Journal of Membrane Computing (2023) 5:182–189
https://doi.org/10.1007/s41965-023-00130-z

REVIEW PAPER

Watson-crick (D)0 L systems: a survey

Petr Sosík[1]

Received: 5 January 2023 / Accepted: 17 August 2023 / Published online: 29 August 2023
© The Author(s) 2023

Abstract
Watson-Crick L systems are string generating formal models obtained by augmenting Lindenmayer systems with the *Watson-Crick morphism* inspired by the Watson-Crick complementarity principle. The Watson-Crick morphism is controlled by a *trigger*—a boolean condition eventually met by the generated word. Despite their simplicity, Watson-Crick (D)0 L systems have interesting mathematical properties, a strong generative power, and they can also characterize some open decision problems. Furthermore, networks of Watson-Crick D0L systems are capable of trading space for time, and thus of characterizing linear time solutions to intractable problems. We provide a survey of results in this field since its origin in 1997 to the present, and we conclude with a series of open research problems and ideas.

Keywords DNA computing · Watson-Crick complementarity · Watson-Crick D0L system

Mathematics Subject Classification 68Q05 · 68Q42 · 68Q45 · 92D20

1 Introduction

The field of molecular computing and, in particular, DNA-inspired computing, has attracted a considerable interest since 1990's, thanks to the series of effective experiments showing possibilities of algorithmic processes at the molecular level, carried out with the use of DNA or RNA molecules. Besides experiments, thousands of theoretical studies appeared, investigating the mathematical and algorithmic properties of DNA. While in the later decades, the focus shifted more to DNA nanotechnology (as can be documented by many results published in the leading journals as Nature, Science, or in the series of DNA computing conferences), there still remain many open questions related to the theoretical properties of DNA from the point of view of computer science.

Out of these properties, two seem crucial for the DNA computing research: the massive parallelism of biochemical reactions including the polymerase chain reaction (PCR), and the Watson-Crick complementarity principle. While the theoretical research focused mainly on the former in an attempt to employ it to solve hard computational problems efficiently, the latter was devoted less attention. In this review, we focus on some formal models depending on and exploiting the Watson-Crick complementarity principle.

The Watson-Crick complementarity pairing, i.e., hydrogen bonds between pairs of nucleotides: adenine (A) with thymine (T), and guanine (G) with cytosine (C), is one of crucial principles in molecular biology, allowing the function of DNA as a carrier of genetic information. Though seemingly simple, mathematical properties of the Watson-Crick complementarity make it an interesting tool which provides traditional computer science formalisms (like formal language generators and acceptors) with rather surprising power. Together with opposite orientation of two single-stranded DNA chains forming the double-stranded molecule of DNA, this principle inspired a sequence of both theoretical results and applications in the field called DNA computing.

Here, we focus on a specific sub-field of DNA computing, founded by Arto Salomaa together with Valeria Mihalache in 1997: Watson-Crick D0L systems and their variants (extended or tabled systems and further generalizations). As the general principle, the Watson-Crick complementarity principle is generalized to the form of a morphism mapping letters to letters and strings to strings. As the title suggests,

✉ Petr Sosík
petr.sosik@fpf.slu.cz

[1] Institute of Computer Science, Faculty of Philosophy and Science, Silesian University in Opava, Bezručovo nám. 13, Opava 74601, Czech Republic

this principle is then coupled with some formal generative paradigms, in particular with Lindenmayer systems.

Though there have already been achieved very interesting results, there still remain many open questions and research avenues which we believe are worth pursuing. To our best knowledge, apart from the invited talk of Erzsébet Csuhaj-Varjú at the 8th Conference on Machines, Computations and Universality, Fontainebleau, France, 2018 (of which only the abstract is publicly available [2]), no survey exists on this topic yet. This paper intends to provide a comprehensive overview of this research field, from its origins to the latest results, as well as to present a series of open research problems.

2 Formal pre-requisites

We start with some elementary notation of formal language theory; the reader is referred to [15] for a broad overview. For a finite alphabet $\Sigma = \{a_1, \ldots, a_n\}$, denote by Σ^* the set of all finite words over Σ, including the empty word λ. For $w \in \Sigma^*$, $|w|$ denotes the length of w, and for $a \in \Sigma$, $|w|_a$ is the number of occurrences of a in w. For $\Gamma \subseteq \Sigma$, $|w|_\Gamma = \sum_{a\in\Gamma} |w|_a$. For $w \in \Sigma^*$ we denote by w^n the catenation of n copies of $w, n \geq 1$.

An *endomorphism* of Σ is a mapping $g : \Sigma^* \longrightarrow \Sigma^*$ preserving catenation, i.e., $g(uv) = g(u)g(v)$ for any $u, v \in \Sigma^*$. The *matrix* M_g of g is the $n \times n$ integer matrix with values $|g(a_j)|_{a_i}$ at positions (i, j), $1 \leq i, j \leq n$.

Definition 1 A *D0L system* is a triple $G = (\Sigma, g, w_0)$, where Σ is an alphabet, g is an endomorphism of Σ^* and $w_0 \in \Sigma^*$ is the *axiom*.

It is customary to define g in terms of *productions*, indicating the image of each letter. With a D0L system, we associate the infinite *sequence* $S(G)$ of words $w_i, i \geq 0$, where $w_{i+1} = g(w_i)$, for all $i \geq 0$, the *length sequence* $|w_i|, i \geq 0$, as well as the *growth function* $f(i) = |w_i|$. Furthermore, the set of all words in $S(G)$ defines the *language* $L(G)$.

A *0 L system* is defined in the same way as a D0L system but g is a finite substitution. In other words, there can be several (but at least one) productions for the same letter, and one of them is non-deterministically chosen for each occurrence of that letter in a derivation step.

In *tabled 0 L systems*, the set of productions is divided into subsets called tables, each table containing at least one production for each letter. One table is non-deterministically chosen in each derivation step. The use of tables is indicated by the letter T in the name of the system.

Extended 0 L systems (marked with the letter E) have their alphabet divided into the set of terminals and non-terminals. Only words consisting of terminals belong to the generated language.

By a combination of these properties we obtain D0L, 0 L, T0L, E0L, DT0L, ET0L and EDT0L variants of Lindenmayer systems.

3 Watson-Crick D0L systems

The basic principle of Watson-Crick D0L systems introduced in [13] is the following mathematical abstraction of biological function of DNA: consider a certain condition (called the trigger) expressing an undesirable state of a single-stranded DNA molecule, e.g., its instability (or just any detectable condition). In this situation, the molecule is replaced by its complement (which, in reality, could be produced by some reaction alike PCR). Note that chemically adenine and guanine are purine bases, while cytosine and thymine are pyrimidine bases. As the C-G paring consists of three hydrogen bonds while A-T uses only two, DNA with higher CG content is more stable to denaturation or melting. Therefore, it is natural to define triggers based on the content of some letters in the string (as long as the string is understood as an abstraction of a DNA strand).

One of the fascinating properties of Watson-Crick D0L systems is that one of the simplest parallel generative mechanism, a D0L system, reaches computational universality when it is complemented with a simple trigger based on majority of certain symbols in the generated string.

The Watson-Crick complementarity principle can be mathematically expressed as follows: denote $\Sigma_{\text{DNA}} = \{A, C, G, T\}$ the *DNA alphabet.* Consider the letter-to-letter endomorphism h_W of Σ^*_{DNA} defined by

$$h_W(A) = T,\ h_W(T) = A,\ h_W(G) = C,\ h_W(C) = G.$$

More generally, a *DNA-like alphabet* Σ_n is an alphabet with $2n$ letters, $n \geq 1$, denoted as follows:

$$\Sigma_n = \{a_1, \ldots, a_n, \bar{a}_1, \ldots, \bar{a}_n\}.$$

We call a_i and $\bar{a}_i$ *complementary letters.* The generalized endomorphism h_W of Σ^*_n defined by

$$h_W(a_i) = \bar{a}_i,\ h_W(\bar{a}_i) = a_i, \quad 1 \leq i \leq n,$$

is called the *Watson-Crick morphism.* Observe that h_W is uniquely determined by Σ_n.

In analogy with the biological DNA, we call the non-barred letters *purines* and the barred letters *pyrimidines,* and we denote $\Sigma^{\text{PUR}}_n = \{a_1, \ldots, a_n\}$ and $\Sigma^{\text{PYR}}_n = \{\bar{a}_1, \ldots, \bar{a}_n\}$. Furthermore, we also denote the set of words in Σ^*_n with strict majority of pyrimidines by *PYR,* and its complement w.r.t.

Σ_n^* by *PUR*. It can be easily shown that both *PYR* and *PUR* are deterministic context-free languages.

Definition 2 ([13]) A *Watson-Crick D0L system* is a pair $G_W = (G, TR)$, where $G = (\Sigma_n, g, w_0)$ is a D0L system with a DNA-like alphabet Σ_n, $TR \subset \Sigma_n^*$ is called the *trigger*, and $w_0 \in \Sigma_n^* \setminus TR$. The associated sequence of words $S(G_W)$ is defined by

$$w_{i+1} = \begin{cases} h_W(g(w_i)), & \text{if } g(w_i) \in TR, \\ g(w_i), & \text{if } g(w_i) \notin TR. \end{cases}$$

If not stated otherwise, all the results we refer to in this survey assume sound triggers: $x \in TR$ implies $h_W(x) \notin TR$ for all $x \in \Sigma_n$.

The transition $w_i \Longrightarrow_{G_W} w_{i+1}$ is called a *derivation step* of G_W. If $w_{i+1} = h_W(g(w_i))$, then we say that a *complementarity transition* has taken place, or that the trigger h_W fires. As usual, we define the language $L(G_W)$ as the set of all words in $S(G_W)$. In the sequel we also use the term *Watson-Crick D0L scheme*, which is just a Watson-Crick D0L system without an axiom.

Finally, we call a Watson-Crick D0L system *standard* if $TR = PYR$. Observe that any word in *PYR* is mapped by h_W to its complement which is in *PUR*, so that the sequence (and the language) of a standard Watson-Crick D0L system contains only words in *PUR*.

3.1 Sequences, equivalences, reachability

Watson-Crick D0L systems were introduced in 1997 in the paper [13] which attracted researchers already familiar with Lindenmayer systems to the phenomenon of Watson-Crick complementarity. The paper focused on growth functions of Lindenmayer systems, i.e., on the sequences of lengths of the generated words. As the (Watson-Crick) D0L systems are deterministic, each such system generates a unique length sequence. The main result of the paper can be formulated as follows: While D0L growth functions are a special case of $\mathbb{N}$-rational functions, Watson-Crick D0L growth functions are not even $\mathbb{Z}$-rational.

Further study [16] focused on representation of a group of decidability problems related to Watson-Crick D0L systems by so-called Watson-Crick walks on D0L graphs. Consider a Watson-Crick D0L system $G_W = (G, TR)$ with an underlying D0L system $G = (\Sigma_n, g, w_0)$, a Watson-Crick morphism h_W and the associated sequence $S(G_W) = w_i, i \geq 0$. We associate to G_W an infinite binary sequence $b_i, i \geq 1$ as follows: $b_i = 0$ if $w_i = g(w_{i-1})$, and $b_i = 1$ if $w_i = h_W(g(w_{i-1}))$. This sequence is called the Watson-Crick road of G_W. Intuitively, a Watson-Crick walk associated to G_W is identified by any prefix of its Watson-Crick road.

Formally, the construction in [16] is more general. Let $\equiv$ be a *morphic equivalence* on Σ_n^* preserving any morphism h of Σ_n^* : if $x \equiv y$, then also $h(x) \equiv h(y)$. A Watson-Crick graph $H_W(G_W, \equiv)$ is a directed graph (digraph) constructed as follows:

- its nodes are the equivalence classes of $L(G_W)$, i.e., sets of words equivalent under $\equiv$;
- each node corresponding to a class X has two outgoing arcs: one (labelled by 0) pointing to the node $g(X)$, and another (labelled by 1) pointing to $h_W g(X)$, where $h_W g$ is the composition of morphisms h_W and g.

Then, any binary sequence of arc labels uniquely identifies a walk in this graph. Only the walks which are prefixes of the Watson-Crick road of G_W are considered in [16]. Note that, in every walk, the binary sequence itself is independent on $\equiv$ but the sequence of visited nodes depends on it. Watson-Crick walks allow to characterize many decidability questions of Watson-Crick D0L systems.

The stability problem. Does the trigger of a Watson-Crick D0L system ever fires? (Or, equivalently: is the associated Watson-Crick road equal to 0^ω)?
The reachability problem. Is a given node in the Watson-Crick graph reachable from the initial node?
Equivalence problems. Are the roads, growth functions, sequences or languages of two Watson-Crick D0L systems equivalent?

These problems were shown undecidable for context-sensitive triggers, and the stability problem was proven decidable for regular triggers.

Another set of intensively studied problems is related to *rational functions*. Rational functions are motivated by growth functions of parallel rewriting systems and they can be defined as a result of iterated growth matrix multiplication. A function $f : \mathbb{N} \longrightarrow \mathbb{Z}$ is called $\mathbb{N}/\mathbb{Z}/\mathbb{Q}$-rational, respectively, if it can be expressed in the form

$$f(i) = \Pi M^i \eta, \ i \geq 0$$

where Π and η are row and column vectors, and M is a square matrix, all with entries from $\mathbb{N}/\mathbb{Z}/\mathbb{Q}$. The corresponding integer sequence $f(0), f(1), f(2), \ldots$ is termed $\mathbb{N}/\mathbb{Z}/\mathbb{Q}$-rational, respectively.

More intuitively, $\mathbb{N}$-rational functions coincide with growth functions of HD0L systems, that is, morphic images of languages of D0L systems. $\mathbb{Z}$-rational functions coincide with differences of two $\mathbb{N}$-rational functions. For further details on rational functions please consult

monographs [12, 18]. Now we are ready to recall an important open problem related to $\mathbb{Z}$-rational sequences:

Problem $\mathbb{Z}_{\text{pos}}$. Decide whether or not a negative number appears in a given $\mathbb{Z}$-rational sequence of integers.

Furthermore, the stability problem for standard Watson-Crick D0L systems was shown to be algorithmically equivalent to the $\mathbb{Z}_{\text{pos}}$ problem.

The results mentioned above were summarized and further elaborated in [14], where further decidability problems of Watson-Crick D0L systems were studied, as the *letter appearance problem*. Its specific version for the case of Watson-Crick D0L systems is the problem of *ghost letters*: is there a letter in Σ_n which never appears in $S(G_W)$? Finally, the problem of *Watson-Crick-ness* is stated as follows: is a given (D0L) word sequence a Watson-Crick D0L sequence? This problem induces another question: which operations on words in a sequence preserve its Watson-Crick-ness? For instance, let *dual sequence* be the complementary image of the original sentence (i.e., all letters are replaced by their complements). Then the duality operation preserves the Watson-Crick-ness of word sequences.

3.2 Computational universality

Another milestone in the research of the Watson-Crick D0L systems was marked by the paper [21] and its extended version [22], showing that standard Watson-Crick D0L systems are universal in the Turing sense. In particular, these papers demonstrate that any partial recursive integer function can be computed by a Watson-Crick D0L scheme augmented with various axioms representing arguments of the function. Both the argument(s) and value(s) of the function are represented by numbers of occurrences of specific letters of the alphabet in the axiom and in the generated string, respectively. If a derivation does not produce an output string consisting of the pre-defined letters, then the computed function is undefined for the corresponding argument.

Example 1 ([22]) The following standard Watson-Crick D0L scheme computes the function $f(n) = \lceil \log_3 n \rceil$, where $\lceil x \rceil$ is the ceiling of x. Let $G = (\Sigma_8, p)$, where

$$\Sigma_8 = \{B, E, a, b, c, d, s, e, \bar{B}, \bar{E}, \bar{a}, \bar{b}, \bar{c}, \bar{d}, \bar{s}, \bar{e}\},$$
$$p(B) = \bar{s}\bar{a} \; p(s) = E \; p(\bar{s}) = \bar{s}c\bar{d} \;\; p(a) = \lambda$$
$$p(\bar{c}) = \lambda \;\; p(\bar{b}) = \lambda \; p(\bar{a}) = \bar{a}\bar{a}\bar{a} \; p(d) = e$$

and $p(X) = X$ for all other $X \in \Sigma_8$. The number of b's represents an argument of the function, while the number of e's is its value. The following derivation corresponds to $\log_3 9 = 2$:

$$Bb^9 \Longrightarrow \bar{s}\bar{a}b^9 \Longrightarrow \bar{s}c\bar{d}\bar{a}^3b^9 \Longrightarrow s(\bar{c}d)^2a^9\bar{b}^9 \Longrightarrow Ee^2$$

Note that, without using triggers, not even tabled D0L systems can provide non-linear dependence between the numbers of "input" and "output" symbols in generated sequences.

The previous result was further generalized in [23] to languages generated by Watson-Crick D0L systems, in the sense that any recursively enumerable language can be obtained as a projection of a language of some standard Watson-Crick D0L system. As a consequence, several problems were proven to be undecidable for standard Watson-Crick D0L systems: the letter appearance problem, the reachability problem, the growth, sequence and language equivalence problem, and the ultimate stability problem.

The letter appearance problem. Does a given letter ever appears in some word in the generated sequence?
The ultimate stability problem. Does the trigger of a Watson-Crick D0L system fires only finitely many times?

3.3 Restrictions: regular triggers, uni-transitional systems

The computational universality of standard Watson-Crick D0L systems clearly demonstrated the power of the triggering mechanism, and simultaneously, it indicated that the standard trigger *PYR* (which is a deterministic context-free language) is already "too strong" to provide an interesting study material. Therefore, the research focused on more restricted versions of Watson-Crick D0L systems. One of such natural restrictions is the form of a trigger. Watson-Crick D0L systems with regular triggers were systematically studied in [11] and the road, growth, sequence and language equivalence problem were shown to be all decidable for these Watson-Crick D0L systems. Furthermore, their growth functions were shown to be D0L growth functions.

To complement the study mentioned above, the Watson-Crick D0L systems with standard (i.e. non-regular) triggers but with a bounded balance were studied in [9]. A Watson-Crick D0L system has a *bounded balance* if the absolute value of the difference between the number of purines and pyrimidines is bounded by a constant in all words in the generated sequence. Again, as the result, the road, growth, sequence and language equivalence problem are all decidable for standard Watson-Crick D0L systems with bounded balance.

Another restriction imposed on standard Watson-Crick D0L systems in order to diminish their power is to allow at most one firing of the trigger, i.e., a single transition to the complementary string during the whole derivation. In [17], uni-transitional Watson-Crick D0L systems (UT systems, for short) were studied. It is rather intuitive that any language generated by a UT system is of the form $F \cup L$, where F is

a finite language and L is a D0L language. Note that, after a finitely many steps till the first transition, a UT system continues as an ordinary D0L system. More interesting, however, are their computability and decidability properties. Consider, for instance, the Watson-Crick D0L scheme in Example 1. The function computed by the scheme is not even $\mathbb{Z}$-rational, while all its computations with axioms of the form Bb^i, $i \geq 0$, are uni-transitional. It was further shown that the stability problem, the finiteness problem and the sequence, road, growth and language equivalence problems of UT systems are all algorithmically equivalent to the $\mathbb{Z}_{\text{pos}}$ problem.

The study of uni-transitional Watson-Crick D0L system continued in [19] with focus on properties of functions computable by UT systems, in the sense of Example 1. Several characterizations of UT-computable functions in terms of $\mathbb{N}$-rational, $\mathbb{Z}$-rational and $\mathbb{Q}$-rational functions were provided. For instance, the following functions are UT-computable: any $\mathbb{N}$-rational function, and any function of the form $f(g^{-1})$, where f is $\mathbb{N}$-rational and g is $\mathbb{Z}$-rational and strictly monotonic. Furthermore, the following problem is algorithmically equivalent to $\mathbb{Z}_{\text{pos}}$: given a UT-computable function f and an integer c, decide whether or not $c \in \text{dom}(f)$.

3.4 Extended Watson-Crick D0L systems, tables

We have already mentioned in Sect. 3.2 that any recursively enumerable language is a projection of the language of a Watson-Crick D0L system. Papers referred to in this section focus on the effect of extended variants of the underlying Lindenmayer systems on the generated language, while simultaneously imposing other kinds of restrictions. The first results in this direction appeared in [1]. In particular, the authors showed that any recursively enumerable language can be generated by a standard Watson-Crick E0L system, or by a standard Watson-Crick EDT0L system where either the number of non-terminals or the number of tables (but not both) is bounded by a constant. (The constant is 2 in the case of tables.) Comparison with the result in [23] about representation of any recursively enumerable language as a projection of the language of a standard Watson-Crick D0L system suggests the conjecture that also standard Watson-Crick ED0L systems might generate all recursively enumerable languages.

In a follow-up study [20] the authors focus on extended Watson-Crick 0 L systems with regular triggers. Rather surprisingly, they show that uni-transitional Watson-Crick E0L systems with regular triggers are still able to generate all recursively enumerable languages.

Then so-called *weak derivation mode* is studied: when a string in *PYR* appears in the generated sequence, the system non-deterministically chooses whether to apply complementary transition or not. Watson-Crick E0L systems with regular triggers can generate all recursively enumerable languages even in the weak mode, but multiple complementary transitions are needed. If restricted to uni-transitional systems, the obtained class of languages is bounded by the class of ET0L languages. As a consequence, when the number of complementary transitions is bounded by a constant, then Watson-Crick ET0L systems with regular triggers generate in the weak mode exactly the class of ET0L languages. Therefore, there is an interesting trade-off between the determinism of the trigger and the number of complementary transitions.

In the last section, the authors show that if a context-free non-standard trigger is allowed, then all recursively enumerable languages can be generated also by uni-transitional Watson-Crick EDT0L systems.

Finally, an interesting connection between tabled Watson-Crick 0 L systems and red-green Turing/register machines was investigated in [5]. Red-green machines are formal devices which perform an infinite computation on a finite input. Each state is marked either red or green. The input is recognized during an infinite run if and only if the sequence of visited states contains finitely many occurrences of red states. It is known that both red-green Turing and register machines can recognize complements of recursively enumerable languages and, hence, they have super-Turing power.

The authors defined recognizing of an input (axiom) by Watson-Crick 0 L schemes according to the result of the ultimate stability problem (see above for definition). Subsequently they showed that computations (and recognizing) of Turing or register red-green machines can be simulated by computations (and recognizing) of standard Watson-Crick DT0L systems, and vice versa. As a consequence, the class of sets (of integers) recognized by red-green Turing/register machines equals the class of sets recognized by standard Watson-Crick DT0L systems.

3.5 Networks of Watson-Crick D0L systems

The first note on networks of Watson-Crick D0L systems appeared already in [3]. The main idea is based on a network graph (a complete one, by default) whose nodes are Watson-Crick D0L systems and edges are communication lines between them. All nodes share the same alphabet, the Watson-Crick morphism h_W and the trigger TR (PYR by default), but have their own D0L morphisms and axioms. Each node generates its own set of strings (i.e., a language) but, unlike conventional D0L system, the D0L morphism applies in each step to all strings in the node's language, and the resulting strings which are in TR are changed to their complements by applying the Watson-Crick morphism h_W. All thus obtained strings are added to the node's language.

At the end of the step, copies of some of the produced strings are communicated to neighbouring nodes and added

to their languages. There are two protocols: according to the protocol (a), the node sends all string to which h_W was applied in the recent step, and according to (b), it sends all string to which h_W was *not* applied. The language generated by the network is the language of a selected *master node*.

The initial study [3] confirmed that, under the protocol (b), the networks of Watson-Crick D0L systems generate all recursively enumerable languages. In the light of the universality results in Sect. 3.2, the network communication can replace the projection used in [23] to obtain any recursively enumerable language. Then the paper [7] demonstrated that both derivation protocols are equivalent, provided that the trigger TR satisfies $x \in TR \Leftrightarrow h_W(x) \notin TR$ for all $x \in \Sigma_n$. The authors further define the string population growth function $p(t)$ of a network as the total number of all strings in the network after t derivation steps. They show that, even under severe restrictions, the network can have a string population growth function which is not $\mathbb{Z}$-rational. Finally, they demonstrate that, under protocol (a), the existence of a node which never communicates any string to others (a "black hole") is at least as difficult to decide as the $\mathbb{Z}_{\text{pos}}$ problem.

The follow-up study [8] focuses on intractable decision problems and it shows that the problems SAT and HPP can be solved in linear time by networks of Watson-Crick D0L systems, thanks to their massive parallelism and population growth. Furthermore, the previous results on string population growth functions are strengthened, showing that a network with two nodes and the four letter DNA alphabet is enough to have a non $\mathbb{Z}$-rational growth function.

Finally, an interesting result was shown in [4]: the networks of Watson-Crick D0L systems remain computationally universal (i.e., capable of generating any recursively enumerable language) even if the nodes do not communicate full generated strings but only their non-empty prefixes or suffixes.

3.6 Other related generative models

A connection between Watson-Crick E0L system and membrane (P) systems operating on multisets was established in [6]. Similarly as 0 L systems are parallel string-rewriting devices, P systems are parallel multiset-rewriting devices. There is also an analogy between context-free productions/rules used in 0 L systems and non-cooperative rules in P systems. The triggering mechanism as implemented in P systems in the form of a *checker*—a finite-state automaton operating over objects in a configuration of the P system. At each of its steps, the checker marks one object in a specified membrane and changes its state. If all objects are marked and the checker reaches its final state, then the configuration is accepted. The P system passes between configurations in the usual maximally parallel (synchronized) mode, but each new configuration must be accepted by the checker; otherwise, the computation is blocked as unsuccessful. A *non-synchronized* mode is also considered: at each step, the P system applies an arbitrary sub-multiset of the multiset of applicable rules. Furthermore, different configuration can be checked by different checkers: there is a simple regular expression determining the order in which checkers are applied. The resulting construction is called the *Darwinian P system.*

The first two results in the paper deal with the issues of synchronicity and the number of necessary checkers: (i) any maximally parallel (i.e., synchronized) ordinary P system can be simulated by a non-synchronized Darwinian P system with two checkers, (ii) in the synchronized mode, any Darwinian P system with n checkers, $n \geq 1$, can be simulated by a system with one checker. It is unknown whether a similar result holds for non-synchronized mode.

Now, recall that, due to [20], uni-transitional Watson-Crick E0L systems with regular triggers can generate all recursively enumerable languages. Actually, the Watson-Crick E0L system used in [20] generates a terminal string only if no complementary transition occurs during its derivation. The working mode of the Darwinian P systems is very similar: if a trigger would fire in a certain configuration (which is verified by a checker), then the derivation is blocked. Therefore, the following results were obtained: any recursively enumerable set of natural numbers can be generated by a Darwinian P system either (i) in the synchronized mode with one checker, or (ii) in the non-synchronized mode with two checkers. It is also possible to restrict the number of states of each checker to 3, but an unlimited number of checkers is necessary in this case.

A different extension of the concept of Watson-Crick 0 L systems was proposed in [10]. A *discrete Watson-Crick dynamical system* is defined in an analogous way as a Watson-Crick D0L scheme, with the exception that an arbitrary mapping $f : \Sigma_n^* \longrightarrow \Sigma_n^*$ is used instead of the D0L endomorphism g.

The first series of results in the paper is devoted to the relation between types of triggers of Watson-Crick dynamical systems and the mathematical properties of the roads of these systems, namely whether they form $\mathbb{N}$-rational or $\mathbb{Z}$-rational sequences, and whether they are periodic or not. Then, a key result of the paper follows in Theorem 6: a set of sufficient conditions is given, allowing to effectively compute the road of a discrete Watson-Crick dynamical system with a given axiom.

This result has interesting consequences for computability and decidability problems of Watson-Crick D0L systems. We say that a Watson-Crick D0L system $G_W = (G, TR)$ with $G = (\Sigma_n, g, w_0)$ is *integral* if the characteristic values of the matrix M_g of g are integers. We say that G_W is *semistable* if the number of 1's is finite in the road of G_W. Consider a standard Watson-Crick D0L system G_W which is integral and

semistable, then the road of G_W can be computed effectively. Furthermore, the road, sequence and language equivalence problems are all decidable, and one can effectively construct a Watson-Crick D0L system which is sequence equivalent and has a regular trigger.

4 Conclusion and open problems

The idea of Watson-Crick 0 L systems, especially in the case of D0L systems, has several attractive properties. Among them, there is its mathematical simplicity and elegance on one hand, and its rather surprising computational potential on the other hand. It is also closely related to some long-standing open problems, especially the problem $\mathbb{Z}_{\text{pos}}$. Last but not least, it interconnects several important scientific concept in different fields: the Watson-Crick complementarity principle in biology, the Lindenmayer generative systems in the field of formal languages, the theory of rational functions and sequences, the field of discrete dynamical systems and some others. Although the main portion of research papers studying this topic appeared between 1997 and 2003, this concept still inspires new result and approaches which appear from time to time.

Among interesting open problems we name, at the first place, the series of decidability problems algorithmically equivalent to the problem $\mathbb{Z}_{\text{pos}}$: the stability problem, the finiteness problem, the sequence, road, growth and language equivalence problems for uni-transitional standard Watson-Crick D0L systems, as well as some other related problems. A few examples of further open questions or research directions follow:

- What is the generative power of standard Watson-Crick ED0L systems, can they generate all recursively enumerable languages?
- How to characterize more precisely the power of Watson-Crick D0L systems with regular triggers?
- What other kind of triggers simpler that the standard ones would lead to the computational universality, or would characterize another interesting class of functions or languages? For instance, what about linear triggers?
- Are the sufficient conditions for the road computability stated in [10] also necessary, and if not, how to weaken them?
- How to specify more precisely the borderline between decidability and undecidability of some basic problems mentioned above, e.g., in terms of triggers or other restrictions on the system?
- Can the networks of Watson-Crick D0L systems characterize the **P** versus **NP** borderline, and how?
- Are there useful connections to other bio-inspired concepts in the formal language theory as the twin shuffle language, Watson-Crick automata or similar?
- What about applications in other areas of computer science as, e.g., the coding theory?

Acknowledgements I am deeply grateful to Arto Salomaa for inspiration, support, tutoring and collaboration in the period when the Watson-Crick D0L systems experienced their golden age. This work was supported by the Silesian University in Opava under the Student Funding Plan, project SGS/11/2023.

Funding Open access publishing supported by the National Technical Library in Prague.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Csima, J., Csuhaj-Varjú, E., & Salomaa, A. (2003). Power and size of extended Watson-Crick L systems. *Theoretical Computer Science, 290*(3), 1665–1678.
2. Csuhaj-Varjú, E.: Watson-Crick complementarity, L systems, computation. In: Durand-Lose, J., Verlan, S. (eds.) Machines, Computations, and Universality: 8th International Conference. Lecture Notes in Computer Science, vol. 10881. Springer, Cham (2018). pp. XI–XII
3. Csuhaj-Varjú, E. (2000). Computing by networks of standard Watson-Crick D0L systems. In M. Ito (Ed.), *Algebraic Systems, Formal Languages and Computations* (Vol. 1166, pp. 42–51). Kyoto University, Research Institute for Mathematical Sciences.
4. Csuhaj-Varjú, E. (2004). Networks of standard Watson-Crick D0L systems with incomplete information communication. In J. Karhumäki, H. Maurer, G. Păun, & G. Rozenberg (Eds.), *Theory Is Forever* (pp. 35–48). Cham: Springer.
5. Csuhaj-Varjú, E., Freund, R., & Vaszil, G. (2017). Watson-Crick T0L systems and red-green register machines. *Fundamenta Informaticae, 155*(1–2), 111–129.
6. Csuhaj-Varjú, E., Martín-Vide, C., Păun, G., & Salomaa, A. (2003). From Watson-Crick L systems to Darwinian P systems. *Natural Computing, 2*(3), 299–318.
7. Csuhaj-Varjú, E., & Salomaa, A. (2003). Networks of Watson-Crick D0L systems. In M. Ito & T. Imaoka (Eds.), *Words, Languages & Combinatorics III* (pp. 134–150). Singapore: World Scientific.
8. Csuhaj-Varjú, E., & Salomaa, A. (2003). The power of networks of Watson-Crick D0L systems. In N. Jonoska, G. Păun, & G.

Rozenberg (Eds.), *Aspects of Molecular Computing* (pp. 106–118). Cham: Springer.
9. Honkala, J. (2003). Decidability results for Watson-Crick D0L systems with nonregular triggers. *Theoretical Computer Science, 302*(1–3), 481–488.
10. Honkala, J. (2017). Discrete Watson-Crick dynamical systems. *Theoretical Computer Science, 701*, 125–131.
11. Honkala, J., & Salomaa, A. (2001). Watson-Crick D0L systems with regular triggers. *Theoretical Computer Science, 259*(1–2), 689–698.
12. Kuich, W., Salomaa, A.: Semirings, Automata, Languages. EATCS monographs on theoretical computer science, vol. 5. Springer, Cham (2012)
13. Mihalache, V., & Salomaa, A. (1997). Watson-Crick D0L systems. *EATCS Bulletin, 62*, 160–175.
14. Mihalache, V., & Salomaa, A. (2001). Language-theoretic aspects of DNA complementarity. *Theoretical Computer Science, 250*(1–2), 163–178.
15. Rozenberg, G., & Salomaa, A. (2012). *Handbook of Formal Languages*. Cham: Springer.
16. Salomaa, A. (1999). Watson-Crick walks and roads on dol graphs. *Acta Cybernetica, 14*(1), 179–192.
17. Salomaa, A. (2002). Uni-transitional Watson-Crick D0L systems. *Theoretical Computer Science, 281*(1–2), 537–553.
18. Salomaa, A., & Soittola, M. (2012). *Automata-theoretic Aspects of Formal Power Series*. Cham: Springer.
19. Salomaa, A., & Sosík, P. (2003). Watson-Crick D0L systems: the power of one transition. *Theoretical Computer Science, 301*(1–3), 187–200.
20. Sears, D., & Salomaa, K. (2012). Extended Watson-Crick L systems with regular trigger languages and restricted derivation modes. *Natural Computing, 11*(4), 653–664.
21. Sosík, P.: D0L system + Watson-Crick complementarity = universal computation. In: Machines, Computations, and Universality, Third International Conference. Lecture Notes in Computer Science, vol. 2055, pp. 308–319 (2001). Springer
22. Sosık, P. (2002). Universal computation with Watson-Crick D0L systems. *Theoretical Computer Science, 289*(1), 485–501.
23. Sosík, P. (2003). Watson-Crick D0L systems: generative power and undecidable problems. *Theoretical Computer Science, 306*(1–3), 101–112.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Petr Sosík is a professor of computer science at the Silesian University in Opava, Faculty of Philosophy and Science. He is a head of research unit and a guarantee of several study programmes. His research covers the area of bio-inspired computing (DNA computing, membrane computing, and morphogenetic systems) and lately machine learning applications. He has (co-)authored over 100 book chapters, journal and refereed conference publications. He supervises the work of several PhD students. Several times he was been awarded the Best Paper Award or Research Result of the Year in various conferences.