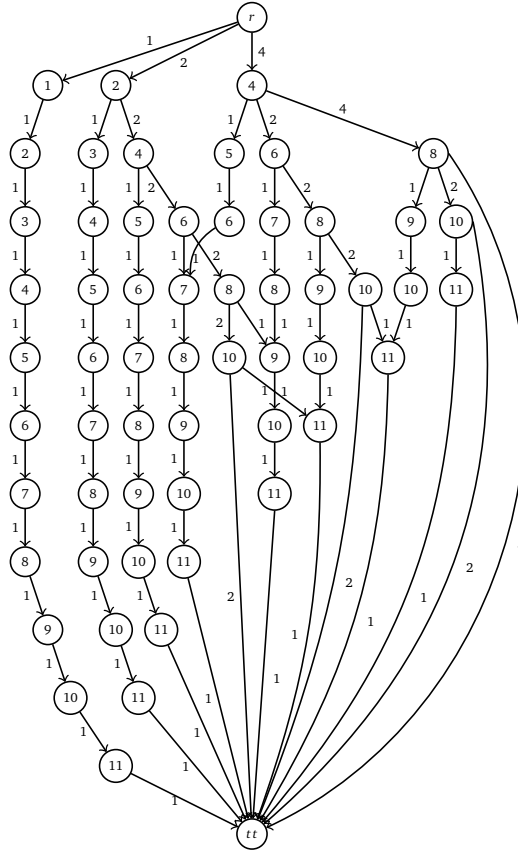


Figure 4: The reduced SB-MDD representing all the solutions of $C_4^1 \odot X = C_{12}^{12}$.

represented by that sequence. As an example, the sequence $[4, 4, 2, 1, 1]$ gives rise to 2 cycles of length $p' = \frac{q}{p} \cdot 4 = 12$, 1 cycle of length $p' = \frac{q}{p} \cdot 2 = 6$, and 2 cycles of length $p' = \frac{q}{p} \cdot 1 = 3$, i.e., the solution $C_{12}^2 \oplus C_6^1 \oplus C_3^2$.

$$\begin{aligned}
&\{C_{12}^3, C_{12}^2 \oplus C_6^2, C_{12}^2 \oplus C_6^1 \oplus C_3^2, C_{12}^2 \oplus C_3^4, C_{12}^1 \oplus C_6^4, C_{12}^1 \oplus C_6^3 \oplus C_3^2, C_{12}^1 \oplus C_6^2 \oplus C_3^4, C_{12}^1 \oplus C_6^1 \oplus C_3^6, \\
&\quad C_{12}^1 \oplus C_3^8, C_6^6, C_6^5 \oplus C_3^2, C_6^4 \oplus C_3^4, C_6^3 \oplus C_3^6, C_6^2 \oplus C_3^8, C_6^1 \oplus C_3^{10}, C_3^{12}\}.
\end{aligned}$$



Figure 5: The SB-MDD (before reduction) representing all the solutions of $C_2^1 \odot X = C_4^5$. The red part is deleted by the pReduction procedure.

The method based on the above described SB-MDD also establishes the instances of equations without solutions via the following criteria:

- if p cannot divide q ;
- if $D_{p,q}$ is the empty set;
- if, after the reduction process, no valid paths from $root$ to tt remain in the SB-MDD structure.

The following example just illustrates how the method establishes whether an instance of a basic equation has no solutions.

Example 3. Consider the equation $C_2^1 \odot X = C_4^5$. The set of divisors of q (smaller or equal to n) is $\{4, 2, 1\}$. Thus, $D_{p,q} = \{2\}$. Indeed, the following situations occur

$$\begin{aligned} r = 4 \text{ and } p' = 8 &\rightarrow \gcd(2, 8) \neq 4 \text{ and } \text{lcm}(2, 8) \neq 4 \\ r = 2 \text{ and } p' = 4 &\rightarrow \gcd(2, 4) = 2 \text{ and } \text{lcm}(2, 4) = 4 \\ r = 1 \text{ and } p' = 2 &\rightarrow \gcd(2, 2) \neq 1 \text{ and } \text{lcm}(2, 2) \neq 4 \end{aligned}$$

Figure 5 shows $M_{5,4,2}$ before the reduction procedure. The red part is deleted when the reduction phase is performed. The SB-MDD has no paths from the root to tt node, and, hence, the equation has no solutions.

Experiments show how this method can achieve interesting performances in time and memory [11].

4.2. Contraction steps

We now present how all feasible Systems (8) can be first generated starting from Equation (6) and then solved. Since Systems (8) may lead to basic equations without solutions and the same basic equation may be reached several times as far as distinct systems are considered, first of all the basic equations that can be involved have to be individuated and among them only the **necessary** ones, *i.e.*, those admitting a solution, have to be solved just once.

The identification of all the involved basic equations consists in considering all the SB-MDD $M_{p_{zi}, p_j, n/n_{zi}}$ defined by varying $z \in \{1, \dots, m\}$, $i \in \{1, \dots, \ell_z\}$, $j \in \{1, \dots, l_b\}$, and $n \in \{1, \dots, n_j\}$. Then, those SB-MDD corresponding to necessary basic equations are computed, *i.e.*, all the necessary basic equations are solved.

We now describe an MDD able to generate all feasible Systems (8). Such an MDD is $CS = CS_1 \times \dots \times CS_{l_b}$, *i.e.*, the Cartesian product of l_b MDD, where each CS_j aims at providing, according to the set of the necessary equations, all the feasible ways by which the monomials of Equation (6) can concur to form the n_j cycles of length p_j of the known term \vec{b} . Clearly, by the Stars and Bars method those ways are at most $\binom{n_j + \ell - 1}{\ell - 1}$ and, hence, there are at most $\prod_{j=1}^{l_b} \binom{n_j + \ell - 1}{\ell - 1}$ feasible Systems (8). Furthermore, by definition, the whole MDD CS will provide all the feasible ways by which all the cycles of \vec{b} can be formed.

Each CS_j is a labelled digraph (V_j, E_j, lab_j) in which there are $m \cdot \ell_z$ levels, one for each monomial $C_{p_{zi}}^{n_{zi}} \odot X_z$ from the left-hand side of Equation (6), besides the level containing the only terminal node tt . The vertex set

is $V_j = (\sum_{z \in \{1, \dots, m\}} \sum_{i \in \{1, \dots, \ell_z\}} V_{j,zi}) + V_{j,(m+1)1}$ where $V_{j,11} = \{root\}$, $V_{j,(m+1)1} = \{tt\}$, and for each pair (z, i) with $z \in \{1, \dots, m\}$ and $i \in \{1, \dots, \ell_z\}$ the set $V_{j,zi} \subseteq \{0, \dots, n_j\}$ of the vertexes of the level (z, i) will be defined in the sequel. Indeed, the graph is built level by level. Moreover, for any node $\alpha \in V_j$, let $val(\alpha) = \alpha$ if $\alpha \neq root$ and $\alpha \neq tt$, while $val(root) = 0$ and $val(tt) = n_j$. To define the edges outgoing from the vertexes of any level along with the corresponding label and then the vertexes of the next level too, first of all we associate each level (z, i) with the set $D_{p_{zi}, p_j} = \{d \in \mathbb{N} \mid 1 \leq d \leq n_j \text{ and } M_{p_{zi}, p_j, d/n_{zi}} \text{ is defined by a necessary equation}\} \cup \{0\}$ of the labels of the edges outgoing from the vertexes of that level. Now, for each level (z, i) with $z \neq m$ and $i \neq \ell_z$, for any vertex $\alpha \in V_{j,zi}$ and any $\beta \in \{0, \dots, n_j\}$, it holds that

- $\beta \in V_{j,z(i+1)}$ and $(\alpha, \beta) \in E_j$ iff $\beta - val(\alpha) \in D_{p_{zi}, p_j}$ and $\beta \leq val(tt)$, whenever $i < \ell_z$;
- $\beta \in V_{j,(z+1)1}$ and $(\alpha, \beta) \in E_j$ iff $\beta - val(\alpha) \in D_{p_{zi}, p_j}$ and $\beta \leq val(tt)$, whenever $i = \ell_z$.

Concerning the level (m, ℓ_z) , for any vertex $\alpha \in V_{j,m\ell_z}$ it holds that $(\alpha, tt) \in E_j$ iff $val(tt) - val(\alpha) \in D_{p_{m\ell_z}, p_j}$. Every edge $(\alpha, \beta) \in E_j$ is associated with the label $lab_j(\alpha, \beta) = val(\beta) - val(\alpha) \in D_{p_{zi}, p_j}$, where (z, i) is such that $\alpha \in V_{j,zi}$. In this way, the labelling map $lab_j: E_j \rightarrow \bigcup_{z \in \{1, \dots, m\}} \bigcup_{i \in \{1, \dots, \ell_m\}} D_{p_{zi}, p_j}$ has been defined too.

We stress that any edge outgoing from vertexes of the level (z, i) represents the cycles of length p_j that the monomial $C_{p_{zi}}^{n_{zi}} \odot X_z$ can contribute to form together with the monomials corresponding to the other edges encountered on a same path from $root$ to tt . The label of the edge is just the number n_j^{zi} of those cycles and the sum of all the labels of the edges in any path from $root$ to tt is just the number n_j cycles of length p_j to be formed by the monomials $C_{p_{zi}}^{n_{zi}} \odot X_z$ of the left-hand side of Equation (6). The value $val(\alpha)$ associated with a node α of a path from $root$ to tt is the partial result of that sum, i.e., the number of cycles of length p_j formed by the monomials encountered on the subpath from $root$ to α .

At this point, the MDD CS is built and, according to the definition of cartesian product of MDDs, the involved MDDs are stacked on top of each other in such a way that each CS_j turns out to be on top of CS_{j+1} and the terminal node of CS_j is collapsed with the root of CS_{j+1} . Any path from the root to the terminal node of CS represents a possible way by which the monomials $C_{p_{zi}}^{n_{zi}} \odot X_z$ of the left-hand side of Equation (6) can concur to form all the cycles of \hat{b} , or, in other words, it corresponds to a possible solution of Equation (6). In particular, since for each pair (z, i) a level (z, i) appears in every CS_j , the set of the l_b edges in any of the above mentioned paths of CS , each of them outgoing from vertexes of the same level (z, i) in one CS_j , defines a feasible way of solving the equation from System (8)

$$C_{p_{zi}}^{n_{zi}} \odot X_z = \bigoplus_{j=1}^{l_b} C_{p_j}^{n_j^{zi}},$$

i.e., a way by which the monomial $C_{p_{zi}}^{n_{zi}} \odot X_z$ gives rise at the same time to n_1^{zi} cycles of length p_1 , n_2^{zi} cycles of length p_2 , ..., and $n_{l_b}^{zi}$ cycles of length p_{l_b} . Therefore, all the monomials encountered in a path of CS contribute to form a possibly feasible System (8).

Example 4. Consider the equation:

$$C_4^1 \odot X_1 \oplus C_2^1 \odot X_2 = C_2^4 \oplus C_4^4 \oplus C_6^7 \oplus C_{12}^7.$$

There are 44 distinct basic equations and among them 27 equations are necessary. Indeed, besides the basic equations defined by $p = 4$ and $q \in \{2, 6\}$, the following ones have no solution: $C_2^1 \odot X_2 = C_4^1$, $C_2^1 \odot X_2 = C_4^3$, $C_2^1 \odot X_2 = C_{12}^1$, $C_2^1 \odot X_2 = C_{12}^3$, $C_2^1 \odot X_2 = C_{12}^5$, and $C_2^1 \odot X_2 = C_{12}^7$.

To illustrate one CS_j , let us consider $j = 2$, or, in other words, the MDD providing all the possible ways by which the two monomials of the given equation can concur to form C_4^4 . Thus, CS_2 has 2 levels, one for each monomial. Any edge outgoing from a level represents the cycles of length 4, along the number of them, that the monomial corresponding to that level can contribute to form. The first level, corresponding to the monomial $C_4^1 \odot X_1$, only contains the node $root$. According to the necessary equations defined $p = 4$ and $q = 4$, the first monomial is able by itself to form n_2^{11} cycles of length 4 where $n_2^{11} \in \{1, 2, 3, 4\}$. Regarding the second monomial, it is able by itself to form either $n_2^{21} = 2$ or $n_2^{21} = 4$ cycles of length 4. As Figure 6 shows, the MDD CS_2 also represents the cases $n_2^{11} = 0$ and/or $n_2^{21} = 0$, i.e.,

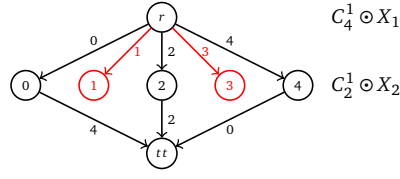


Figure 6: The MDD CS_2 represents all the possible ways by which, according to the set of necessary equations, the two monomials can concur to form C_4^4 . The red part is deleted by the pReduction procedure. The value $val(\alpha)$ associated with each node α is also reported.

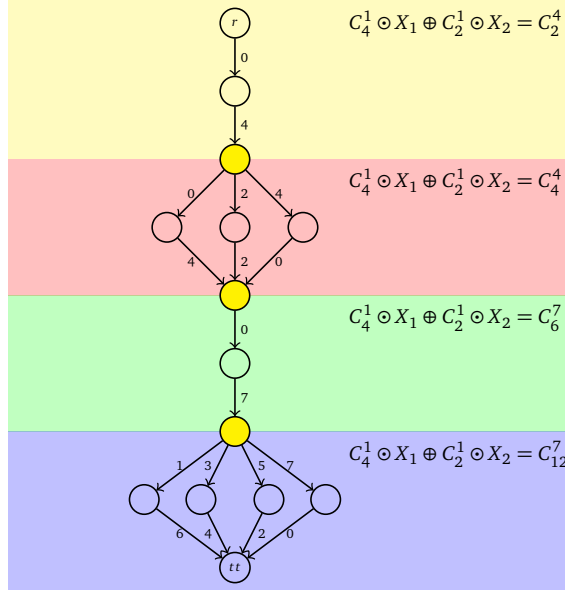


Figure 7: The MDD CS represents all the feasible ways by which, according to the set of necessary equations, the monomials of the equation from Example 4 can concur to form its right-hand side. According to the cartesian product of MDD, the yellow nodes are at the same time the tt node of a CS_j and the root node of CS_{j+1} . The four MDDs are depicted by different colours (the red MDD corresponds to that from Figure 6). In each CS_j the first (resp., second) level corresponds to the monomial $C_4^1 \odot X_1$ (resp., $C_2^1 \odot X_2$). The values $val(\alpha)$ associated to nodes are omitted for simplicity.

where at least one of the two monomials does not contribute to the generation of such cycles at all. Any path from root to tt provides a feasible way by which the two monomials concur to form $n_2 = 4$ cycles of length $p_2 = 4$. Figure 7 illustrates the MDD $CS = CS_1 \times CS_2 \times CS_3 \times CS_4$ associated with the given equation and obtained by stacking each CS_j on top of CS_{j+1} . Any path from the root to the terminal node of CS represents a possible way by which the monomials of the left-hand side of the given equation can concur to form all the cycles of its known term.

Now, solving any equation from a System (8) means computing the cartesian product among the solutions of the l_b equations in (9). Since each of them can be equivalently rewritten as a basic equation, this can be performed by computing the cartesian product of the SB-MDD, each providing the solutions of the involved basic equation. As usual, such a cartesian product, that we name **SB-Cartesian MDD**, is obtained by stacking the SB-MDDs on top of each other. In this way, one can get the values of the X_z satisfying any equation of the System (8) defined by a path of CS . We stress that an SB-Cartesian MDD is not a SB-MDD. In particular, although it is satisfied by each of its component SB-MDD, the order constraint among the edge labels of any path from the root to the terminal node of an SB-Cartesian MDD does not hold.

To provide the solutions of a System (8), for each X_z the intersection among the solutions of all the equations

involving the same variable X_z is required. Then, once the values of the x_z will have been computed starting from the values of X_z by means of the algorithm presented in Section 4.3, a further intersection of the sets of values of a same x_z arisen from distinct X_z (if any) will be performed. Indeed, there can be equations in distinct variables that however are (distinct) powers of the a same variable x_z . We now deal with the first mentioned intersection (the second one is standard and it can be performed in such a way that only one root of the variables X_z that are powers of a same x_z is computed).

According to the current state of the art, there exists an algorithm that, starting from two MDD, possibly two SB-MDD, each of them providing the solutions of an equation, builds a new MDD able to compute the intersection between the solutions of the two equations. Essentially, each node in the new structure corresponds to two nodes, one from each MDD, and the procedure recreates an outgoing edge in the structure if it is common to both the MDDs. For more details, we refer the reader to [15] and [4].

Nevertheless, such an algorithm can not be used if SB-Cartesian MDDs are involved, as it happens instead in our scenario, unless each monomial gives rise to cycles of a unique length, *i.e.*, the solutions of each corresponding equation are computed by a SB-MDD. Indeed, the result of the above mentioned algorithm depends on the order by which the SB-Cartesian MDDs are considered when the intersection is performed. In [11], a new algorithm performing the intersection has been proposed in such a way that it properly works independently of that order. Let us recall its underlying idea.

The algorithm starts to compute the intersection among the solutions of equations provided by all the SB-MDDs, if any. If it is not empty, such an intersection consists of a set of candidate solutions that form the so-called initial guess. Otherwise, the initial guess is the set of the solutions provided by one of the SB-Cartesian MDDs. The current set of candidate solutions which at the beginning is just the initial guess is updated by means of the intersection between itself and the set of the solutions provided by one of the SB-Cartesian MDDs that have not yet been considered. Any intersection essentially consists in visiting the chosen SB-Cartesian MDD CS to establish whether a candidate solution is provided by one among the SB-MDD components CS_j of CS . If this does not happen, it is removed from the set of candidate solutions.

4.3. Roots of DDS

We now deal with the problem of retrieving the value of each DDS \hat{x}_z once the DDS X_z have been computed. Since each X_z is the w_z -th power of \hat{x}_z , we are going to introduce the concept of w -th root in the semiring of DDS and provide an algorithm for computing the w -th roots of the a-abstractions of DDS.

First of all, let us formally define the notion of w -root of a general DDS.

Definition 4. Let $w \geq 2$ be a natural number. The w -th root of a DDS S is a DDS having w -th power equal to S .

Clearly, the a-abstraction of the w -th root of a DDS is the w -th root of the a-abstraction of that system. The goal is now to compute the w -th root of the a-abstraction of any DDS. Namely, for any given a-abstraction

$$C_{o_1}^{s_1} \oplus \dots \oplus C_{o_h}^{s_h},$$

with $0 < o_1 < o_2 < \dots < o_h$, we want to solve the equation

$$\hat{x}^w = C_{o_1}^{s_1} \oplus \dots \oplus C_{o_h}^{s_h}, \quad (11)$$

where the unknown is expressed as

$$\hat{x} = C_{p_1}^{n_1} \oplus \dots \oplus C_{p_l}^{n_l},$$

for some naturals $l, p_1, \dots, p_l, n_1, \dots, n_l$ $p_1 < \dots < p_l$, and n_1, \dots, n_l to be determined.

Assumption. From now on, without loss of generality, we will assume $p_1 < \dots < p_l$, and $o_1 < \dots < o_h$.

Since providing a closed formula for \hat{x} is essentially unfeasible, we are going to compute the sets $C_{p_i}^{n_i}$ one by one starting from $i = 1$. Such a computation will be iteratively performed by considering the generation of the sets $C_{o_j}^{s_j}$ by carrying out the w -th power of the sum of sets $C_{p_i}^{n_i}$.

Proposition 4. For any natural $l \geq 2$, if $\hat{x} = C_{p_1}^{n_1} \oplus \dots \oplus C_{p_l}^{n_l}$ is a solution of the equation $\hat{x}^w = C_{o_1}^{s_1} \oplus \dots \oplus C_{o_h}^{s_h}$, then all the following facts hold:

- (i) $l \leq h$ and $\{p_1, \dots, p_l\} \subseteq \{o_1, \dots, o_h\}$
- (ii) $p_1 = o_1$ and $p_2 = o_2$;
- (iii) $n_1 = \sqrt[w]{\frac{s_1}{o_1^{w-1}}} \in \mathbb{N}$;
- (iv) $n_2 = \begin{cases} \frac{\sqrt[w]{o_2 s_2 + o_1 s_1} - \sqrt[w]{o_1 s_1}}{o_2} \in \mathbb{N}, & \text{if } \text{lcm}(o_1, o_2) = o_2, \\ \sqrt[w]{\frac{s_2}{o_2^{w-1}}} \in \mathbb{N}, & \text{otherwise.} \end{cases}$

Proof.

(i): According to Proposition 3, for each $i \in \{1, \dots, l\}$, a set $C_{\lambda_i^*}^s$ with $\lambda_i^* = p_i$ appears in \hat{x}^w when the tuple (k_1, \dots, k_l) with $k_i = w$ and $k_{i'} = 0$ for $i' \neq i$ is involved in the sum. Hence, $\{p_1, \dots, p_l\} \subseteq \{o_1, \dots, o_h\}$ and $l \leq h$.

(ii): Since p_1 is the smallest value among all possible $\text{lcm } \lambda_i^*$ from Proposition 3 and o_1 is the smallest among the lengths o_1, \dots, o_h of the cycles to be generated when the w -th power of \hat{x} is performed, it must necessarily hold that $p_1 = o_1$ in order that, in particular, cycles of length o_1 are generated. Moreover, since p_2 and o_2 follow in ascending order p_1 and o_1 , respectively, and p_2 is also the successor of p_1 among all the above mentioned lcm , it must also hold that $p_2 = o_2$ in order that cycles of length o_2 are generated too.

(iii): Actually, it holds that $(C_{o_1}^{n_1})^w = C_{o_1}^{s_1}$, which, by Corollary 1, is equivalent to $C_{o_1}^{o_1^{w-1} n_1^w} = C_{o_1}^{s_1}$. This implies that $o_1^{w-1} n_1^w = s_1$ and, hence, $n_1 = \sqrt[w]{\frac{s_1}{o_1^{w-1}}}$.

(iv): if $\text{lcm}(o_1, o_2) > o_2$, when computing the w -th power of \hat{x} , by Lemma 1, $C_{p_1}^{n_1}$ does not contribute to form $C_{o_2}^{s_2}$ and, necessarily, it holds that $(C_{o_2}^{n_2})^w = C_{o_2}^{o_2^{w-1} n_2^w} = C_{o_2}^{s_2}$. So, we get $o_2^{w-1} n_2^w = s_2$, the latter implying that $n_2 = \sqrt[w]{\frac{s_2}{o_2^{w-1}}}$. If $\text{lcm}(o_1, o_2) = o_2$, both $C_{p_1}^{n_1}$ and $C_{p_2}^{n_2}$ contribute to $C_{o_2}^{s_2}$. In particular, it holds that $(C_{p_1}^{n_1} \oplus C_{p_2}^{n_2})^w = C_{o_1}^{s_1} \oplus C_{o_2}^{s_2}$. By Proposition 1, one finds

$$(C_{p_1}^{n_1})^w \oplus \bigoplus_{i=1}^{w-1} \binom{w}{i} (C_{p_1}^{n_1})^i \odot (C_{p_2}^{n_2})^{w-i} \oplus (C_{p_2}^{n_2})^w = C_{o_1}^{s_1} \oplus C_{o_2}^{s_2}.$$

Since $(C_{p_1}^{n_1})^w = C_{o_1}^{s_1}$ and by Corollary 1 and Proposition 2, that can be rewritten as follows

$$C_{p_2}^{p_2^{w-1} n_2^w} \oplus \bigoplus_{i=1}^{w-1} \binom{w}{i} C_{\frac{1}{\text{lcm}(p_1, p_2)} \cdot p_1^i n_1^i \cdot p_2^{w-i} n_2^{w-i}} = C_{o_2}^{s_2}$$

Recalling that $\text{lcm}(o_1, o_2) = o_2$, $p_1 = o_1$, and $p_2 = o_2$, the latter equality is true iff

$$o_2^{w-1} n_2^w + \sum_{i=1}^{w-1} \binom{w}{i} o_1^i n_1^i o_2^{w-i-1} \cdot n_2^{w-i} = s_2,$$

i.e., once both sides are first multiplied by o_2 and then added to the term $(o_1 n_1)^w$, iff

$$(o_1 n_1 + o_2 n_2)^w = o_2 s_2 + (o_1 n_1)^w.$$

By (i) and (iii), we get

$$n_2 = \frac{\sqrt[w]{o_2 s_2 + o_1 s_1} - \sqrt[w]{o_1 s_1}}{o_2}.$$

□

The following theorem explains how to compute n_{i+1} and p_{i+1} once n_1, \dots, n_i and p_1, \dots, p_i are also known.

Theorem 1. Let $\hat{x} = C_{p_1}^{n_1} \oplus \dots \oplus C_{p_l}^{n_l}$ be a solution of the equation $\hat{x}^w = C_{o_1}^{s_1} \oplus \dots \oplus C_{o_h}^{s_h}$. For any fixed natural i with $2 \leq i < l$, if $n_1, \dots, n_i, p_1, \dots, p_i$ are known and $t \in \{i, \dots, h\}$, $s'_1, \dots, s'_t, o'_1, \dots, o'_t$ are positive integers such that $(C_{p_1}^{n_1} \oplus C_{p_2}^{n_2} \oplus \dots \oplus C_{p_i}^{n_i})^w = C_{o'_1}^{s'_1} \oplus C_{o'_2}^{s'_2} \oplus \dots \oplus C_{o'_t}^{s'_t}$, then the following facts hold:

$$(1) \quad p_{i+1} = o_{\xi_{i+1}},$$

where $\xi_{i+1} = \min \{j \in \{1, \dots, h\} \text{ with } o_j > p_i \mid o_j > o'_t \vee (o_j = o'_z \text{ for some } 1 \leq z \leq t \text{ with } s'_z < s_j)\}$;

$$(2) \quad n_{i+1} = \begin{cases} \frac{\sqrt[w]{o_{\xi_{i+1}} s_{\xi_{i+1}}} + Q_i^* - \sum_{j=1}^i p_j n_j}{o_{\xi_{i+1}}}, & \text{if } \text{lcm}(p_1, \dots, p_{i+1}) = p_{i+1}, \\ \frac{\sqrt[w]{o_{\xi_{i+1}} s_{\xi_{i+1}}} + Q_i^{**} - \sum_{e=1}^{j-1} p_{i_e} n_{i_e}}{o_{\xi_{i+1}}}, & \text{otherwise,} \end{cases}$$

where

$$Q_i^* = \sum_{\substack{k_1 + \dots + k_i = w \\ 0 \leq k_1, \dots, k_i \leq w \\ \lambda_i^* \neq p_{i+1}}} \binom{w}{k_1, \dots, k_i} \prod_{t=1}^i (p_t n_t)^{k_t},$$

with λ_i^* as in Proposition 3,

$$Q_i^{**} = \sum_{\substack{k_{i_1} + \dots + k_{i_{j-1}} = w \\ 0 \leq k_{i_1}, \dots, k_{i_{j-1}} \leq w \\ \lambda_{i_{j-1}}^{**} \neq p_{i+1}}} \binom{w}{k_{i_1}, \dots, k_{i_{j-1}}} \prod_{t=1}^{j-1} (p_{i_t} n_{i_t})^{k_{i_t}},$$

and, regarding Q_i^{**} , the set $\{i_1, \dots, i_j\}$ is the maximal subset of $\{1, \dots, i+1\}$ such that $i_1 < \dots < i_j$, $i_j = i+1$, and p_{i_e} divides p_{i+1} for each $1 \leq e \leq j$ (i.e., $\text{lcm}(p_{i_1}, \dots, p_{i_j}) = p_{i+1}$), and where, for each $1 \leq e \leq j$ and for any tuple k_{i_1}, \dots, k_{i_e} , $\lambda_{i_e}^{**}$ denotes the lcm of those p_{i_e} with $e \in \{1, \dots, e\}$ and $k_{i_e} \neq 0$ (while $\lambda_{i_e}^{**} = 1$ iff all $k_{i_e} = 0$).

Proof. (1) We deal with the following two mutually exclusive cases a) and b).

Case a): for some $j \in \{1, \dots, h\}$ the following condition holds: there exists $z \in \{1, \dots, t\}$ such that $o_j = o'_z$ and $s'_z < s_j$. This means that, when the w -th power is performed, cycles from the part $(C_{p_1}^{n_1} \oplus \dots \oplus C_{p_i}^{n_i})$ of the solution give rise to a number s'_z of cycles of length $o'_z = o_j$ where s'_z is lower than the number s_j of cycles of length o_j that are expected once the w -th power of the whole solution is computed. Consider the minimum among all the indexes j satisfying the above introduced condition. It is clear that ξ_{i+1} is just such a minimum and $o_{\xi_{i+1}}$ is the minimum among the values o_j corresponding to those indexes j . Since by Corollary 1 and regarding each j satisfying the above mentioned condition the w -th power of cycles of length $o'_z = o_j$ gives rise to cycles of length o'_z , by item (i) of Proposition 4 p_{i+1} comes from the set $\{o_1, \dots, o_h\}$, and it is the successor of p_i , we get that p_{i+1} can be nothing but $o_{\xi_{i+1}}$, or, equivalently, $i+1 = \xi_{i+1}$. Indeed, according to Proposition 3, if cycles of length greater than $o_{\xi_{i+1}}$ were added to the part $(C_{p_1}^{n_1} \oplus \dots \oplus C_{p_i}^{n_i})$ of the solution instead of cycles of length $o_{\xi_{i+1}}$, they would give rise to cycles of greater length, barring the generation of the missing cycles of length $o_{\xi_{i+1}}$.

Case b): there is no index $j \in \{1, \dots, h\}$ satisfying the above mentioned condition. Similar arguments from case a) over the values o_j and the corresponding indexes j such that $o_j > o'_t$ lead to the conclusion that ξ_{i+1} is the minimum of such indexes, $i+1 = \xi_{i+1}$, and $p_{i+1} = o_{\xi_{i+1}}$.

(2) We deal with the following two mutually exclusive cases:

Case 2.1): $\text{lcm}(p_1, \dots, p_{i+1}) = p_{i+1}$. By Proposition 3, we can write

$$(C_{p_1}^{n_1} \oplus \dots \oplus C_{p_{i+1}}^{n_{i+1}})^w = \bigoplus_{\substack{k_1 + \dots + k_{i+1} = w \\ 0 \leq k_1, \dots, k_{i+1} \leq w}} \binom{w}{k_1, k_2, \dots, k_{i+1}} C_{\lambda_{i+1}^*}^{\frac{1}{\lambda_{i+1}^*} \prod_{t=1}^{i+1} (p_t n_t)^{k_t}}$$

Among all the addends of the latter sum, only the ones with a multinomial coefficient defined by k_1, \dots, k_{i+1} such that $\lambda_{i+1}^* = p_{i+1}$ give rise to cycles of length p_{i+1} , where $p_{i+1} = o_{\xi_{i+1}}$. In particular, it holds that

$$\bigoplus_{\substack{k_1 + \dots + k_{i+1} = w \\ 0 \leq k_1, \dots, k_{i+1} \leq w \\ \lambda_{i+1}^* = p_{i+1}}} \binom{w}{k_1, \dots, k_{i+1}} C_{\lambda_{i+1}^*}^{\frac{1}{\lambda_{i+1}^*} \prod_{t=1}^{i+1} (p_t n_t)^{k_t}} = C_{o_{\xi_{i+1}}}^{s_{\xi_{i+1}}},$$

and, hence,

$$\sum_{\substack{k_1 + \dots + k_{i+1} = w \\ 0 \leq k_1, k_2, \dots, k_{i+1} \leq w \\ \lambda_{i+1}^* = p_{i+1}}} \binom{w}{k_1, \dots, k_{i+1}} \cdot \frac{1}{\lambda_{i+1}^*} \cdot \prod_{t=1}^{i+1} (p_t n_t)^{k_t} = s_{\xi_{i+1}}. \quad (12)$$

Since $\lambda_{i+1}^* = o_{\xi_{i+1}}$, when both sides of Equation (12) are first multiplied by $o_{\xi_{i+1}}$ and then summed to the quantity

$$\sum_{\substack{k_1 + \dots + k_{i+1} = w \\ 0 \leq k_1, \dots, k_{i+1} \leq w \\ \lambda_{i+1}^* \neq p_{i+1} \wedge k_{i+1} = 0}} \binom{w}{k_1, \dots, k_{i+1}} \prod_{t=1}^{i+1} (p_t n_t)^{k_t} = \sum_{\substack{k_1 + \dots + k_i = w \\ 0 \leq k_1, \dots, k_i \leq w \\ \lambda_i^* \neq p_{i+1}}} \binom{w}{k_1, \dots, k_i} \prod_{t=1}^i (p_t n_t)^{k_t} = Q_i^*,$$

Equation (12) becomes

$$\sum_{\substack{k_1 + \dots + k_{i+1} = w \\ 0 \leq k_1, \dots, k_{i+1} \leq w}} \binom{w}{k_1, \dots, k_{i+1}} \prod_{t=1}^{i+1} p_t^{k_t} n_t^{k_t} = o_{\xi_{i+1}} s_{\xi_{i+1}} + Q_i^*. \quad (13)$$

Indeed, by the assumption that $\text{lcm}(p_1, \dots, p_{i+1}) = p_{i+1}$, there can be no tuple (k_1, \dots, k_{i+1}) from the sum of Equation (13) such that both the conditions $k_{i+1} \neq 0$ and $\lambda_{i+1}^* \neq p_{i+1}$ hold. Now, Equation (13) can be rewritten as

$$(p_1 n_1 + \dots p_i n_i + p_{i+1} n_{i+1})^w = o_{\xi_{i+1}} s_{\xi_{i+1}} + Q_i^*.$$

Since Q_i^* does not depend on n_{i+1} , and, in particular, Q_i^* can be computed on the basis of n_1, \dots, n_i , we get

$$n_{i+1} = \frac{\sqrt[w]{o_{\xi_{i+1}} s_{\xi_{i+1}} + Q_i^*} - \sum_{j=1}^i p_j n_j}{o_{\xi_{i+1}}}$$

Case 2.2): $\text{lcm}(p_1, \dots, p_{i+1}) > p_{i+1}$. When computing the w -th power of \hat{x} the set $C_{o_{\xi}}^{s_{\xi}} = C_{p_{i+1}}^{s_{\xi}}$ can be formed only by the contribution of those sets $C_{p_{i_1}}^{n_{i_1}}, \dots, C_{p_{i_j}}^{n_{i_j}}$ (including $C_{p_{i+1}}^{n_{i+1}}$) such that $i_1 < \dots < i_j$, $i_j = i+1$, and p_{i_e} divides p_{i+1} for each $1 \leq e \leq j$. Since $\text{lcm}(p_{i_1}, \dots, p_{i_j}) = p_{i+1}$, we can proceed in the same way as the case 2.1) but with the indexes i_1, \dots, i_j instead of $1, \dots, i+1$, respectively. Therefore, it holds that

$$\sum_{\substack{k_{i_1} + \dots + k_{i_j} = w \\ 0 \leq k_{i_1}, \dots, k_{i_j} \leq w}} \binom{w}{k_{i_1}, \dots, k_{i_j}} \prod_{t=1}^j p_{i_t}^{k_{i_t}} n_{i_t}^{k_{i_t}} = o_{\xi_{i+1}} s_{\xi_{i+1}} + Q_i^{**}, \quad (14)$$

where

$$Q_i^{**} = \sum_{\substack{k_{i_1} + \dots + k_{i_{j-1}} = w \\ 0 \leq k_{i_1}, \dots, k_{i_{j-1}} \leq w \\ \lambda_{i_{j-1}}^{**} \neq p_{i+1}}} \binom{w}{k_{i_1}, \dots, k_{i_{j-1}}} \prod_{t=1}^{j-1} (p_{i_t} n_{i_t})^{k_{i_t}},$$

and, hence, Equation (14) can be rewritten as

$$(p_{i_1} n_{i_1} + \dots p_{i_{j-1}} n_{i_{j-1}} + p_{i_j} n_{i_j})^w = o_{\xi_{i+1}} s_{\xi_{i+1}} + Q_i^{**}.$$

Since Q_i^{**} does not depend on $n_{i+1} = n_{i_j}$, and, in particular, Q_i^{**} can be computed on the basis of $n_{i_1}, \dots, n_{i_{j-1}}$, we get

$$n_{i+1} = \frac{\sqrt[w]{o_{\xi_{i+1}} s_{\xi_{i+1}} + Q_i^{**}} - \sum_{e=1}^{j-1} p_{i_e} n_{i_e}}{o_{\xi_{i+1}}},$$

□

At this point it is clear that the w -th root of a DDS is always unique, if it exists (i.e., if all n_i 's turn out to be natural numbers).

5. Intersection between abstractions

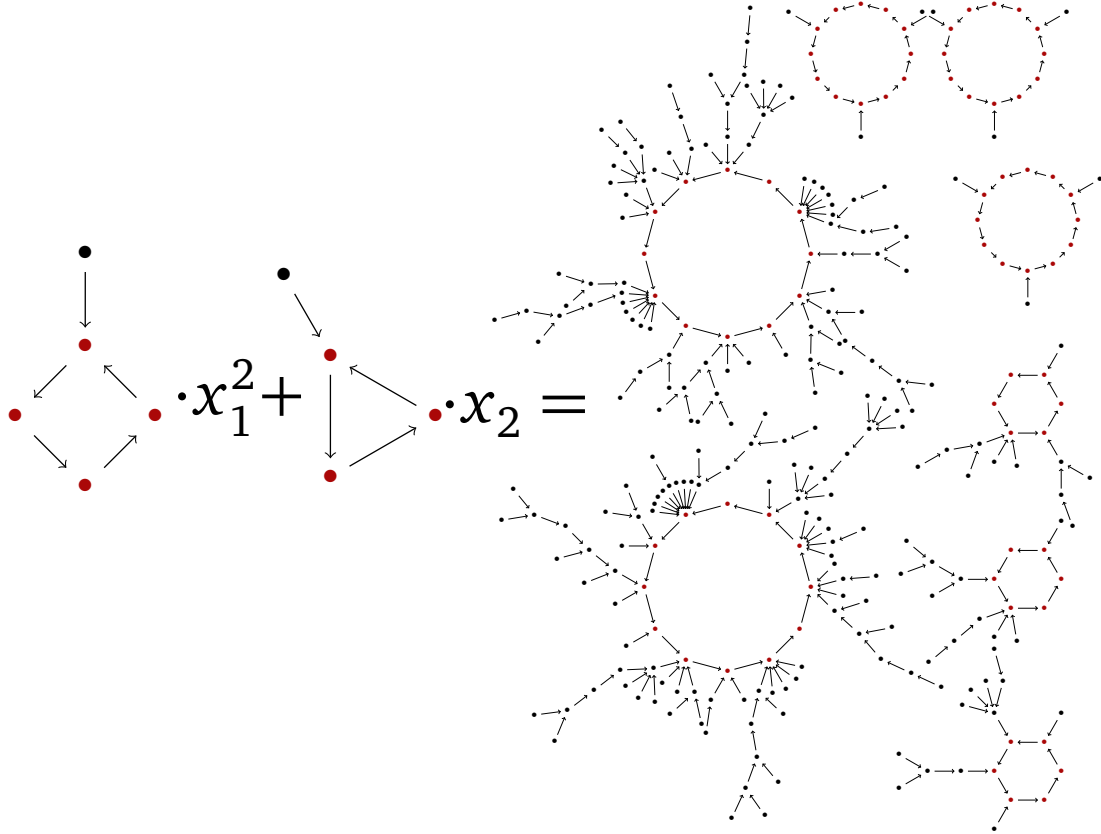


Figure 8: An example of Equation 1. The coefficients a_1 , a_2 and the know term b are depicted by their dynamics graphs.

Once considered both the c-abstraction and a-abstraction of Equation (1) and provided the two corresponding solution sets, the final step to perform - that we name intersection between abstractions - is combining each solution from the first set with each solution from the second one to establish what resulting pairs lead to a possible solution of Equation (1). In other words, (x_1, \dots, x_ν) is a solution candidate of Equation (1) if each of the tuples $(|x_1|, \dots, |x_\nu|)$ and $(\hat{x}_1, \dots, \hat{x}_\nu)$ belongs to the solution set of the c-abstraction and a-abstraction

equation, respectively. Moreover, a solution of the c-abstraction equation can be combined with one of the a-abstraction equation, if for every i the total number of periodic points of \hat{x}_i is at most $|x_i|$. Let us illustrate such a final step by the following example.

Example 5. Consider the equation

$$a_1 \cdot x_1^2 + a_2 \cdot x_2 = b$$

where a_1 , a_2 , and b are as in Figure 8. The corresponding c-abstraction and a-abstraction equations are

$$5 \cdot |x_1|^2 + 4 \cdot |x_2| = 293 ,$$

and

$$C_4^1 \odot \hat{x}_1^2 \oplus C_3^1 \odot \hat{x}_2 = C_6^3 \oplus C_{12}^5 ,$$

respectively. At this point, we aim at enumerating the solutions of both the abstraction equations. Regarding the c-abstraction one, the MDD of Figure 9 provides the following solutions:

$$|x_1| = 7, |x_2| = 12$$

$$|x_1| = 5, |x_2| = 42$$

$$|x_1| = 3, |x_2| = 62$$

$$|x_1| = 1, |x_2| = 72$$

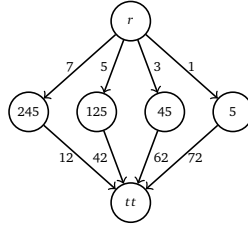


Figure 9: The reduced MDD representing all the solutions of $5 \cdot |x_1|^2 + 4 \cdot |x_2| = 293$. There are $\mathcal{V} = 2$ variables. The first level and the corresponding outgoing edges represent the variable $|x_1|$ and its possible values. The second level and the outgoing edges represent $|x_2|$.

As far the a-abstraction equation is concerned, there are 16 basic equations and, according to the necessary ones, the MDD CS of Figure 10 provide all the feasible way by which the two monomials $C_4^1 \odot X_1$ and $C_3^1 \odot X_2$ can concur to form $C_6^3 \oplus C_{12}^5$, where $X_1 = \hat{x}_1^2$ and $X_2 = \hat{x}_2$. Namely, only the monomial $C_3^1 \odot \hat{x}_2$ contributes to form C_6^3 (see CS₁), while there are several ways by which both of them contribute to form C_{12}^5 . If among the necessary equations involving $X_1 = \hat{x}_1^2$, one considers only those admitting a non empty set of solutions x_1 , i.e., after the computation of the square root of the values of X_1 has been performed too, the following two feasible Systems (8) remain:

$$\left\{ C_3^1 \odot \hat{x}_2 = C_6^3 \oplus C_{12}^5 \right.$$

$$\left. \begin{array}{l} C_4^1 \odot \hat{x}_1^2 = C_{12}^3 \\ C_3^1 \odot \hat{x}_2 = C_6^3 \oplus C_{12}^2 \end{array} \right\}$$

In both cases, the values of $X_2 = \hat{x}_2$ are computed by a Cartesian products of SB-MDDs. Due to the form of \hat{a}_1 and \hat{a}_2 and the fact that the two monomials contain distinct variables, in each system there are no equations involving

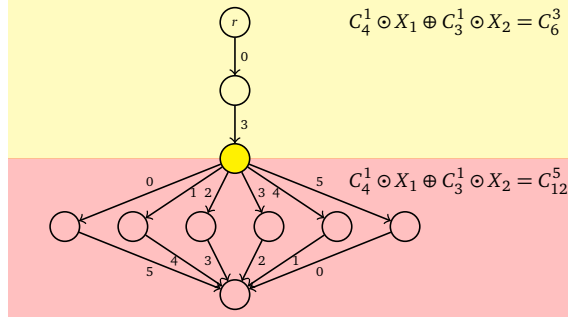


Figure 10: The MDD $CS = CS_1 \times CS_2$ represents all the feasible ways by which, according to the set of necessary equations, the monomials of the a -abstraction equation from Example 5 can concur to form its right-hand side. According to the cartesian product of MDD, the yellow node is at the same time the tt node of CS_1 and the $root$ node of CS_2 . In each of the two MDD, the first (resp., second) level corresponds to the monomial $C_4^1 \odot X_1$ (resp., $C_3^1 \odot X_2$). The values $val(a)$ associated to nodes are omitted for simplicity.

the same variable. Hence, no intersection operation between solutions of equations is required. The solutions of the a -abstraction equation are:

$$\begin{aligned}
 x_1^s &= C_3^1, x_2^s = C_6^1 \oplus C_4^2 \\
 x_1^s &= C_3^1, x_2^s = C_2^3 \oplus C_4^2 \\
 x_1^s &= \emptyset, x_2^s = C_6^1 \oplus C_{12}^1 \oplus C_4^2 \\
 x_1^s &= \emptyset, x_2^s = C_6^1 \oplus C_4^5 \\
 x_1^s &= \emptyset, x_2^s = C_2^3 \oplus C_{12}^1 \oplus C_4^2 \\
 x_1^s &= \emptyset, x_2^s = C_2^3 \oplus C_4^5
 \end{aligned}$$

Some solutions (x_1^s, x_2^s) of the a -abstraction equation can be coupled to no solution $(|x_1|, |x_2|)$ of the c -abstraction equation to lead a solution of the given original equation. Namely, by the solutions of the c -abstraction equation, x_1 necessarily has at least one state. Therefore, the only possible value of x_1^s is C_3^1 . This implies that x_1 must have at least 3 states and $|x_2| \geq 14$ (since x_2^s consists of 14 periodic points). Then, the solutions $(|x_1| = 1, |x_2| = 72)$ and $(|x_1| = 7, |x_2| = 12)$ of the c -abstraction equation can not be coupled with any solution of the c -abstraction equation. This process leads to the identification of the following candidate solutions of the given original equation:

$$(x_1, x_2) \in R^2 \text{ s.t. } (x_1^s = C_3^1) \text{ and } (x_2^s \in \{C_6^1 \oplus C_4^2, C_2^3 \oplus C_4^2\}) \text{ and } ((|x_1| = 3 \wedge |x_2| = 62) \text{ or } (|x_1| = 5 \wedge |x_2| = 72))$$

6. Conclusion

This paper presents a complete algorithmic pipeline for solving both the c - and a -abstractions of polynomial equations (with constant right-hand term) over DDS. The pipeline includes a number of subtleties allowing reasonable performances that are compatible with practical applications.

Devising an algorithm that solves in an efficient way the t -abstraction of an equation over DDS is certainly the main step for further researches concerning this subject. Actually, this is a rather complex task.

A further interesting research direction consists in trying to understand the precise computational complexity of problems that arise when considering the different tasks of the pipeline. For example, what is the computational complexity of establishing whether a basic equation has solutions? It is clear that the problem is in NP but we conjecture that in fact it is in P. Along the same line of thoughts, one finds that the problem of enumerating the solutions of a basic equation is in EnumP but is it complete for this class? Now, stepping to the more complex problem of deciding whether an a -abstraction equation admits a solution, what is precisely its complexity class?

References

- [1] Adamatzky, A., Goles, E., Martínez, G.J., Tsompanas, M.I., Tegelaar, M., Wosten, H.A.B., 2020. Fungal automata. *Complex Syst.* 29. URL: https://www.complex-systems.com/abstracts/v29_i04_a02/.
- [2] Alonso-Sanz, R., 2012. Cellular automata and other discrete dynamical systems with memory, in: Smari, W.W., Zeljkovic, V. (Eds.), *Proceedings of HPCS*, IEEE. p. 215.
- [3] Aracena, J., Cabrera-Crot, L., Salinas, L., 2021. Finding the fixed points of a boolean network from a positive feedback vertex set. *Bioinform.* 37, 1148–1155. URL: <https://doi.org/10.1093/bioinformatics/btaa922>, doi:10.1093/bioinformatics/btaa922.
- [4] Bergman, D., Cire, A.A., van Hove, W., 2014. MDD propagation for sequence constraints. *Journal of Artificial Intelligence Research* 50, 697–722.
- [5] Bergman, D., Cire, A.A., Van Hove, W.J., Hooker, J., 2016. *Decision diagrams for optimization*. volume 1. Springer.
- [6] Bower, J.M., Bolouri, H., 2004. *Computational modeling of genetic and biochemical networks*. MIT press.
- [7] Chaudhuri, P., Chowdhury, D., Nandi, S., Chattopadhyay, S., 1997. *Additive Cellular Automata Theory and Applications*. volume 1. IEEE Press.
- [8] Darwiche, A., Marquis, P., 2002. A knowledge compilation map. *Journal of Artificial Intelligence Research* 17, 229–264.
- [9] Demongeot, J., Melliti, T., Noual, M., Regnault, D., Sené, S., 2022. On boolean automata isolated cycles and tangential double-cycles dynamics, in: Adamatzky, A. (Ed.), *Automata and Complexity - Essays Presented to Eric Goles on the Occasion of His 70th Birthday*, Springer. pp. 145–178. URL: https://doi.org/10.1007/978-3-030-92551-2_11, doi:10.1007/978-3-030-92551-2_11.
- [10] Dennunzio, A., Dorigatti, V., Formenti, E., Manzoni, L., Porreca, A.E., 2018. Polynomial equations over finite, discrete-time dynamical systems, in: *Proc. of ACR18*, pp. 298–306.
- [11] Formenti, E., Régis, J.C., Riva, S., 2021. MDDs boost equation solving on discrete dynamical systems, in: *International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, Springer. pp. 196–213.
- [12] Jongsma, C., 2019. Basic set theory and combinatorics, in: *Introduction to Discrete Mathematics via Logic and Proof*. Springer, pp. 205–253.
- [13] Marañón, G.Á., Encinas, L.H., del Rey, Á.M., 2008. A multisecret sharing scheme for color images based on cellular automata. *Information Sciences* 178, 4382–4395.
- [14] Nandi, S., Kar, B.K., Chaudhuri, P.P., 1994. Theory and applications of cellular automata in cryptography. *IEEE Trans. Computers* 43, 1346–1357.
- [15] Perez, G., Régis, J.C., 2015. Efficient operations on MDDs for building constraint programming models, in: *IJCAI 2015*, pp. 374–380.
- [16] Sené, S., 2012. *On the bioinformatics of automata networks*. HDR. University of Évry Val d’Essonne, France. URL: <https://tel.archives-ouvertes.fr/tel-00759287>.
- [17] Siebert, H., 2009. Dynamical and structural modularity of discrete regulatory networks, in: *COMPMOD*, pp. 109–124.